

# **Stochastic Control for Cooperative Cyber-Physical Networking**

Zur Erlangung des akademischen Grades eines  
Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik  
des Karlsruher Instituts für Technologie (KIT)

**genehmigte**

**Dissertation**

von

**Florian Rosenthal**

Tag der mündlichen Prüfung:

06. Mai 2022

Erster Gutachter:

Prof. Dr.-Ing. Uwe D. Hanebeck

Zweiter Gutachter:

Prof. Dr. Daniel Quevedo



This document is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0): <https://creativecommons.org/licenses/by-sa/4.0/deed.en>

# Acknowledgment

This thesis is the culmination of the research I carried out at the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics of the Karlsruhe Institute for Technology (KIT) within the research project *CoCPN: Cooperative Cyber-Physical Networking*. First of all, I would like to thank my advisor Uwe D. Hanebeck for giving me the opportunity to become part of his research group and his constant trust and support during my work on this project. I also thank Daniel Quevedo for his willingness to be my co-advisor despite of the large physical distance between Karlsruhe and Brisbane.

Being part of the Priority Programme “Cyber-Physical Networking” (SPP 1914) established by the German Research Foundation (*Deutsche Forschungsgemeinschaft*, DFG), my research has mostly been funded by the DFG under grants HA 3789/15-1 and HA 3789/21-1.

Furthermore, I would like to thank the other members of CoCPN, Markus Jung and Martina Zitterbart from the Institute of Telematics, for the close and fruitful collaboration. This thesis would not have been possible without our often eye-opening discussions and their more “communication-oriented” perspective on many issues that emerged during our research.

I enjoyed the work with a lot of colleagues. Maxim Dolgov, Jannik Steinbring, Christof Chlebek, Gerhard Kurz, Christian Tesch, Selim Özgen, Lukas Michiels, Ajit Basarur, Benjamin Noack, Johannes Westermann, Marko Ristic, Christopher Funk, Jana Mayer, Susanne Radtke, Marcel Reith-Braun, Dominik Prossel, Daniel Frisch, Michael Fennel, Benjamin Siebler, Antonio Zea, Florian Pfaff, Kailai Li, Dagmar Gambichler, Alexander Riffel, Achim Langendörfer, and Sascha Faber all accompanied me during my time at the ISAS. I will always remember the joint travels to conferences around the world and our scientific discussions that arose even after work and which occasionally digressed into mycology and ornithology. Special thanks go to Christof and Jana for being wonderful office colleagues and to Michael and Marcel for proof-reading a great deal of this thesis.

Lastly, and most importantly, I give sincere thanks to my family for their invaluable support throughout my studies and the subsequent period as Ph.D. student. A thousand thanks!

Karlsruhe, May 2022

Florian Rosenthal



# Kurzfassung

Die stetig fortschreitende Digitalisierung erlaubt einen immer autonomeren und intelligenteren Betrieb von Produktions- und Fertigungslinien, was zu einer stärker werdenden Verzahnung der physikalischen Prozesse und der Software-Komponenten zum Überwachen, Steuern und Messen führt. *Cyber-physische Systeme (CPS)* spielen hierbei eine Schlüsselrolle, indem sie sowohl die physikalischen als auch die Software-Komponenten zu einem verteilten System zusammenfassen, innerhalb dessen Umgebungszustände, Messwerte und Steuerbefehle über ein Kommunikationsnetzwerk ausgetauscht werden. Die Verfügbarkeit von kostengünstigen Geräten und die Möglichkeit bereits existierende Infrastruktur zu nutzen sorgen dafür, dass auch innerhalb von CPS zunehmend auf den Einsatz von Standard-Netzen auf Basis von IEEE 802.3 (Ethernet) und IEEE 802.11 (WLAN) gesetzt wird. Nachteilig bei der Nutzung von Standard-Netzen sind jedoch auftretende Dienstgüte-Schwankungen, welche aus der gemeinsamen Nutzung der vorhandenen Infrastruktur resultieren und für die Endsysteme in Form von sich ändernden Latenzen und Daten- und Paketverlusten sichtbar werden.

Regelkreise sind besonders anfällig für Dienstgüte-Schwankungen, da sie typischerweise isochrone Datenübertragungen mit festen Latenzen benötigen, um die gewünschte Regelgüte zu garantieren. Für die Vernetzung der einzelnen Komponenten, das heißt von Sensorik, Aktorik und Regler, setzt man daher klassischerweise auf Lösungen, die diese Anforderungen erfüllen. Diese Lösungen sind jedoch relativ teuer und unflexibel, da sie den Einsatz von spezialisierten Netzwerken wie z.B. Feldbussen benötigen oder über komplexe, speziell entwickelte Kommunikationsprotokolle realisiert werden wie sie beispielsweise die Time-Sensitive Networking (TSN) Standards definieren.

Die vorliegende Arbeit präsentiert Ergebnisse des interdisziplinären Forschungsprojekts *CoCPN: Cooperative Cyber-Physical Networking*, das ein anderes Konzept verfolgt und explizit auf CPS abzielt, die Standard-Netze einsetzen. CoCPN benutzt einen neuartigen, kooperativen Ansatz um i) die *Elastizität* von Regelkreisen innerhalb solcher CPS zu erhöhen, das heißt sie in die Lage zu versetzen, mit den auftretenden Dienstgüte-Schwankungen umzugehen, und ii) das Netzwerk über die Anforderungen der einzelnen Regler in Kenntnis zu setzen. Kern von CoCPN ist eine verteilte Architektur für CPS, welche es den einzelnen Regelkreisen ermöglicht, die verfügbare Kommunikations-Infrastruktur gemeinsam zu nutzen. Im Gegensatz zu den oben genannten Lösungen benötigt CoCPN dafür keine zentrale Instanz mit globaler Sicht auf das Kommunikationssystem, sodass eine enge Kopplung an die Anwendungen vermieden wird. Stattdessen setzt CoCPN auf eine lose Kopplung zwischen Netzwerk und Regelkreisen, realisiert in Form eines Austauschs von Meta-Daten über den sog. *CoCPN-Translator*. CoCPN implementiert ein Staukontrollverfahren, welches den typischen Zusammenhang zwischen erreichbarer Regelgüte und Senderate ausnutzt: die erreichbare Regelgüte steigt mit der Senderate und umgekehrt. Durch Variieren der zu erreichenden Regelgüte kann das Sendeverhalten der Regler so eingestellt werden, dass die vorhandenen Kommunikations-Ressourcen optimal ausgenutzt und gleichzeitig Stausituationen vermieden werden.

In dieser Arbeit beschäftigen wir uns mit den regelungstechnischen Fragestellungen innerhalb von CoCPN. Der Schwerpunkt liegt hierbei auf dem Entwurf und der Analyse von Algorithmen, die

auf Basis der über den CoCPN-Translator ausgetauschten Meta-Daten die notwendige Elastizität liefern und es dadurch den Reglern ermöglichen, schnell auf Änderungen der Netzwerk-Dienstgüte zu reagieren. Dazu ist es notwendig, dass den Reglern ein Modell zur Verfügung gestellt wird, das die Auswirkungen von Verzögerungen und Paketverlusten auf die Regelgüte erfasst.

Im ersten Teil der Arbeit wird eine Erweiterung eines existierenden Modellierungs-Ansatzes vorgestellt, dessen Grundidee es ist, sowohl die Dynamik der Regelstrecke als auch den Einfluss von Verzögerungen und Paketverlusten durch ein *hybrides System* darzustellen. Hybride Systeme zeichnen sich dadurch aus, dass sie sowohl kontinuierlich- als auch diskretwertige Zustandsvariablen besitzen. Unsere vorgestellte Erweiterung ist in der Lage, Änderungen der Netzwerk-Dienstgüte abzubilden und ist nicht auf eine bestimmte probabilistische Darstellung der auftretenden Verzögerungen und Paketverluste beschränkt. Zusätzlich verzichtet unsere Erweiterung auf die in der Literatur übliche Annahme, dass Quittungen für empfangene Datenpakete stets fehlerfrei und mit vernachlässigbarer Latenz übertragen werden. Verglichen mit einem Großteil der verwandten Arbeiten, ermöglichen uns die genannten Eigenschaften daher eine realistischere Berücksichtigung der Netzwerk-Einflüsse auf die Regelgüte.

Mit dem entwickelten Modell kann der Einfluss von Verzögerungen und Paketverlusten auf die Regelgüte präzisiert werden. Auf Basis dieser Prädiktion können Stellgrößen dann mit Methoden der *stochastischen modellprädiktiven Regelung* (*stochastic model predictive control*) berechnet werden. Unsere realistischere Betrachtung der Netzwerk-Einflüsse auf die Regelgüte führt hierbei zu einer gegenseitigen Abhängigkeit von Regelung und Schätzung. Zur Berechnung der Stellgrößen muss der Regler den Zustand der Strecke aus den empfangenen Messungen schätzen. Die Qualität dieser Schätzungen hängt von den berechneten Stellgrößen und deren Auswirkung auf die Regelstrecke ab. Umgekehrt beeinflusst die Qualität der Schätzungen aber maßgeblich die Qualität der Stellgrößen: Ist der Schätzfehler gering, kann der Regler bessere Entscheidungen treffen. Diese gegenseitige Abhängigkeit macht die Berechnung von optimalen Stellgrößen unmöglich und bedingt daher die Fokussierung auf das Erforschen von approximativen Ansätzen.

Im zweiten Teil dieser Arbeit stellen wir zwei neuartige Verfahren für die stochastische modellprädiktive Regelung über Netzwerke vor. Im ersten Verfahren nutzen wir aus, dass bei hybriden Systemen oft sogenannte *multiple model*-Algorithmen zur Zustandsschätzung verwendet werden, welche den geschätzten Zustand in Form einer Gaußmischdichte repräsentieren. Auf Basis dieses Zusammenhangs und einer globalen Approximation der Kostenfunktion leiten wir einen Algorithmus mit geringer Komplexität zur Berechnung eines (suboptimalen) Regelgesetzes her. Dieses Regelgesetz ist nichtlinear und ergibt sich aus der gewichteten Kombination mehrerer unterlagerter Regelgesetze. Jedes dieser unterlagerten Regelgesetze lässt sich dabei als lineare Funktion genau einer der Komponenten der Gaußmischdichte darstellen. Unser zweites vorgestelltes Verfahren besitzt gegensätzliche Eigenschaften. Das resultierende Regelgesetz ist linear und basiert auf einer Approximation der Kostenfunktion, welche wir nur lokal, das heißt nur in der Umgebung einer erwarteten Trajektorie des geregelten Systems, berechnen. Diese Trajektorie wird hierbei durch die Prädiktion einer initialen Zustandsschätzung über den Optimierungshorizont gewonnen. Zur Berechnung des Regelgesetzes schlagen wir dann einen iterativen Algorithmus vor, welcher diese Approximation durch wiederholtes Optimieren der System-Trajektorie verbessert. Simulationsergebnisse zeigen, dass unsere neuartigen Verfahren eine signifikant höhere Regelgüte erzielen können als verwandte Ansätze aus der Literatur.

Der dritte Teil der vorliegenden Arbeit beschäftigt sich erneut mit dem hybriden System aus dem ersten Teil. Die im Rahmen dieser Arbeit verwendeten Netzwerk-Modelle, das heißt die verwendeten probabilistischen Beschreibungen der Verzögerungen und Paketverluste, werden vom CoCPN-Translator auf Grundlage von im Netzwerk gesammelten Status-Informationen erzeugt. Diese Status-Informationen bilden jedoch stets nur Ausschnitte ab und können nie exakt den "Zustand" des Netzwerks repräsentieren. Dementsprechend können die resultierenden Netzwerk-Modelle nicht als fehlerfrei erachtet

werden. In diesem Teil der Arbeit untersuchen wir daher den Einfluss möglicher Fehler in den Netzwerk-Modellen auf die zu erwartende Regelgüte. Weiterhin gehen wir der Frage nach der Existenz von Reglern, die robust gegenüber solchen Fehlern und Unsicherheiten sind, nach. Dazu zeigen wir zunächst, dass sich Fehler in den Netzwerk-Modellen immer als eine polytopische Parameter-Unsicherheit im hybriden System aus dem ersten Teil manifestieren. Für solche polytopischen hybride System leiten wir dann eine sowohl notwendige als auch hinreichende Stabilitätsbedingung her, was einen signifikanten Beitrag zur Theorie der hybriden Systeme darstellt. Die Auswertung dieser Bedingung erfordert es zu bestimmen, ob der *gemeinsame Spektralradius (joint spectral radius)* einer Menge von Matrizen kleiner als eins ist. Dieses Entscheidungsproblem ist bekanntermaßen  $\mathcal{NP}$ -schwer, was die Anwendbarkeit der Stabilitätsbedingung stark limitiert. Daher präsentieren wir eine hinreichende Stabilitätsbedingung, die in polynomieller Zeit überprüft werden kann, da sie auf der Erfüllbarkeit von linearen Matrixungleichungen basiert. Schließlich zeigen wir, dass die Existenz eines Reglers, der die Stabilität des betrachteten polytopischen hybriden Systems garantiert, von der Erfüllbarkeit einer ähnlichen Menge von Matrixungleichungen bestimmt wird. Diese Ungleichungen sind weniger restriktiv als die bisher in der Literatur bekannten, was die Synthese von weniger konservativen Reglern erlaubt.

Schließlich zeigen wir im letzten Teil dieser Arbeit die Anwendbarkeit des kooperativen Konzepts von CoCPN in Simulations-Szenarien, in denen stark ausgelastete Netzwerk-Ressourcen mit anderen Anwendungen geteilt werden müssen. Wir demonstrieren, dass insbesondere das Zusammenspiel unserer modellprädiktiven Verfahren mit dem Staukontrollverfahren von CoCPN einen zuverlässigen Betrieb der Regelkreise ohne unerwünschte Einbußen der Regelgüte auch dann ermöglicht, wenn sich die Kommunikationsbedingungen plötzlich und unvorhergesehen ändern. Insgesamt stellt unsere Arbeit somit einen wichtigen Baustein auf dem Weg zu einem flächendeckenden Einsatz von Standard-Netzen als flexible und adaptive Basis für industrielle CPS dar.



# Abstract

The latest advancements in automation and digitalization have initiated a general transition towards a more intelligent and autonomous operation of manufacturing and production systems that increasingly pushes the traditional boundaries between the physical processes and the software components used for sensing, monitoring, and actuation. *Cyber-physical systems (CPS)* play a leading role in this transition as they tightly integrate physical and software (cyber) components that share a communication network for the exchange of system states, sensor readings, and control commands. More and more CPS are deployed with off-the-shelf networking equipment to benefit from the availability of cheap devices and the option to re-use existing infrastructure. However, general-purpose networks based on the IEEE 802.3 (Ethernet) and IEEE 802.11 (WLAN) standards are characterized by fluctuations of the provided quality of service (QoS), which result in changing communication conditions that become visible in the form of varying latencies, packet loss rates, and achievable data rates for all end systems that share the communication resources.

Control loops that use a shared communication system to realize the communication between sensors, actuators, and controllers are called *networked control systems (NCS)*. They usually have strict communication requirements and demand data transmissions with guaranteed latencies, rendering them particularly vulnerable to fluctuations of the network QoS. State-of-the-art approaches ensure a reliable operation without performance degradation by providing communication conditions that meet the imposed requirements. However, this usually necessitates the deployment of highly specialized networks such as fieldbuses, the use of complex, made-to-measure communication protocols such as the time-sensitive networking (TSN) standards, or demands central entities with a global view on the communication system, rendering these approaches expensive and relatively inflexible.

The present thesis covers results of the interdisciplinary research project *CoCPN: Cooperative Cyber-Physical Networking* that pursues a different concept and specifically targets CPS deployed with standard networking equipment. CoCPN implements a novel, cooperative approach to make the control loops within such CPS *elastic*, that is, to increase their flexibility to cope with changing communication conditions, and to render the communication system aware of the communication requirements of the controllers. The cornerstone of the approach is a novel, distributed architecture for CPS that allows the control loops to cooperatively share the available communication capacity. This architecture forgoes a central entity with a global view on the communication resources, thereby avoiding tight coupling between the communication infrastructure and the control loops that sit atop. Instead, CoCPN introduces the *CoCPN translator* that keeps the implemented cooperation lightweight and distributed by enabling the exchange of meta data between the communication system and the control loops. The CoCPN architecture is motivated by the typical connection between achievable control performance and sending rate – the control performance improves if the controller increases its sending rate and vice versa. CoCPN leverages this connection and dynamically adjusts the target control performance that the controllers shall achieve, which in turn adapt their sending rates as much as is needed to achieve this target performance. Based on this connection, CoCPN implements a control-aware congestion control mechanism to realize a fair sharing of the communication resources

that avoids potentially severe degradations of the network QoS and keeps the performance of all control loops in balance.

The present thesis centers around the “control-related portion” of CoCPN. More specifically, our focus is on the design and analysis of algorithms that yield the desired elasticity and, based on the information exchange provided by the CoCPN translator, enable networked controllers to quickly respond to variations of the network QoS and to efficiently use the available communication capacity. To achieve this goal, controllers must not only be aware of the shared communication system but also need a model that accurately describes the impact of its most relevant influencing factors, namely the impact of packet delays and losses.

The first part of this thesis develops an extension of an existing modeling approach for the design and analysis of networked control algorithms that expresses the plant dynamics and the impact of the packet delays and losses in terms of a single *hybrid system*. Hybrid systems are dynamical systems with continuous- and discrete-valued state variables. The developed extension is flexible enough to reflect changes of the network QoS, supports different probabilistic descriptions of the occurring packet delays and losses, and does not rely on the typical assumption that acknowledgment packets are transmitted instantaneously and received with negligible communication delay. Thanks to these properties, we are able to model the influence of the shared communication system more realistically compared to the vast majority of the literature.

Equipped with the developed hybrid system model, controllers can predict the influence of packet delays and losses on the control performance, rendering *stochastic model predictive control (SMPC)* well-suited to compute the control inputs. To that end, controllers must estimate the plant state based on the information provided by the received measurements. However, our pursued more realistic consideration of packet delays and losses results in an interdependency between decision-making and state estimation: Each computed control input does not only affect the plant state, which is the desired control action, but also the quality of the controller’s future state estimates, which in turn determine the quality of future control inputs – when the estimation error is low, better decisions can be made. This interdependency is known as *dual effect* and prevents the computation of control inputs that are optimal with regards to a selected cost function. Hence, any practical SMPC approach is necessarily an approximation.

In the second part of this thesis, we propose two novel approaches for the computation of suboptimal control laws. The first approach exploits that estimates of the plant state in networked control algorithms are often provided by *multiple model* estimation algorithms. We use this connection to derive a control algorithm with low computational complexity that combines the output of multiple individual controllers, each of which corresponds to one of the models used by the estimator. The underlying control law is nonlinear, relies on a global approximation of the cost function, and neglects the contribution of the dual effect on the cost. Our second proposed approach possesses contrasting properties. The underlying control law is linear and minimizes a local approximation of the cost via repeated improvements of a given reference trajectory, which is obtained by propagating an initial state estimate over the optimization horizon. Moreover, the approach takes the influence of the dual effect on the cost into account. In evaluation scenarios, we show that our novel approaches can achieve significantly better control performance than state-of-the-art networked SMPC approaches from the literature.

In the third part of this thesis, we return our attention to the hybrid system model developed in the first part. The different probabilistic descriptions of the packet delays and losses supported by this model are provided by the CoCPN translator, which creates them by translating monitoring information collected inside the communication system into appropriate network models. The collected monitoring data can only provide *snapshots* of the past so that the resulting network models are always estimates

of the “state” of the communication system. Consequently, questions about the influence of modeling errors on the achievable control performance and the existence of controllers that are robust to such errors arise. We first show that uncertain or even completely unknown network models always lead to a polytopic parameter uncertainty in the hybrid system model. Subsequently, we derive a necessary and sufficient condition for the mean square stability of such polytopic systems, which constitutes a substantial contribution to the theory of hybrid systems. The condition demands verifying whether the *joint spectral radius* of a set of matrices is less than one, which, unfortunately, is known to be  $\mathcal{NP}$ -hard, thereby limiting the practical applicability of the derived condition. For this reason, we propose a sufficient stability condition that requires to test the feasibility of a set of *linear matrix inequalities*, which is typically done by state-of-the-art solvers in polynomial time. A by-product of the derived condition is a similar set of inequalities whose feasibility guarantees the existence of a state feedback controller that stabilizes the plant in the mean square sense. The set of inequalities is less restrictive than related ones from the literature and, thus, enables the synthesis of less conservative controllers.

In the final part of this thesis, we integrate our contributions into the CoCPN architecture and showcase the applicability of its cooperative concept in simulation scenarios with highly utilized network resources that have to be shared with unrelated traffic from other applications. Especially the collaboration between our SMPC approaches and the control-aware congestion control allows control loops to operate reliably without undesirable performance degradations even when the communication conditions change rapidly. Consequently, our work paves the way for the ubiquitous use of standard networking equipment as a flexible and adaptable foundation for CPS in industrial applications.



# Notation

## Symbols

	CoCPN translator.
$\mathbb{R}$	Set of real numbers.
$\mathbb{Q}$	Set of rational numbers.
$\mathbb{N}$	Set of natural numbers.
$\mathbb{R}^n$	$n$ -dimensional Euclidean space.
$\mathbb{R}^{n \times m}$	Vector space of all $n$ -by- $m$ -dimensional matrices.
$\Delta^n$	$n$ -dimensional standard/probability simplex.
$\mathcal{S}^n$	$n$ -dimensional simplex in $\mathbb{R}^m$ .
$\mathbb{H}^n$	Banach space of all $(N + 1)$ -tuples of $n$ -by- $n$ -dimensional matrices.
$\delta(\cdot)$	Dirac delta function.
$\mathbb{1}_{\{a\}}$	Indicator function of the predicate $a$ .
$n!$	Factorial of the nonnegative integer $n$ .
$\text{conv}(\mathcal{A})$	Convex hull of the set $\mathcal{A}$ .
$a_{i:j}$	Sequence $a_i, a_{i+1}, \dots, a_j$ of elements of the same kind (e.g., vectors or matrices).
$\{a_k\}$	Infinite sequence $a_0, a_1, \dots$ of elements of the same kind (e.g., vectors or matrices).
$(a_1, \dots, a_n)$	Tuple, i.e., an ordered list, of $n$ elements of the same kind (e.g., vectors or matrices).
$\dot{a}$	First derivative of $a$ with respect to time.
$\ddot{a}$	Second derivative of $a$ with respect to time.
$\square$	End of proof.

## Special Vectors

$\underline{0}$	Zero vector of arbitrary dimension.
$\underline{1}$	Vector of ones of arbitrary dimension.
$\underline{1}_n$	$n$ -dimensional vector of ones.
$e_n^{(i)}$	$i$ -th standard basis vector in $\mathbb{R}^n$ .

## Operations on Vectors

$\underline{x}^T$	Transpose of the vector $\underline{x}$ .
$\ \underline{x}\ _2$	Euclidean norm of the vector $\underline{x}$ .
$\ \underline{x}\ $	Any norm of the vector $\underline{x}$ .
$\underline{x} \otimes \underline{y}$	Kronecker product of the vectors $\underline{x}$ and $\underline{y}$ .
$\underline{x} > 0$	All entries of the vector $\underline{x}$ are positive.
$\underline{x} \geq 0$	All entries of the vector $\underline{x}$ are nonnegative.
$n_x$	Dimension of the vector $\underline{x}$ , i.e., its number of entries.
$E\{\underline{x}\}$	Mean/first moment of the random vector $\underline{x}$ .

## Special Matrices

$\mathbf{0}$	Zero matrix of arbitrary dimension, not necessarily square.
$\mathbf{0}_{n \times m}$	$n$ -by- $m$ zero matrix.
$\mathbf{0}_n$	$n$ -by- $n$ zero matrix.
$\mathbf{1}_n$	$n$ -by- $n$ matrix of ones.
$\mathbf{I}$	Identity matrix of arbitrary dimension.
$\mathbf{I}_n$	$n$ -by- $n$ identity matrix.

## Operations on Matrices

$\mathbf{A}^T$	Transpose of the matrix $\mathbf{A}$ .
$\mathbf{A}^{-1}$	Inverse of the matrix $\mathbf{A}$ .
$\mathbf{A}^\dagger$	Moore-Penrose pseudoinverse of the matrix $\mathbf{A}$ .
$\text{tr}[\mathbf{A}]$	Trace of the matrix $\mathbf{A}$ .
$\det[\mathbf{A}]$	Determinant of the matrix $\mathbf{A}$ .
$\ \mathbf{A}\ _F$	Frobenius norm of the matrix $\mathbf{A}$ .
$\ \mathbf{A}\ $	Any norm of the matrix $\mathbf{A}$ .
$\mathbf{A} \otimes \mathbf{B}$	Kronecker product of the matrices $\mathbf{A}$ and $\mathbf{B}$ .
$\text{vec}(\mathbf{A})$	Vectorization of the matrix $\mathbf{A}$ .
$\mathbf{A} \succ 0$	Matrix $\mathbf{A}$ is positive definite.
$\mathbf{A} \succeq 0$	Matrix $\mathbf{A}$ is positive semidefinite.
$\mathbf{A} > 0$	All entries of the matrix $\mathbf{A}$ are positive.
$\mathbf{A} \geq 0$	All entries of the matrix $\mathbf{A}$ are nonnegative.
$\rho(\mathbf{A})$	Spectral radius of the square matrix $\mathbf{A}$ .
$\hat{\rho}(\mathcal{M})$	Joint spectral radius of a finite set of square matrices $\mathcal{M}$ .
$\Pi_k(\mathcal{M})$	All products of length $k$ whose factors are from a set of square matrices $\mathcal{M}$ .
$\ \mathcal{X}\ _{\mathbb{H}}$	Norm on $\mathbb{H}^n$ .
$\hat{\mathcal{X}}(\mathcal{X})$	Vectorization of the element $\mathcal{X} \in \mathbb{H}^n$ .
$\mathcal{E}^{(i)}(\mathcal{X})$	Family of mappings on $\mathbb{H}^n$ , defined as $\mathcal{E}^{(i)}(\mathcal{X}) = \sum_{j=0}^{N+1} t_{ij} \mathbf{X}^{(j)}$ .

## Variable Conventions

$\text{ACK}_k$	Acknowledgment sent by the actuator at time $k$ .
$a_k(\underline{U}_{k'})$	Age (in time steps) of the control sequence $\underline{U}_{k'}$ at time $k$ .
$\mathbf{A}$	System matrix in the discrete-time plant dynamics.
$\mathbf{A}_c$	System matrix in the continuous-time plant dynamics.

$\tilde{\mathbf{A}}^{(i)}$	System matrix in the dynamics of $\underline{\psi}_k$ associated with mode $\theta_k = i$ .
$\overline{\mathbf{A}}_k^{(i)}$	System matrix in the dynamics of $\underline{\xi}_k$ associated with mode $\theta_k = i$ at time $k$ .
$\mathcal{A}_k$	Set of acknowledgments received by the controller at time $k$ .
$\mathbf{B}$	Input matrix in the discrete-time plant dynamics.
$\mathbf{B}_c$	Input matrix in the continuous-time plant dynamics.
$\tilde{\mathbf{B}}^{(i)}$	Input matrix in the dynamics of $\underline{\psi}_k$ associated with mode $\theta_k = i$ .
$\overline{\mathbf{B}}_k^{(i)}$	Input matrix in the dynamics of $\underline{\xi}_k$ associated with mode $\theta_k = i$ at time $k$ .
$\mathbf{C}$	Measurement matrix in the measurement model.
$\overline{\mathbf{C}}$	Measurement matrix in the measurement model of $\underline{\xi}_k$ at time $k$ .
$f_a$	Controller sampling rate (in hertz).
$\mathbf{F}$	System matrix in the dynamics of $\underline{\eta}_k$ .
$\mathbf{G}$	Input matrix in the dynamics of $\underline{\eta}_k$ .
$\mathbf{G}_c$	System noise matrix in the continuous-time plant dynamics.
$\mathcal{I}_k$	Information set available to the controller at time $k$ .
$\mathcal{J}$	Cost function to be minimized over the optimization horizon.
$K$	Length of the optimization horizon (in time steps).
$K_s$	Duration of a simulation run (in time steps).
$\mathbf{K}_k$	Observer gain of a linear control law at time $k$ .
$L$	The maximum number of stored measurements.
$\mathbf{L}_k$	Controller gain of a linear control law at time $k$ .
$\mathcal{L}$	Set of all vertices of the transition matrix polytope.
$M^{\text{ac}}$	Maximum value of $\tau_k^{\text{ac}}$ .
$M$	Maximum value of $\tau_k^{\text{ca}}$ .
$M^{\text{sc}}$	Maximum value of $\tau_k^{\text{sc}}$ .
$N$	Number of control inputs in a control sequence.
$N_s$	Number of simulation runs in a Monte Carlo simulation.
$\mathbf{N}_t$	Covariance matrix of $\underline{o}_t$ at stage $t$ of the optimization horizon.
$\underline{o}_t$	System noise in the dynamics of $\tilde{\mathbf{x}}_t$ at stage $t$ of the optimization horizon.
$p_{ij}^{\text{ca}}$	Transition probability $\text{P}[\tau_k^{\text{ca}} = j   \tau_{k-1}^{\text{ca}} = i]$ .
$p_k^{(i)}$	Probability $\text{P}[\tau_k^{\text{ca}} = i]$ , the $i$ -th element of $\underline{p}_k$ .
$\underline{p}_k$	Probability distribution of $\tau_k^{\text{ca}}$ at time $k$ .
$\mathbf{P}^{\text{ca}}$	Transition matrix of $\tau_k^{\text{ca}}$ .
$\tilde{\mathbf{P}}^{\text{ca}}$	Transition matrix of $\tau_k^{\text{ca}}$ .
$\tilde{\mathbf{P}}_t$	Costate matrix associated with $\tilde{\mathbf{X}}_t$ at stage $t$ of the optimization horizon.
$\tilde{\mathbf{P}}_t^{(i)}$	Costate matrix associated with $\tilde{\mathbf{X}}_t^{(i)}$ at stage $t$ of the optimization horizon.
$\tilde{\mathcal{P}}_t$	Tuple composed of costate matrices $\tilde{\mathbf{P}}_t^{(i)}$ at stage $t$ of the optimization horizon.
$\underline{\mathbf{P}}_t^{(i)}$	Costate matrix associated with $\underline{\mathbf{X}}_t^{(i)}$ at stage $t$ of the optimization horizon.
$\overline{\mathbf{P}}_t^{(i)}$	Costate matrix associated with $\overline{\mathbf{X}}_t^{(i)}$ at stage $t$ of the optimization horizon.
$\mathbf{Q}_t$	State weighting matrices in the cost function.
$R$	Number of vertices of the transition matrix polytope.
$\mathbf{R}_t$	Input weighting matrices in the cost function.
$t_a$	Controller sampling interval, $t_a = 1/f_a$ (in seconds).
$t_s$	Duration of a simulation run (in seconds).
$t_{k,ij}$	Transition probability from mode $i$ to mode $j$ at time $k$ , i.e., $\text{P}[\theta_{k+1} = j   \theta_k = i]$ .
$\mathbf{T}_k$	Transition matrix of $\theta_k$ .
$\underline{u}_k$	Control input applied by the actuator at time $k$ .
$\underline{u}_{k+i k}$	Control input computed at time $k$ for application at time $k+i$ .

$\underline{u}_k^{\text{df}}$	Default input applied by the actuator at time $k$ if no applicable inputs are available.
$\underline{U}_k^{\text{bf}}$	Control sequence buffered by the actuator at time $k$ .
$\underline{U}_k$	Control sequence computed at time $k$ .
$\bar{\mathbf{U}}$	Distribution matrix in the lumped dynamics for $\theta_k$ .
$\mathcal{U}_k$	Set of control sequences received by the actuator at time $k$ .
$\underline{v}_k$	Measurement noise at time $k$ .
$\bar{\underline{v}}_k$	Measurement noise affecting $\bar{\underline{y}}_k$ at time $k$ .
$\mathbf{V}$	Measurement noise covariance matrix.
$\bar{\mathbf{V}}$	Collection matrix in the lumped dynamics for $\theta_k$ .
$\bar{\mathbf{V}}$	Covariance matrix of $\bar{\underline{v}}_k$ .
$\mathcal{V}_t$	Cost-to-go from stage $t$ of the optimization horizon to the terminal stage $K$ .
$\mathcal{V}_t^{(i)}$	Mode-conditioned cost-to-go from stage $t$ of the optimization horizon to the terminal stage $K$ .
$\bar{\mathcal{V}}_t$	Upper bound for $\mathcal{V}_t$ .
$\underline{w}_k$	System noise at time $k$ .
$\tilde{\underline{w}}_k$	System noise in the dynamics of $\underline{\psi}_k$ at time $k$ .
$\bar{\underline{w}}_k$	System noise in the dynamics of $\underline{\xi}_k$ at time $k$ .
$\mathbf{W}$	System noise covariance matrix.
$\mathbf{W}_c$	Intensity of the zero-mean white noise in the continuous-time plant dynamics.
$\tilde{\mathbf{W}}$	Covariance matrix of $\tilde{\underline{w}}_k$ .
$\bar{\mathbf{W}}$	Covariance matrix of $\bar{\underline{w}}_k$ .
$\underline{x}_k$	Plant state at time $k$ .
$\hat{\underline{x}}_0$	Mean of initial plant state $\underline{x}_0$ .
$\tilde{\underline{x}}_t$	Closed-loop state at stage $t$ of the optimization horizon, defined as $\tilde{\underline{x}}_t = \begin{bmatrix} \hat{\underline{\xi}}_t \\ \hat{\underline{\xi}}_t^{\text{T}} \end{bmatrix}^{\text{T}}$ .
$\mathbf{X}_0$	Covariance of initial plant state $\underline{x}_0$ .
$\tilde{\mathbf{X}}_t$	Second moment of $\tilde{\underline{x}}_t$ at stage $t$ of the optimization horizon, defined as $\tilde{\mathbf{X}}_t = \text{E}\{\tilde{\underline{x}}_t \tilde{\underline{x}}_t^{\text{T}}\}$ .
$\tilde{\mathbf{X}}_t^{(i)}$	Second moment of $\tilde{\underline{x}}_t$ at stage $t$ of the optimization horizon, defined as $\tilde{\mathbf{X}}_t^{(i)} = \text{E}\{\tilde{\underline{x}}_t \tilde{\underline{x}}_t^{\text{T}} \mathbb{1}_{\{\theta_t=i\}}\}$ .
$\underline{\mathbf{X}}_k^{(i)}$	Second moment of $\hat{\underline{\xi}}_k$ at time $k$ , defined as $\underline{\mathbf{X}}_k^{(i)} = \text{E}\{\hat{\underline{\xi}}_k \hat{\underline{\xi}}_k^{\text{T}} \mathbb{1}_{\{\theta_k=i\}}\}$ .
$\bar{\mathbf{X}}_k^{(i)}$	Estimation error covariance at time $k$ , defined as $\bar{\mathbf{X}}_k^{(i)} = \text{E}\left\{\left(\underline{\xi}_k - \hat{\underline{\xi}}_k\right) \left(\underline{\xi}_k - \hat{\underline{\xi}}_k\right)^{\text{T}} \mathbb{1}_{\{\theta_k=i\}}\right\}$ .
$\underline{y}_k$	Measurement taken by the sensor at time $k$ .
$\bar{\underline{y}}_k$	Augmented measurement at time $k$ , defined as $\bar{\underline{y}}_k = \begin{bmatrix} \gamma_{k k} \underline{y}_k^{\text{T}} & \gamma_{k k-1} \underline{y}_{k-1}^{\text{T}} & \cdots & \gamma_{k k-L} \underline{y}_{k-L}^{\text{T}} \end{bmatrix}^{\text{T}}$ .
$\mathcal{Y}_k$	Set of measurements received by the controller at time $k$ .
$\underline{z}_k$	Vectorized tuple of second moments at time $k$ , defined as $\underline{z}_k = \hat{\underline{\varphi}}(\mathcal{Z}_k)$ .
$\mathcal{Z}_k$	Tuple composed of second moments $\Psi_k^{(i)}$ at time $k$ .
$\gamma_{k k-l}$	Binary variable encoding availability of measurement $\underline{y}_{k-l}$ at time $k$ .
$\underline{\gamma}_k$	Binary vector encoding availability of measurements $\underline{y}_k, \underline{y}_{k-1}, \dots, \underline{y}_{k-l}$ at time $k$ .
$\mathbf{\Gamma}_t^{(i)}$	System matrix in the dynamics of $\tilde{\underline{x}}_t$ associated with mode $\theta_t = i$ at stage $t$ .
$\delta$	Upper bound for $t_{k,NN}$ , i.e., $\text{P}[\theta_{k+1} = N   \theta_k = N] \leq \delta < 1$ .
$\underline{\eta}_k$	All inputs from past control sequences that are applicable at time $k$ or later.
$\theta_k$	Markov chain describing the mode dynamics in the augmented dynamical system.
$\mathbf{\Lambda}^{(i)}$	$i$ -th vertex of the transition matrix polytope.

$\mu_k^{(i)}$	Probability $P[\theta_k = i]$ , the $i$ -th element of $\underline{\mu}_k$ .
$\underline{\mu}_k$	Probability distribution of $\theta_k$ at time $k$ .
$\underline{\xi}_k$	Augmented state at time $k$ , defined as $\underline{\xi}_k = [\underline{x}_k^T \ \underline{x}_{k-1}^T \ \cdots \ \underline{x}_{k-L}^T \ \underline{\eta}_k^T]^T$ .
$\underline{\hat{\xi}}_k$	Controller state, i.e., its estimate of $\underline{\xi}_k$ , at time $k$ .
$\underline{\pi}_k$	Control law at time $k$ .
$\tau_k^{\text{ac}}$	Stochastic process to model delays (in time steps) and losses in the transmission of acknowledgments, defined on the set $\{0, 1, \dots, M^{\text{ac}}\}$ .
$\tau_k^{\text{ca}}$	Stochastic process to model delays (in time steps) and losses in the transmission of control sequences, defined on the set $\{0, 1, \dots, M\}$ .
$\tau_k^{\text{sc}}$	Stochastic process to model delays (in time steps) and losses in the transmission of measurements, defined on the set $\{0, 1, \dots, M^{\text{sc}}\}$ .
$\underline{\tau}_k^{\text{ca}}$	Augmented stochastic process, defined as $\underline{\tau}_k^{\text{ca}} = (\tau_k^{\text{ca}}, \tau_{k-1}^{\text{ca}}, \dots, \tau_{k-(N-1)}^{\text{ca}})$ .
$\underline{\psi}_k$	Augmented state at time $k$ , defined as $\underline{\psi}_k = [\underline{x}_k^T \ \underline{\eta}_k^T]^T$ .
$\Psi_k$	Second moment of $\underline{\psi}_k$ at time $k$ , defined as $\Psi_k = E\{\underline{\psi}_k \underline{\psi}_k^T\}$ .
$\Psi_k^{(i)}$	Second moment of $\underline{\psi}_k$ at time $k$ , defined as $\Psi_k^{(i)} = E\{\underline{\psi}_k \underline{\psi}_k^T \mathbf{1}_{\{\theta_k=i\}}\}$ .

## Abbreviations

ACK	Acknowledgment
CAN	Controller Area Network
CoCC	Cooperative Congestion Control
CoCPN	CoCPN: Cooperative Cyber-Physical Networking
CPS	Cyber-Physical System
DiffServ	Differentiated Services
DP	Dynamic Programming
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMM	Interacting Multiple Model
IP	Internet Protocol
JSR	Joint Spectral Radius
LMI	Linear Matrix Inequality
MJLS	Markov Jump Linear System
MMSE	Minimum Mean Square Error
MPC	Model Predictive Control
MSS	Mean Square Stable
MTU	Maximum Transmission Unit
NCS	Networked Control System
NTP	Network Time Protocol
PDF	Probability Density Function
PPP	Point-to-Point Protocol
PTP	Precision Time Protocol
OoC	Quality of Control
QoS	Quality of Service
SMPC	Stochastic Model Predictive Control
TCP	Transmission Control Protocol
TSN	Time-Sensitive Networking
UDP	User Datagram Protocol



# Contents

<b>Acknowledgment</b>	<b>III</b>
<b>Kurzfassung</b>	<b>V</b>
<b>Abstract</b>	<b>IX</b>
<b>Notation</b>	<b>XIII</b>
<b>Contents</b>	<b>XIX</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Outline and Contributions . . . . .	3
1.1.1 Model for Design and Analysis of Networked Control Algorithms . . . . .	4
1.1.2 Networked Stochastic Model Predictive Control . . . . .	4
1.1.3 Networked Stochastic Robust Control . . . . .	5
1.2 Related Work . . . . .	6
<b>2 CoCPN: Cooperative Cyber-Physical Networking</b>	<b>7</b>
2.1 The CoCPN Architecture . . . . .	8
2.2 CoCPN-Sim at a Glance . . . . .	12
<b>3 Description of the Considered Networked Control System</b>	<b>15</b>
3.1 Formalization of the NCS . . . . .	17
3.2 Properties of the Network Models . . . . .	19
3.2.1 Independent Packet Delays and Losses . . . . .	20
3.2.2 Correlated Packet Delays and Losses . . . . .	20
<b>4 Construction of the Augmented Dynamics</b>	<b>23</b>
4.1 Definition of the Augmented State . . . . .	24
4.2 Dynamics of $\theta_k$ : Independent Packet Delays and Losses . . . . .	27
4.3 Dynamics of $\theta_k$ : Correlated Packet Delays and Losses . . . . .	28
4.4 Discussion of the Modeling Approach . . . . .	32
4.5 Conclusions . . . . .	34
<b>5 Sequence-Based Stochastic Model Predictive Control</b>	<b>37</b>
5.1 Problem Formulation . . . . .	38
5.2 Intractability of the Optimal Solution . . . . .	41
5.3 Multiple Model Based SMPC . . . . .	45
5.3.1 Derivation of the Control Laws . . . . .	45
5.3.2 Tailoring of a Multiple Model Estimator . . . . .	51

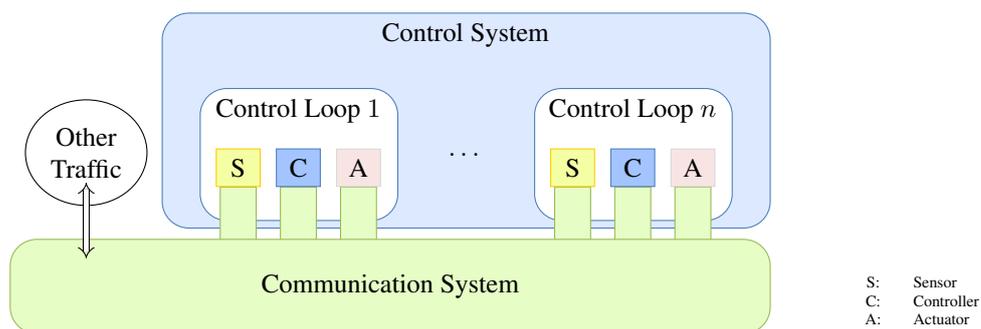
5.3.3	Summary of the Proposed SMPC Algorithm . . . . .	54
5.4	SMPC Based on Local Approximation of the Cost-to-Go . . . . .	54
5.4.1	Definition of the Augmented Dynamical System . . . . .	55
5.4.2	Reformulation of the Cost Function . . . . .	57
5.4.3	Derivation of the Control Laws . . . . .	61
5.4.4	Summary of the Proposed SMPC Algorithm . . . . .	70
5.5	Evaluation . . . . .	71
5.5.1	Control of a Double Integrator . . . . .	72
5.5.2	Control of an Overhead Crane . . . . .	75
5.6	Conclusions . . . . .	78
<b>6</b>	<b>Sequence-Based Stochastic Robust Control</b>	<b>81</b>
6.1	The Joint Spectral Radius . . . . .	83
6.2	Derivation of the Polytopic MJLS . . . . .	84
6.2.1	Polytopic MJLS: Independent Packet Delays and Losses . . . . .	85
6.2.2	Polytopic MJLS: Correlated Packet Delays and Losses . . . . .	87
6.3	A Necessary and Sufficient Stability Condition . . . . .	89
6.4	Sufficient Conditions for Stability and Stabilizability . . . . .	92
6.5	Numerical Example . . . . .	95
6.6	Conclusions . . . . .	96
<b>7</b>	<b>Integration into the CoCPN Architecture</b>	<b>99</b>
7.1	Description of the Considered Control Task . . . . .	100
7.2	Implementation of the CoCPN Translator Interface . . . . .	101
7.3	Simulation Scenarios . . . . .	106
7.3.1	Multiple NCS with UDP Cross Traffic . . . . .	108
7.3.2	Single NCS with Mixed TCP/UDP Cross Traffic . . . . .	113
7.4	Conclusions . . . . .	116
<b>8</b>	<b>Conclusions</b>	<b>117</b>
8.1	Summary . . . . .	117
8.2	Future Research . . . . .	119
<b>A</b>	<b>Mathematical Concepts and Results</b>	<b>123</b>
A.1	Metric Space Concepts . . . . .	123
A.2	Norms and Banach Spaces . . . . .	124
A.3	Kronecker Product and Vectorization . . . . .	126
A.4	Positive Definite and Positive Semidefinite Matrices . . . . .	128
A.5	The Moore-Penrose Pseudoinverse . . . . .	129
A.6	The Perron-Frobenius Theorem and Stochastic Matrices . . . . .	130
A.7	The Banach Space of Tuples of Square Matrices . . . . .	131
<b>B</b>	<b>Proofs of the Results in Chapter 4</b>	<b>133</b>
B.1	Proof of Theorem 4.1 . . . . .	133
B.2	Proof of Lemma 4.1 . . . . .	134
B.3	Proof of Theorem 4.2 . . . . .	136
<b>C</b>	<b>Proofs of the Results in Chapter 5</b>	<b>137</b>
C.1	Proof of Lemma 5.1 . . . . .	137
C.2	Proof of Lemma 5.2 . . . . .	138

C.3	Proof of Theorem 5.1 . . . . .	139
C.4	Proof of Lemma 5.3 . . . . .	141
C.5	Proof of Theorem 5.2 . . . . .	142
C.6	Proof of Lemma 5.4 . . . . .	143
C.7	Proof of Theorem 5.3 . . . . .	144
<b>D</b>	<b>Proofs of the Results in Chapter 6</b>	<b>147</b>
D.1	Proof of Lemma 6.1 . . . . .	147
D.2	Proof of Lemma 6.2 . . . . .	148
D.3	Proof of Theorem 6.5 . . . . .	149
D.4	Proof of Theorem 6.6 . . . . .	150
D.5	Proof of Theorem 6.7 . . . . .	152
D.6	Proof of Theorem 6.8 . . . . .	155
<b>E</b>	<b>Equations of Motion of the Double Inverted Pendulum</b>	<b>159</b>
<b>F</b>	<b>Parameters used in the Simulations and Numerical Examples</b>	<b>163</b>
F.1	Parameters used in the Simulations in Section 5.5 . . . . .	163
F.1.1	Parameters used in Section 5.5.1 . . . . .	163
F.1.2	Parameters used in Section 5.5.2 . . . . .	164
F.2	Parameters used in the Numerical Example in Section 6.5 . . . . .	164
F.3	Parameters used in in Chapter 7 . . . . .	165
F.3.1	Parameters of the Double Inverted Pendulum . . . . .	165
F.3.2	Parameters of the Network . . . . .	165
F.3.3	Parameters of the Controllers . . . . .	166
	<b>List of Algorithms</b>	<b>169</b>
	<b>List of Figures</b>	<b>171</b>
	<b>List of Tables</b>	<b>173</b>
	<b>Bibliography</b>	<b>175</b>
	<b>Own Publications</b>	<b>195</b>
	<b>(Co-) Supervised Student Theses</b>	<b>197</b>



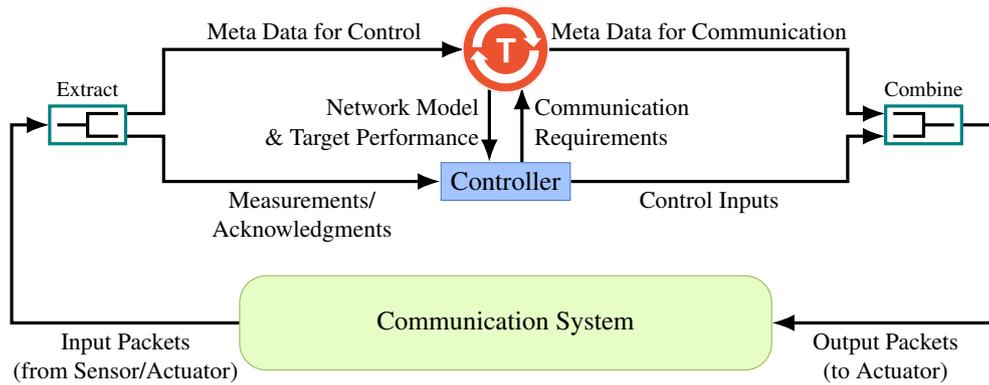
# Introduction

In recent years, the traditional line in manufacturing and production between the physical processes and higher-level services has become more and more blurred. This trend is part of a general transition towards a more intelligent, flexible, and autonomous operation of manufacturing and production systems, or power and water distribution systems [1–4]. This transition, often referred to by the umbrella terms *Industry 4.0* and *Industrial Internet of Things*, is driven by the recent advancements in automation and digitalization and the desire to integrate higher-level (cloud-based) services for, e.g., lifecycle management or data analytics and visualization [5–7]. *Cyber-physical systems (CPS)* are widely considered a cornerstone for a successful realization of this transition, as they tightly integrate physical and software (cyber) components into a single distributed system [8, 9]. Conceptually, a CPS can be viewed as an ensemble of control loops that share a communication system to exchange sensor readings, control commands, and system or environmental states for sensing, monitoring, and actuation of physical components [10, 11]. This is illustrated in Fig. 1.1.



**Figure 1.1:** High level view of a cyber-physical system (adapted from own publication [290]).

Control loops that use a shared communication system are called *networked control systems (NCS)* to dissociate them from “traditional” control loops, where dedicated point-to-point wiring is employed to connect sensors, actuators, and controllers [12, 13]. In industrial and automotive control applications, highly specialized wired networks, so-called *fieldbuses*, have been standard to date. For instance,



**Figure 1.2:** Illustration of the data exchange provided by the CoCPN translator  (adapted from own publication [290]).

in 2018 the number of installed Profibus devices reached the mark of 60 million [14]. Other common fieldbuses are FlexRay [15], EtherCAT [16], and solutions based on Controller Area Network (CAN), such as CANopen [17] or DeviceNet [18]. Properly designed fieldbus networks provide data transmissions with guaranteed latencies and offer simplified installation and maintenance compared to the traditional point-to-point wiring [13, 19]. Similarly, specialized wireless networks based on the IEEE 802.15.4 standard, such as WirelessHART or ISA100.11a, have found widespread usage in industrial environments since they also offer guaranteed latencies [20–22].

Lately, however, more and more CPS are deployed with off-the-shelf networking equipment. This development is boosted by the availability of cheap devices, the ubiquitous presence of general-purpose networks (e.g., the Internet), which offers the option to use already existing communication infrastructure, and a gain in flexibility compared to fieldbuses [23]. Specifically devices that support the IEEE 802.11 (WLAN) standards allow for significantly higher data rates compared to the wireless technologies mentioned above, whose underlying physical layers constrain their reliable operation to applications with sampling periods above 50 ms [24–26]. On the other hand, the quality of service (QoS) provided by general-purpose networks is subject to fluctuations. These occur due to the co-existence of data flows with different priorities, queuing at intermediate nodes that emerges from the lack of any statically pre-determined scheduling, and protocol mechanisms for medium access or flow and congestion control. For the end systems, variations of the QoS become visible in terms of varying latencies, packet loss rates, and goodput. These factors can severely degrade the achievable performance of control loops and, even worse, render the closed-loop dynamics unstable [27]. Consequently, for a reliable operation within such CPS, control loops must be *elastic*, that is, they must be enabled to operate reliably even under changing communication conditions.

The interdisciplinary research project *CoCPN: Cooperative Cyber-Physical Networking* achieves this goal by implementing cooperation between control loops and the shared communication system. The key idea is to cooperatively share the available communication resources by balancing the performance that each control loop shall achieve. To avoid tight coupling between the control loops and the communication system, CoCPN uses a flexible, distributed, and lightweight cooperation in the form of a data exchange, which is illustrated in Fig. 1.2. To interact with the communication system, each controller is equipped with a so-called *CoCPN translator* , which provides the formal interfaces for the meta data passed from the controller to the communication system and vice versa. The CoCPN translator supplies each controller with information about the QoS that the communication system currently provides. To that end, it processes monitoring data collected inside the communication system and translates it into a suitable representation in the form of a network model. The network model enables the controller to take quantities such as expected packet delays or loss rates into

account when computing new control inputs. Additionally, each controller is provided with a target control performance and adapts its usage of the communication resources accordingly by decreasing or increasing its sending rate. The target performance is computed by a control-aware congestion control mechanism, which is part of the communication system, such that all control loops can achieve the same performance. This does usually not lead to equal sending rates for all controllers, since the individual communication requirements, for instance regarding the timely transmission of control commands, are dependent on the control task. Hence, to compute the target control performance, the communication system uses the CoCPN translator to obtain information on the communication requirements of a controller.

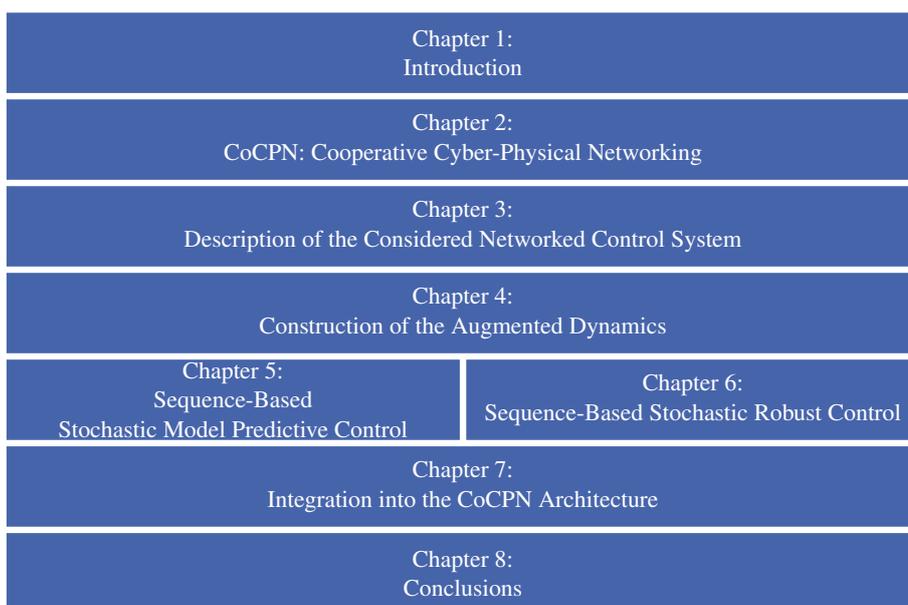
Within CoCPN, the goal of this thesis is the realization of the “control-related portion” of the cooperation. This comprises, first, the design and the implementation of the related parts of the CoCPN translator and its interfaces for the interaction with the communication system, and, second, the provisioning and processing of the exchanged information. In particular, we focus on the design and analysis of algorithms for networked control that, based on the information provided by the CoCPN translator, enable controllers to 1) quickly respond to changes of the network QoS and 2) efficiently utilize the available communication resources to achieve and maintain the wanted target control performance.

In the next section, we outline the structure and the contributions of this thesis. In the subsequent section, then, we briefly discuss relevant related work.

## 1.1 Outline and Contributions

The main research question of this thesis is how to exploit the information provided by the CoCPN translator so that the available communication resources are optimally used. This makes it necessary to consider the plant dynamics, the potential impact of sensor noise and other unmodeled disturbances, and also the impact of the shared communication system during the controllers’ decision-making. For a consistent and rigorous consideration of the uncertainties resulting from these factors, we approach the aforementioned research question with methods from *stochastic control*.

The structure of this thesis is visualized in Fig. 1.3. We begin with a more detailed description of the



**Figure 1.3:** The structure of this thesis.

CoCPN architecture in Chapter 2. In particular, we formalize the interfaces provided by the CoCPN translator and introduce the notion of *quality of control (QoC)*, which is used i) by the controllers to report their communication requirements to the communication system and ii) by the control-aware congestion control to balance the usage of shared network resources between different NCS. Then, the main contributions of this thesis are proposed, each of which is briefly summarized in one of the subsections below.

The first contribution, presented in Chapters 3 and 4, is an extension of an existing model for the design and analysis of networked control algorithms that allows us to pursue a more realistic consideration of the impact of packet delays and losses in this thesis in comparison to the majority of the literature. This model forms the basis for the two novel algorithms for networked stochastic model predictive control that are derived in Chapter 5 and which constitute our second contribution. Likewise, the model is used in Chapter 6, where we investigate the impact of uncertainties in the network model provided by the CoCPN translator on the control performance. Conditions for the stability of the closed-loop dynamics and the existence of stabilizing control laws in the presence of such uncertainties are the final contribution of this thesis.

In Chapter 7, we integrate our contributions into the CoCPN architecture and illustrate CoCPN and its cooperative concept in different simulation scenarios with a challenging control task. We show that especially our proposed model predictive control algorithms, supported by the data exchange provided by the CoCPN translator, avoid without undesirable performance degradations even when the communication conditions change rapidly. Finally, we conclude this thesis in Chapter 8 with a summary of our results and a discussion of potential future research topics.

### 1.1.1 Model for Design and Analysis of Networked Control Algorithms

**Challenge** To consider the influence of the shared communication system during the computation of new control inputs, networked controllers need to be equipped with a network model. The model needs to accurately describe the impact of the relevant influencing factors, namely the impact of packet delays and losses, and must be flexible enough to reflect changes of the network QoS. In particular, the network model shall not be limited to a specific representation of the occurring packet delays and losses.

**Contribution** We extend an existing modeling approach for the design and analysis of networked control algorithms, which compactly expresses the whole NCS, that is, the plant dynamics and the impact of the shared communication system, in terms of a *hybrid system*, which is a special type of a dynamical system whose state variables decompose into a continuous- and a discrete-valued subset. Originally, this model was proposed under the assumption that packet delays and losses can be accurately described by an independent and stationary random process. For a more realistic treatment of packet delays and losses in this thesis, we relax this assumption. We show that the same hybrid system model arises if we allow the random process to be non-stationary and correlated. Additionally, we avoid the assumption that acknowledgments are instantaneously delivered without failure, which is posited in the majority of the literature. Instead, we adopt a more practical point of view and take delays and losses during their transmission into account.

### 1.1.2 Networked Stochastic Model Predictive Control

**Challenge** Given the obtained hybrid system model, we can predict the impact of packet delays and losses on the plant behavior and the control performance. Hence, *stochastic model predictive control (SMPC)* approaches are inherently well-suited for the computation of control inputs. However,

for the considered NCS, the lack of reliable actuator feedback leads to the so-called dual effect: The control inputs affect the plant state – namely, the desired control action – and also the controller’s future uncertainty of the plant state, that is, the future estimation error. The presence of the dual effect makes the analytical determination of control laws that are optimal with regards to a selected cost function impossible, and even worse, renders their numerical computation intractable.

**Contribution** We propose two novel approaches for the computation of suboptimal control laws. Both approaches exhibit contrasting properties. The first approach proposes a control law which is a nonlinear function of the available information and based on a global approximation of the cost-to-go, which we obtain by exploiting the connection between hybrid systems and multiple model algorithms for state estimation. However, the impact of the dual effect on the cost is neglected. In contrast, in the second approach, the underlying control law is a linear function of the available information and derived based on a local approximation of the cost-to-go, i.e., we iteratively minimize an upper bound of the cost via repeated improvements of a given reference trajectory. Moreover, this approximation takes the impact of the dual effect into account. Simulations demonstrate that our approaches can achieve markedly better control performance than state-of-the-art approaches for networked SMPC from the literature.

Neither of the two proposed approaches can be deemed superior to the other. The second approach belongs to the class of closed-loop feedback approaches and, as such, anticipates the availability of future measurements during the computation of the control inputs. On the other hand, the first approach lacks this feature due to the negligence of the dual effect, which is disadvantageous from a theoretical perspective. However, as the simulation results confirm, the resulting performance loss is typically only slight for short optimization horizons. Moreover, this approach might be more suitable from a practical perspective because its computational complexity is by far lower.

### 1.1.3 Networked Stochastic Robust Control

**Challenge** The network model provided by the CoCPN translator is based on monitoring data collected inside the communication system. Thus, it is always only an estimate of the “state” of the underlying communication system. Errors or uncertainties in the provided model, however, directly affect the control performance, since it is needed to construct the hybrid system model used to compute the control inputs. Consequently, investigating the influence of modeling errors on the achievable control performance and the design of control algorithms that are robust to this kind of modeling errors is crucial.

**Contribution** To address this challenge, we first show that an uncertain or even completely unknown network model results in a polytopic parameter uncertainty in the hybrid system model. We then derive a necessary and sufficient condition for the mean square stability of such a polytopic system, thereby extending existing results from the literature. As it is  $\mathcal{NP}$ -hard to determine whether the found condition is satisfied or not, we proceed with the search for a more practical criterion. To that end, we propose a sufficient stability condition in terms of a set of linear matrix inequalities that are easy to evaluate by state-of-the-art solvers and less restrictive than comparable ones from the literature. Finally, we obtain a similar set of inequalities whose feasibility implies the existence of a linear state feedback law that stabilizes the plant in the mean square sense.

## 1.2 Related Work

In the literature, we can identify many research results that focus on the interaction of control and communication in networked control systems or, more generally, in cyber-physical systems. In this section, we provide an overview of approaches, which, similar to CoCPN, focus on the coupling effects induced by the shared usage of communication resources. Relevant references in the fields of our contributions are presented later in the respective chapters.

A broad spectrum of related work is concerned with the design and analysis of networked control algorithms for communication networks with desired properties such as low latencies or packet error rates. These properties, then, permit the design of specialized and made-to-measure algorithms with provable performance guarantees. On the other hand, fulfilling the posited properties typically requires the deployment of dedicated hardware, which limits the applicability of the resulting algorithms. Recent approaches in this regard are [28–30], which develop methods for control over specially tailored low power wireless networks [31, 32]. Similarly, the research published in [33–35] addresses the control over wireless networks based on the IEEE 802.15.4 standard, and control over FlexRay and EtherCAT fieldbus networks is considered in [36–39].

A second line of research aims to increase the predictability of general-purpose networks to support real-time capable data transmissions. To that end, several dedicated communication protocols have been standardized, the most notable of which are the time-sensitive networking (TSN) standards from the IEEE 802.1 working group, the White Rabbit extension of the Precision Time Protocol (PTP), and the standards resulting from the IETF deterministic networking (DetNet) effort [40, 41]. By providing global transmission schedules, these protocols allow for time-triggered communication with deterministic latencies and low jitter [42, 43]. However, the transmission schedules are typically determined offline, and the high computational complexity renders necessary updates, e.g., due to starting or stopping data flows, difficult [44, 45]. Moreover, the required accurate time synchronization, which is especially crucial for high-bandwidth applications, can be hard to maintain in wireless environments [46–48].

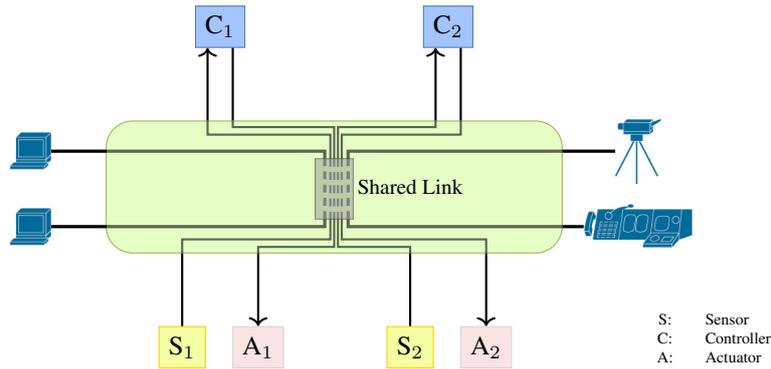
Finally, a great deal of research addresses the coupling effects that arise from the shared usage of the communication system by carrying out cross-layer optimizations. The key idea is to formulate optimization problems, whose solutions maximize the performance of all networked control systems in the CPS and at the same time optimize the sharing of the communication resources. To formulate the optimization problems, the published results typically assume that the communication resources are assigned to the senders according to some policy, e.g., the medium access is granted based on time slots or transmission frequencies [49–56]. A different track is pursued in the works [57–60], which indirectly optimize the sharing of the communication resources by finding optimal controller sampling rates. However, cross-layer optimization demands a central entity with a global view of the communication system and the communication demands of all end systems. Hence, the presented approaches tightly couple the control loops and the communication system, rendering them relatively inflexible.

# CoCPN: Cooperative Cyber-Physical Networking

As introduced in the previous chapter, CoCPN specifically targets cyber-physical systems that are deployed with standard networking equipment as an adaptable and flexible basis for networked applications. To ensure a reliable operation of control loops within such applications, three main challenges must be addressed. First, control loops typically have different requirements regarding data rates or latencies and are assigned different priorities within the CPS. Second, the access to limited communication resources must be balanced between all control loops such that high performance of the whole CPS is ensured. Lastly, sharing common network infrastructure can cause undesirable side effects as different end systems may interfere with each other. To illustrate these challenges, consider Fig. 2.1, which shows two control loops that share a link within the network. Sharing this link establishes an *implicit coupling* between the two control loops, even if they are unrelated in the physical world. Any over-utilization of the link, which, for instance, might occur when  $C_2$  increases its sending rate, leads to larger communication delays or, even worse, packet losses. As this affects all packets traversing the link, the performance achieved by  $C_1$ , and thus the performance of the overall CPS, degrades.

CoCPN addresses the aforementioned challenges by a cooperative usage of the available communication resources. More precisely, it relies on cooperation i) between the communication system and the control loops and ii) among the control loops themselves. For this purpose, CoCPN introduces the *CoCPN translator* . The CoCPN translator enables the data exchange between control loops and the communication system. Solely based on the exchanged data, the cooperation is realized in a flexible, lightweight, and distributed manner, and tight coupling between the control loops and the communication system is avoided. In particular, no central entity with a global view on the communication resources is needed.

In the first section of this chapter, Section 2.1, we provide a detailed description of the CoCPN architecture. In the course of the description, we formalize the interface provided by the CoCPN translator and define the notion of *quality of control (QoC)*, which is used within CoCPN to report the controllers' communication requirements to the communication system and to balance the usage



**Figure 2.1:** Implicit coupling of two networked control systems due to a shared link (adapted from own publication [290]).

of the shared communication resources. The CoCPN architecture is fully implemented in *CoCPN-Sim*, an open source simulation and evaluation framework for cyber-physical systems that has been developed within the scope of our research [295].<sup>1</sup> By combining well established tools from control engineering (Matlab) and the communications community (OMNeT++/INET), it permits detailed and fine-grained investigations of the interplay between the control loops and the internal mechanisms of the communication system for, e.g., congestion and flow control. We briefly introduce CoCPN-Sim in the second section of this chapter, Section 2.2.

*The following descriptions of the CoCPN architecture and CoCPN-Sim are extended versions of the descriptions given in the publication [290].*

## 2.1 The CoCPN Architecture

As mentioned above, one aspect of CoCPN is to implement cooperation between the communication system and the control loops. The cooperation is enabled by the CoCPN translator, which processes monitoring data collected within the communication system. To provide secure and timely information with low overhead, the monitoring building block SERUM (Secure Network Route Monitoring) was developed [61]. Instances of SERUM are part of the network layer of the end systems and routers and collect status information along the communication paths, such as link and queue utilizations and packet loss and error rates. The CoCPN translator translates the collected data into a probabilistic description of the occurring packet delays and losses that is suitable for the controller. We will formalize the probabilistic network model later in Section 3.2.

The network model permits the controller to take the impact of potential packet delays and losses into account during the computation of new control inputs. To that end, CoCPN leverages the *sequence-based control* paradigm. Sequence-based control, which is often also called *packetized predictive control*, is a well-established method to mitigate the impact of packet delays and losses [62–67]. The general idea is to transmit a complete sequence  $\underline{U}_k$  of control inputs that also contains predictive inputs for the next, say  $N - 1$ , time steps, in addition to the current one  $u_k$ . The actuator buffers these predictive inputs and applies them to the plant if follow-up data packets sent by the controller do not arrive in time. An appropriately chosen sequence length thus compensates for packet delays and losses and prevents the plant from running in open-loop operation. We will formalize both the concept of a control sequence and the buffering procedure employed by the actuator in Section 3.1.

<sup>1</sup> CoCPN-Sim is released under the GNU General Public License (GPL) and available on github: <https://github.com/spp1914-cocpn/cocpn-sim>

To prevent undesired side effects from the implicit coupling between control loops mentioned above, CoCPN additionally implements cooperation among the control loops to support a cooperative usage of the shared communication resources. The key enabler of this cooperation is the control-aware *cooperative congestion control (CoCC)*, which seeks to distribute the available communication capacity among the control loops such that their control performance is kept in balance [68,69]. The working principle of CoCC is motivated by the fact that disturbance rejection and control performance of a digital control controller improve with its sampling rate  $f_a$  [70, 71]. In fact, the achievable control performance typically increases monotonically with  $f_a$  [72–74]. Thus, the relationship between control performance and sampling rate can be written as  $\text{performance} = p(f_a)$  for some function  $p$  that is (strictly) increasing.

Exploiting this connection, CoCC dynamically adjusts the target control performance that the controllers shall achieve. Each controller, then, in turn accordingly increases or decreases its sampling rate and, consequently, its sending rate, as much as is necessary to reach this target performance. Based on this principle, CoCC implements a fair sharing of the available communication capacity that avoids over-utilized links and allows all controllers to achieve the same performance. This does usually not lead to equal sending rates for all controllers because their communication requirements differ. Thus, for the computation of the target control performance, CoCC must know the communication requirements of each controller and relate them to the achievable control performance.

However, how control performance is assessed or measured depends on the control task, application-specific requirements and constraints, and the type of controller in use. For instance, for classical approaches such as proportional-integral-derivative (PID) controllers one uses other performance measures than for MPC approaches [75–77]. Hence, CoCPN uses an abstract, normalized performance index, referred to as *quality of control (QoC)*, which is defined as follows.

**Definition 2.1:** (Quality of Control (QoC))

*The notion of quality of control (QoC) defines an abstract measure of control performance that attains values in  $[0, 1]$  and is comparable among all control loops in the CPS. Within the CoCPN architecture, the controllers use it to report their communication requirements to the communication system, and CoCC uses it to balance the available communication capacity. For this purpose, the CoCPN translator provides a mapping*

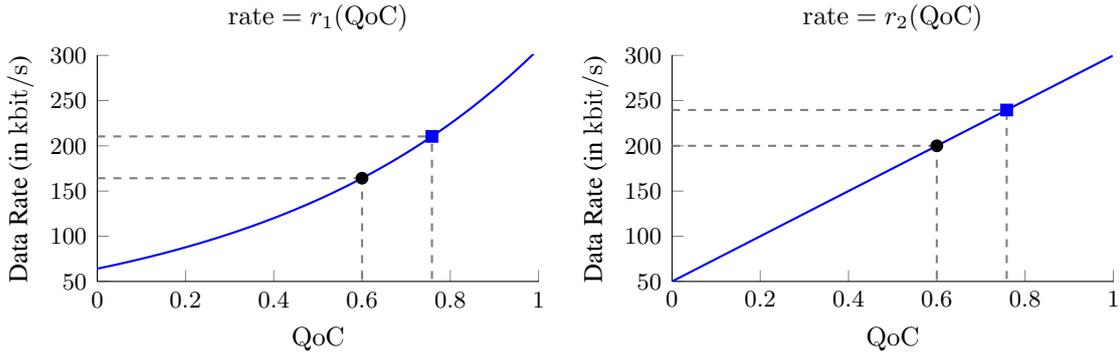
$$\text{rate} = r(\text{QoC}), \quad (2.1)$$

*with  $\text{rate} \geq 0$  (in bits per second), that describes the relationship between control performance and the data rate required to achieve it.*

To provide the mapping  $r$ , the CoCPN translator first translates the application-dependent performance measure performance into QoC. Their relationship must be specified by the system designer in advance and should be such that  $\text{QoC} = 0$  corresponds to some minimal performance that still guarantees desirable properties such as stability of the closed-loop dynamics.

In Chapter 7, where we integrate the control algorithms to be developed in this thesis into the CoCPN architecture, we use the control performance that can be achieved with different sampling rates between minimum and maximum values  $f_{a_{\min}}$  and  $f_{a_{\max}}$  as the basis for the QoC calculation. Based thereon, we first obtain a mapping  $\text{performance} = p(f_a)$  such that  $p(f_{a_{\max}}) = 1$  and  $0 < p(f_a) < 1$  for  $f_{a_{\min}} \leq f_a < f_{a_{\max}}$ . This mapping has an intuitive interpretation:  $p(f_a)$  quantifies the “amount” of control performance that remains when the sampling rate is reduced from  $f_{a_{\max}}$  to  $f_a$ . A straightforward normalization then defines a relationship between performance and QoC according to

$$\text{QoC} \triangleq \frac{p(f_a) - p(f_{a_{\min}})}{1 - p(f_{a_{\min}})},$$



**Figure 2.2:** Illustration of possible mappings  $\text{rate} = r(\text{QoC})$  provided by the CoCPN translator  $\text{\textcircled{T}}$ .

such that  $\text{QoC} = 1$  for  $f_{\text{amax}}$  and  $\text{QoC} = 0$  for  $f_{\text{amin}}$ .

Based on the defined relationship between performance and QoC, the CoCPN translator then translates the mapping  $\text{performance} = p(f_a)$  into a mapping of the form (2.1). This mapping is again application-dependent and should satisfy  $r(\text{QoC}_1) < r(\text{QoC}_2)$  for  $\text{QoC}_1 < \text{QoC}_2$ , that is, it should be strictly increasing to reflect that the achievable control performance increases with the sending rate. A conceptual implementation of the interface between the controller and the communication system provided by the CoCPN translator is given in Algorithm 2.1.<sup>2</sup>

---

**Algorithm 2.1** Conceptual Implementation of the  $\text{\textcircled{T}}$  Interface: Controller  $\rightarrow$  Communication System

---

**Input:**  $\text{performance} = p(f_a)$  // Application-dependent relationship  
**Output:**  $\text{rate} = r(\text{QoC})$  // Strictly increasing mapping  
 1: Translate performance into QoC // Application-dependent translation  
 2: Translate  $f_a$  into rate // From Hz to bit/s  
 3: Compute mapping  $r$   
 4: **return**  $r$

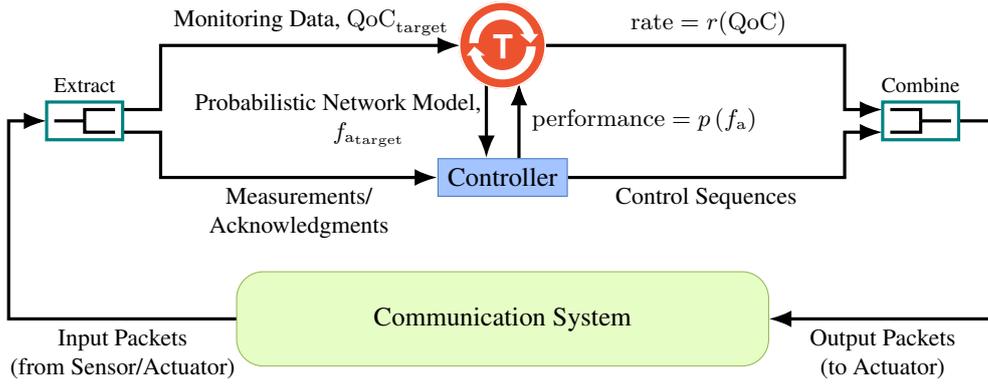
---

To illustrate the notion of QoC, let us again consider the two control loops in Fig. 2.1 that share a link within the network. Let the corresponding relationships between QoC and data rate be given by the functions  $r_1$  and  $r_2$  depicted in Fig. 2.2. If  $\text{QoC}(C_1) = \text{QoC}(C_2)$ , say 0.6 (marked by the black dot in the figure), then from the perspective of CoCPN and CoCC the performance of the two controllers is equal and the available link capacity is shared in a fair manner. However, as Fig. 2.2 exhibits, this sharing does not result in equal data rates since we get  $r_1(0.6) = 164$  kbit/s but  $r_2(0.6) = 200$  kbit/s, indicating the different communication requirements of  $C_1$  and  $C_2$ . If, on the other hand,  $\text{QoC}(C_1) \neq \text{QoC}(C_2)$ , the available link capacity is not shared in a fair way between  $C_1$  and  $C_2$  because then either  $C_1$  achieves better performance than  $C_2$  (in case  $\text{QoC}(C_1) > \text{QoC}(C_2)$ ), or  $C_2$  achieves better performance than  $C_1$  (in case  $\text{QoC}(C_1) < \text{QoC}(C_2)$ ). Hence, CoCC must adjust the sharing of the link capacity to restore fairness.

Based on the notion of QoC, the basic (simplified) working principle of CoCC is as follows. Given a link  $l$  with capacity  $r_l$  (in bits per second) that is shared by  $n$  control loops, CoCC tries to find the

---

<sup>2</sup>In our current implementation of the CoCPN architecture, Algorithm 2.1 is only called once at the beginning. That is, we assume that the communication requirements of the controller, expressed by the mapping  $r$ , do not change over time. However, CoCPN supports changes of  $r$  at runtime.



**Figure 2.3:** Refined illustration of the data exchange provided by the CoCPN translator  $\text{\textcircled{T}}$  enabling the cooperative usage of the communication resources.

maximum  $QoC_{\text{target}}$  that satisfies

$$r_l \geq \sum_{i=1}^n r_i(QoC_{\text{target}}), \quad (2.2)$$

where  $r_i$  is the mapping (2.1) provided by the CoCPN translator associated with the  $i$ -th controller. Note that (2.2) expresses the fairness goal of CoCPN since  $QoC_{\text{target}}$  is equal for all control loops. Based on Newton's method, CoCC iteratively computes  $QoC_{\text{target}}$  in a distributed, round-based manner with fixed round durations. For a detailed description of the distributed implementation and the challenges that arise in multi-hop networks, we refer the reader to [68,69].

Continuing the example with the two control loops in Fig. 2.1, let us assume that the shared link provides a capacity of  $r_l = 450$  kbit/s for the control data flows. Then, (2.2) becomes

$$450 \geq r_1(QoC_{\text{target}}) + r_2(QoC_{\text{target}}).$$

With  $r_1$  and  $r_2$  as shown in Fig. 2.2, the maximum  $QoC_{\text{target}}$  satisfying this inequality is 0.75, highlighted by the blue square in the figure.

Using again the mapping  $r$ , the CoCPN translator extracts the corresponding target sampling rate  $f_{a_{\text{target}}}$ . The controller, finally, alters its sampling rate accordingly and sets  $f_a = f_{a_{\text{target}}}$ , which in turn leads to a changed sending rate. A conceptual implementation of the interface between the communication system and the controller provided by the CoCPN translator is shown in Algorithm 2.2.

---

**Algorithm 2.2** Conceptual Implementation of the  $\text{\textcircled{T}}$  Interface: Communication System  $\rightarrow$  Controller

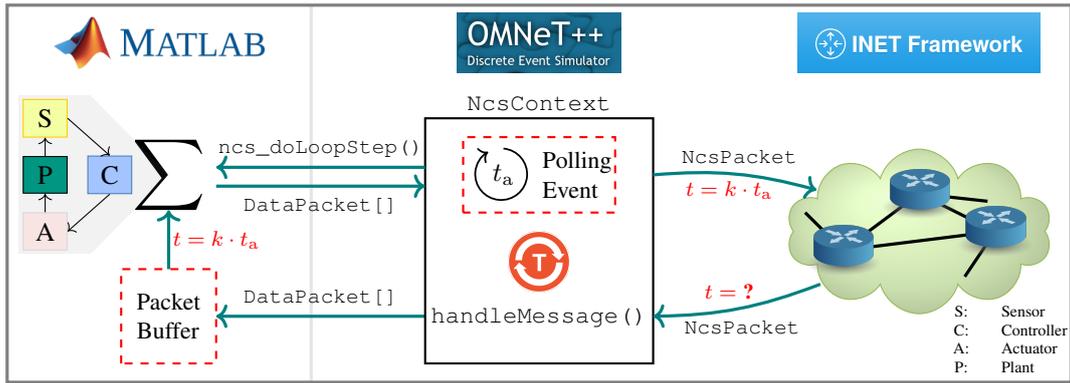
---

**Input:** Monitoring Data,  $QoC_{\text{target}}$

**Output:** Network Model, Target Sampling Rate  $f_{a_{\text{target}}}$

- 1: Translate monitoring data into network model // As formalized in Section 3.2
  - 2: Translate  $QoC_{\text{target}}$  back into target sampling rate  $f_{a_{\text{target}}}$  // Based on mapping  $r$
  - 3: **return** Network model,  $f_{a_{\text{target}}}$
- 

The CoCPN architecture and the data exchange provided by the CoCPN translator are illustrated in Fig. 2.3. CoCPN is fully implemented in the simulation and evaluation framework *CoCPN-Sim*, which we introduce next.



**Figure 2.4:** Illustration of the data flow between Matlab and OMNeT++/INET within CoCPN-Sim (adapted from own publication [290]).

To conclude this section, we emphasize that QoC is not a suitable measure to determine which controllers are superior to others in a given application. This would be a typical task of controller performance assessment, where one usually draws on established (theoretical) benchmarks [78, 79]. Instead, within the CoCPN, we assume that all controllers are properly designed to achieve the intended control goals.

## 2.2 CoCPN-Sim at a Glance

CoCPN-Sim integrates the event-driven simulation framework *OMNeT++* [80] and the numerical computing platform *Matlab*. *Matlab* is widely used in the control community for the design and analysis of control systems, and *OMNeT++* and its accompanying model suite *INET* [81] are often used for network analyses. Hence, in CoCPN-Sim, *Matlab* supplies the mathematical computing toolbox for all control-related tasks within a CPS, and *OMNeT++* and *INET* are used to model the communication system. Thanks to this combination, CoCPN-Sim allows the fine-grained analysis of CPS and the consideration of sophisticated network scenarios and topologies.

To realize the required communication between the components of a networked control system – for instance, control sequences sent from the controller to the actuator and measurements sent from the sensor to the controller – in the simulation, data has to be exchanged between *Matlab* and *OMNeT++* and then translated into an equivalent representation suitable for further processing in *OMNeT++*. Additionally, we must integrate their clock-driven communication into the event-driven workflow of *OMNeT++*. To that end, CoCPN-Sim defines a data flow interface with components that reside both in its *Matlab* part and its *OMNeT++* part. This interface is sketched in Fig. 2.4. Its main component is called *NcsContext* and resides inside the *OMNeT++* part. An *NcsContext* represents a control loop within *OMNeT++* and offers a polling event which is used by the simulation kernel to periodically (with period  $t_a = 1/f_a$ ) trigger the clock-driven communication between the components of the NCS modeled in *Matlab*. As illustrated in the figure, the *NcsContext* calls the hook function `ncs_doLoopStep` which then prompts *Matlab* to conduct all necessary computations. A conceptual implementation of this function is shown in Algorithm 2.3.

As a result of this call, a set of *DataPackets* is handed back to *OMNeT++*. A *DataPacket* is a unified and serializable representation of a message to be exchanged between the components of the NCS, e.g., a control sequence to be sent to the actuator or a new measurement taken by the sensor. After being transformed into an *INET*-compatible representation, the messages are forwarded to the network model. From this moment on, all further processing steps happen asynchronously

---

**Algorithm 2.3** Conceptual Implementation of `ncs_doLoopStep`

---

**Input:** Time Step  $k$ , Network Model (optional), Target Sampling Rate  $f_{a_{\text{target}}}$  (optional)**Output:** `DataPacket[]` // Output Packets

- 1: Fetch input packets from packet buffer
  - 2: Process input packets // Control sequences, measurements, and acknowledgments
  - 3: Update network model (optional) // As formalized in Section 3.2
  - 4: Adapt sampling rate:  $f_a = f_{a_{\text{target}}}$  (optional)
  - 5: Create output packets // Control sequences, measurements, and acknowledgments
  - 6: **return** Output Packets
- 

with regards to the time domain of the NCS. Once a message arrives at a network node, i.e., at an end system that corresponds to a component of the NCS, the `NcsContext` passes it back to Matlab where it is stored in a buffer to be processed in the next control cycle, i.e., during the next invocation of `ncs_doLoopStep`.

Additionally, the `NcsContext` provides the implementations of the CoCPN translator interface. That is, it offers an implementation of Algorithm 2.1 to supply the communication system with the mapping rate  $= r(\text{QoC})$ . Likewise, it provides an implementation of Algorithm 2.2, which is used by the network monitoring SERUM and the congestion control CoCC to exchange the monitoring data and the target QoC.<sup>3</sup> The resulting network model and the target sampling rate  $f_{a_{\text{target}}}$  are forwarded to the controller via `ncs_doLoopStep`. Note that they are optional parameters because SERUM and CoCC can operate at different rates, i.e., at rates other than  $f_a$ .

To conclude our brief introduction of CoCPN-Sim, we outline the typical steps that are required to create a simulation. First, one implements the components of the NCS, i.e., controller, actuator, sensor, and plant, in Matlab. Then, second, these implementations are packaged into a single C++ shared library by the Matlab Compiler SDK [82]. Next, the CoCPN translator interface is implemented. Subsequently, the desired network topology is specified in OMNeT++. To that end, any of the models shipped with INET can be used. Alternatively, one can also implement tailored models for, e.g., hosts, switches, or routers. Finally, all components within OMNeT++ are parameterized according to the desired scenario.

We will return to CoCPN-Sim in Chapter 7, when we integrate the sequence-based control algorithms to be developed in this thesis into the CoCPN architecture. The design and analysis of the algorithms is based on the consideration of a single networked control system, the description and formalization of which is the goal of the next chapter.

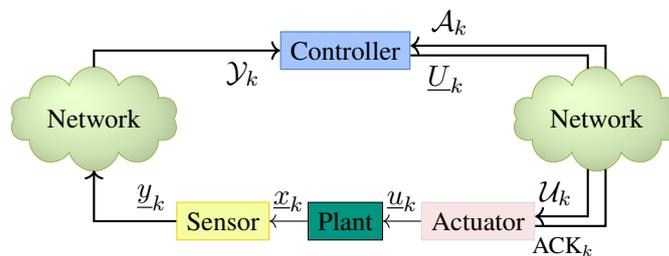
---

<sup>3</sup>SERUM and CoCC are omitted in Fig. 2.4. They reside in the OMNeT++ part of CoCPN-Sim.



## Description of the Considered Networked Control System

For the design and analysis of sequence-based control algorithms that fit into the CoCPN architecture introduced in the previous chapter, we will consider a single NCS as depicted in Fig. 3.1 throughout this thesis.



**Figure 3.1:** The NCS with a sequence-based controller and application layer acknowledgments considered in this thesis.

All components of the NCS operate on a time-triggered basis, and we assume that the clocks of the sensor, the controller, and the actuator are synchronized. This assumption is reasonable in practice because dedicated, well-established networking protocols for clock synchronization are available. The most prominent examples are the Network Time Protocol (NTP) and the Precision Time Protocol (PTP) [83–85], both of which are supported by many operation systems and typically implemented on standard network equipment. Under ideal communication conditions in local area networks, NTP can achieve one millisecond accuracy. Among end systems connected over the Internet, accuracies of tens of milliseconds are possible in the absence of asymmetric routes and heavy network congestion. PTP is even more suitable for control applications because clock accuracy in the sub-microsecond range can be obtained [86].

Thus, the following assumption is justified and stands for the rest of this thesis.

**Assumption 3.1:**

*Transmitted data packets are marked with timestamps so that the experienced delays can be computed by the receivers.*

The components of the NCS communicate with each other using networks with a best-effort packet delivery subject to random transmission delays and data losses. In particular, the successful delivery of data packets is not guaranteed, and no acknowledgments are sent back to the sender upon successful delivery. Such a datagram-oriented service is provided by the *User Datagram Protocol (UDP)* [87], one of the two most common transport protocols in IP-based communication networks. The other one is the *Transmission Control Protocol (TCP)*, which in contrast to UDP provides a reliable transport service since it assures that all transmitted data arrives at the receiver side error-free and in order and without losses or duplicates and additions [88]. TCP achieves reliability by retransmitting potentially lost packets – a packet is assumed lost if a corresponding acknowledgment is not received in a certain amount of time – and by using checksums and sequence numbers for packet reordering and the detection of errors, duplicates, and phantom packets at the receiver side. However, retransmissions result in additional packet delays, which are undesirable in control applications since the carried payload may have become outdated [13, 89]. Outdated data, such as measurements of previous plant states, is often discarded by the receiver anyhow because processing it would gain no or only little information.<sup>1</sup> Moreover, additional packet delays are also caused by flow and congestion control mechanisms, as they may hold back packets at the sender side. Thus, in control applications, UDP is typically preferred over TCP.

On the other hand, acknowledgments constitute the only means for the controller to determine which packets were successfully transmitted to the actuator and, consequently, which control inputs were actually applied to the plant. It is well known that even for linear plants, the instantaneous and failure-free delivery of acknowledgments is required for the existence of computationally tractable optimal control laws [90, 91]. In the NCS literature, networks that meet this requirement are called *TCP-like* networks, whereas networks that do not provide acknowledgments are called *UDP-like*. The notion of TCP-like networks is, however, only loosely related to real-world TCP implementations since TCP-like communication does not require a reliable transport service. Moreover, in real-world communication networks, acknowledgments are also affected by communication delays and packet losses. Arguably, these effects can be mitigated by implementing preferred handling of acknowledgment packets, for instance, by defining expedited forwarding [92] as offered by the Differentiated services (DiffServ) architecture [93, 94], yet physical constraints such as propagation delay and processing speeds of the intermediate nodes still apply.

To combine the advantages of UDP over TCP for control applications with the benefits for the controller that arise when feedback about the status of sent packets is available, we will rely on *application layer acknowledgments* in this thesis. This is motivated by the fact that the actuator actively discards certain data packets from the controller once it is clear that the carried payload will never be processed. This strategy is a variant of what is named *active packet dropout* or *past packets rejection logic* in the literature [13, 95] and detailed in the next section. Based on this strategy, the actuator sends out acknowledgments only for data packets that are actually used. We emphasize that, from the perspective of the underlying communication system, these acknowledgments are regular data packets and hence subject to delays and losses. We also emphasize that there are relatively few works in the literature that consider the imperfect transmission of acknowledgments from the actuator, which is noticeable compared to the multitude of contributions dealing with TCP-like and UDP-like communication schemes. To the best of our knowledge, the only available results can be found in [96–98]. However, these works only consider the potential loss of acknowledgment packets but do not investigate the

---

<sup>1</sup>This will become evident later in the next section.

impact of transmission delays, and none of these works is embedded into the framework of sequence-based control. In this regard, the more realistic treatment of acknowledgment packets compared to the literature pursued in this thesis is our first contribution.

We note that, depending on the application, it is also beneficial for the sensor that the controller acknowledges the successful reception of measurements. Typical examples are *event-based* methods that try to minimize communication expenditure and transmit measurements to the controller only when really necessary [99–102]. The transmission decision usually involves estimating the controller’s expected information gain when the measurement is processed. Knowing which of the transmitted measurements were received can significantly improve this estimate and, consequently, improve the sensor’s decision-making. However, such methods are out of the scope of this thesis, so that acknowledgments from the controller are not needed.

In the remainder of this thesis, whenever we use the term “acknowledgment” or its abbreviation “ACK” we thus refer to an acknowledgment sent back from the actuator to the controller. We formalize the actuator’s acknowledgment procedure in the next section, together with a detailed description of its past packets rejection logic.

Before we proceed, we make the following assumption regarding the transmission of measurements, control sequences and ACKs.

**Assumption 3.2:**

*Measurements, control sequences, and ACKs fit into a single data packet and are not fragmented into multiple individually transmitted and routed packets.*

This assumption is justified because today’s general-purpose communication infrastructure is primarily based on the IEEE 802.3 (Ethernet) and IEEE 802.11 (WLAN) standards. These allow packets to contain payloads up to 1500 bytes and 2304 bytes, respectively, thus providing space for up to 187 (288) double-precision floating-point numbers [103, 104].

## 3.1 Formalization of the NCS

In the NCS shown in Fig. 3.1, the plant is linear with discrete-time dynamics given by

$$\begin{aligned}\underline{x}_{k+1} &= \mathbf{A}\underline{x}_k + \mathbf{B}\underline{u}_k + \underline{w}_k, \\ \underline{y}_k &= \mathbf{C}\underline{x}_k + \underline{v}_k,\end{aligned}\tag{3.1}$$

where  $k \in \mathbb{N} \cup \{0\}$  is the discrete time step,  $\underline{x}_k \in \mathbb{R}^{n_x}$  the plant state,  $\underline{u}_k \in \mathbb{R}^{n_u}$  the control input applied by the actuator,  $\underline{y}_k \in \mathbb{R}^{n_y}$  the measurement, and  $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$ ,  $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$  and  $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$  are the known system, input, and measurement matrix, respectively. State and measurement are corrupted by mutually independent and zero-mean white noise  $\underline{w}_k$  and  $\underline{v}_k$  with covariance matrices  $\mathbf{W}$  and  $\mathbf{V}$ . The initial plant state  $\underline{x}_0$  is Gaussian with mean  $\hat{\underline{x}}_0$  and covariance matrix  $\mathbf{X}_0$  and independent of  $\underline{w}_k$  and  $\underline{v}_{k'}$  for all  $k, k'$ .

Throughout this thesis, (3.1) results from the continuous-time plant dynamics

$$\dot{\underline{x}}(t) = \mathbf{A}_c \underline{x}(t) + \mathbf{B}_c \underline{u}(t) + \mathbf{G}_c \underline{w}(t),\tag{3.2}$$

where  $\underline{w}(t) \in \mathbb{R}^{n_w}$  is zero-mean white noise with intensity  $\mathbf{W}_c$ , i.e.,

$$\mathbb{E}\{\underline{w}(t)\underline{w}(t')^T\} = \mathbf{W}_c \delta(t - t'),$$

which is discretized with sampling interval  $t_a = 1/f_a$  during which  $\underline{u}(t)$  is assumed constant. Consequently,  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{W}$  are given by

$$\mathbf{A} = e^{\mathbf{A}_c t_a}, \quad \mathbf{B} = \int_0^{t_a} e^{\mathbf{A}_c s} \mathbf{B}_c ds, \quad \mathbf{W} = \int_0^{t_a} e^{\mathbf{A}_c s} \mathbf{G}_c \mathbf{W}_c \mathbf{G}_c^T e^{\mathbf{A}_c^T s} ds. \quad (3.3)$$

Recall from the previous chapter that the controller adapts its sampling rate during operation and sets  $f_a = f_{a_{\text{target}}}$  according to Algorithm 2.2 whenever the desired target performance  $\text{QoC}_{\text{target}}$  changes. This in turn necessitates that  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{W}$  in the plant dynamics (3.1) are recomputed using (3.3) with  $t_a = 1/f_{a_{\text{target}}}$ . In this regard, the considered plant dynamics is time-invariant over a limited period only, namely until the controller's next rate adaptation.

To reflect the randomness of packet delays and losses in real networks, we treat them as realizations of an underlying random phenomenon. To describe them formally, we use three distinct stochastic processes  $\tau_k^{\text{ca}}$ ,  $\tau_k^{\text{ac}}$ , and  $\tau_k^{\text{sc}}$ , each of which defined on a finite subset of  $\mathbb{N} \cup \{0\}$ . More precisely,  $\tau_k^{\text{ca}}$  is defined on the set  $\{0, 1, \dots, M-1, M\}$ , and a realization  $\tau_k^{\text{ca}} = i$  for  $0 \leq i < M$  indicates that the packet sent from the controller to the actuator at time  $k$  will experience a delay of  $i$  time steps. The realization  $\tau_k^{\text{ca}} = M$  indicates that this packet will experience a delay of at least  $M$  time steps or get lost. Similarly,  $\tau_k^{\text{ac}}$  is defined on  $\{0, 1, \dots, M^{\text{ac}}-1, M^{\text{ac}}\}$ , and  $\tau_k^{\text{ac}} = i$  for  $0 \leq i < M^{\text{ac}}$  indicates that the ACKs sent from the actuator to the controller at time  $k$  will suffer a delay of  $i$  time steps. A higher delay or a packet loss is indicated by  $\tau_k^{\text{ac}} = M^{\text{ac}}$ . Finally,  $\tau_k^{\text{sc}}$  is defined on  $\{0, 1, \dots, M^{\text{sc}}-1, M^{\text{sc}}\}$ . For  $0 \leq i < M^{\text{sc}}$ ,  $\tau_k^{\text{sc}} = i$  indicates that the packet sent from the sensor to the controller at time  $k$  will experience a delay of  $i$  time steps and, as before,  $\tau_k^{\text{sc}} = M^{\text{sc}}$  indicates a delay of at least  $M^{\text{sc}}$  time steps or a packet loss.

Note that this modeling approach allows us to handle unbounded delays and to treat packet delays and losses in a unified manner. Note also that due to Assumption 3.1 the values of  $\tau_k^{\text{ca}}$ ,  $\tau_k^{\text{ac}}$ , and  $\tau_k^{\text{sc}}$  are known to the respective receivers. The mathematical properties of the stochastic processes are further specified later in Section 3.2.

As a result of the delays and losses, the controller can receive none, one, or multiple data packets from the sensor and the actuator at a given time step. The corresponding set of received measurements is denoted by

$$\mathcal{Y}_k = \{y_{k'} \mid k' + \tau_{k'}^{\text{sc}} = k\}, \quad (3.4)$$

and the corresponding set of received ACKs is

$$\mathcal{A}_k = \{\text{ACK}_{k'} \mid k' + \tau_{k'}^{\text{ac}} = k\}. \quad (3.5)$$

The acknowledgment procedure of the actuator is formalized later in this section.

Based on the received data, the controller computes a new control sequence  $\underline{U}_k$  and sends it to the actuator. Such a control sequence consists of  $N$  control inputs for the current and the next  $N-1$  time steps and is of the form

$$\underline{U}_k = \left[ \underline{u}_{k|k}^T \quad \underline{u}_{k+1|k}^T \quad \cdots \quad \underline{u}_{k+N-1|k}^T \right]^T \in \mathbb{R}^{Nn_u}. \quad (3.6)$$

In (3.6), we use the subscript  $k+i|k$  to indicate that the control input  $\underline{u}_{k+i|k}$  is computed at time  $k$  for application at time  $k+i$ ,  $i = 0, 1, \dots, N-1$ .

Due to packet delays and losses, none, one, or multiple control sequences can become available at the actuator at every time step. In analogy to (3.4) and (3.5), this set is given by

$$\mathcal{U}_k = \{\underline{U}_{k'} \mid k' + \tau_{k'}^{\text{ca}} = k\}.$$

However, in contrast to the controller, the actuator need not process all received data since the control sequence  $\underline{U}_k$  supersedes all older ones  $\underline{U}_0, \underline{U}_1, \dots, \underline{U}_{k-1}$ . Hence, at every time step, the actuator buffers only the newest available control sequence, that is, the one with the youngest *age* according to the following definition.

**Definition 3.1:**

*At time  $k$ , the age of a control sequence  $\underline{U}_{k'}$ , where  $k' \leq k$ , is the difference  $k - k'$ . It will be denoted by  $a_k(\underline{U}_{k'})$  in the following and we set  $a_k(\underline{U}_{k'}) = \infty$  whenever  $k' > k$ .*

Let  $\underline{U}_{k''}$  denote the newest of the received control sequences, i.e.,

$$\underline{U}_{k''} = \arg \min_{\underline{U}_{k'} \in \mathcal{U}_k} a_k(\underline{U}_{k'}) .$$

Then, the buffered control sequence  $\underline{U}_k^{\text{bf}}$  is given by

$$\underline{U}_k^{\text{bf}} = \begin{cases} \underline{U}_{k''} & a_k(\underline{U}_{k-1}^{\text{bf}}) > a_k(\underline{U}_{k''}) \\ \underline{U}_{k-1}^{\text{bf}} & \mathcal{U}_k = \emptyset \text{ or } a_k(\underline{U}_{k-1}^{\text{bf}}) < a_k(\underline{U}_{k''}) \end{cases} . \quad (3.7)$$

If  $\underline{U}_{k''}$  replaces the buffered control sequence, the actuator creates an ACK and sends it to the controller to signal that  $\underline{U}_{k''}$  is in use. Otherwise, no ACK is sent. We will show in the next chapter that this acknowledgment strategy enables the controller to infer control inputs that were applied to the plant from the set  $\mathcal{A}_k$ .

The actuator subsequently applies the control inputs provided by  $\underline{U}_k^{\text{bf}}$  until a newer control sequence arrives to replace it according to (3.7). If the last control input provided by  $\underline{U}_k^{\text{bf}}$  has been applied and no newer sequence is available to replace it, the actuator falls back to a default value  $\underline{u}_k^{\text{df}}$ . We will assume  $\underline{u}_k^{\text{df}} = \underline{0}$  in this thesis. This strategy is known as the *zero-input strategy* in the literature [105]. We emphasize that the results of this thesis are not limited to the application of this strategy. They are readily extended to other choices of the default input, such as holding the previously applied input, i.e.,  $\underline{u}_k^{\text{df}} = \underline{u}_{k-1}$ . Finally, we assume that the default input is also applied initially until the first applicable control sequence arrives at the actuator, that is, the buffer is initially empty.

To conclude this section, we make the following assumption that stands throughout this thesis. It ensures that the controller can influence the plant and serves to rule out pathological cases where any control effort is futile.<sup>2</sup>

**Assumption 3.3:**

*The length of the control sequences,  $N$ , is chosen such that the probability of two consecutive applications of the default input is less than one.*

## 3.2 Properties of the Network Models

Recall that we defined the stochastic processes  $\tau_k^{\text{ca}}$ ,  $\tau_k^{\text{ac}}$ , and  $\tau_k^{\text{sc}}$  in the previous section to model the packet delays and losses that occur during operation of the NCS. In the remainder of this thesis, we will often use the term “network model” to refer to any of them. The context should make clear which one is meant.

<sup>2</sup>An example of such a pathological case would be to choose  $N = 2$  if any packet experienced a delay of at least two time steps.

**Assumption 3.4:**

The stochastic processes  $\tau_k^{ca}$ ,  $\tau_k^{ac}$ , and  $\tau_k^{sc}$  are mutually independent, non-stationary processes.

The non-stationarity assumption is appropriate to reflect the changing communication conditions that originate from variations of the network QoS.

Since the controller knows the delays of received measurements and ACKs (cf. Assumption 3.1), we need not specify further properties of  $\tau_k^{sc}$  and  $\tau_k^{ac}$ . We only require that the probabilities  $P[\tau_k^{sc} = i]$  be computable by the controller for all  $i = 0, 1, \dots, M^{sc}$  at every time step  $k$ . This is needed for the model predictive control algorithm proposed in Section 5.4. In this algorithm, the controller attempts to anticipate the availability of future measurements during its prediction of the closed-loop plant dynamics.

To consider the impact of delayed and lost control sequences on the closed-loop plant dynamics, we will construct an augmented dynamical system in Chapter 4. The key idea is to express the plant dynamics (3.1) and the actuator's buffering procedure (3.7) jointly in a single model, which then provides the basis for controller design and analysis. For this model, we need a characterization of  $\tau_k^{ca}$  because its realizations determine the buffered sequence  $\underline{U}_k^{bf}$  and, thus, the actual plant input. Recall from Chapter 2 that the CoCPN translator processes monitoring data collected in the communication system to create a probabilistic model of the packet delays and losses that is suitable for the controller. Once new monitoring data becomes available, this model is updated and forwarded to the controller (cf. Algorithm 2.2). Throughout this thesis, we do not make any assumption about the rate with which new monitoring data reaches the CoCPN translator and whether this happens periodically or not. We only assume that a probabilistic model of the packet delays and losses is available to the controller at every time step.

Specifically, we consider two different cases, described in the sections below. In both cases, we condense the controller's knowledge at time  $k$  about the communication conditions in the form of a discrete probability distribution, expressed by the stochastic vector  $\underline{p}_k = [p_k^{(0)} \ p_k^{(1)} \ \dots \ p_k^{(M)}]^T$ , whose entries provide the probability associated with each realization, i.e., we have

$$p_k^{(i)} = P[\tau_k^{ca} = i],$$

for  $i = 0, 1, \dots, M$ .

**3.2.1 Independent Packet Delays and Losses**

In the first model, the CoCPN translator directly supplies the controller with the probability distribution  $\underline{p}_k$ . We do not take possible temporal correlations of packet delays and losses into account. Then,  $\tau_k^{ca}$  becomes a non-stationary independent process, that is, the delay and loss probabilities  $p_k^{(i)}$  are independent over time and in general not equal, i.e.,  $\underline{p}_k \neq \underline{p}_{k'}$  for  $k' \neq k$ . However, the controller cannot foresee future changes of the individual probabilities. Hence, it is reasonable to assume that they remain constant until the next model update by the CoCPN translator.

**3.2.2 Correlated Packet Delays and Losses**

Packet delays and losses mainly result from the concurrence of different factors, including congestion, queuing, and processing at intermediate nodes, protocol behavior such as packet retransmission and congestion control mechanisms, and contention for medium access [35, 106, 107]. The impact of these factors on successive packets is usually similar, implying that delays and losses are correlated

over time. To model these correlations, we use a Markov chain approach, as is common in the literature – in particular when wireless communication is considered [34, 108–115]. The probabilistic model forwarded from the CoCPN translator to the controller is then a transition matrix  $\mathbf{P}_{k-1}^{\text{ca}}$ . The entries of  $\mathbf{P}_{k-1}^{\text{ca}}$  are the transition probabilities, which are given by  $p_{ij}^{\text{ca}} = \text{P}[\tau_k^{\text{ca}} = j | \tau_{k-1}^{\text{ca}} = i]$  for  $i, j = 0, 1, \dots, M$ . Based on  $\underline{p}_{k-1}$ , the probability distribution at time  $k$  is then obtained by the expression

$$\underline{p}_k^{\text{T}} = \underline{p}_{k-1}^{\text{T}} \mathbf{P}_{k-1}^{\text{ca}}.$$

Since the controller cannot foresee future changes of the transition probabilities, it is reasonable to assume that they remain constant until the next model update by the CoCPN translator. Then,  $\tau_k^{\text{ca}}$  is a time-homogeneous Markov chain. Regarding the transition probabilities, we make the following assumption.

**Assumption 3.5:**

*All transition probabilities  $p_{ij}^{\text{ca}}$  are positive, i.e.,  $\mathbf{P}^{\text{ca}} > 0$ .*

Of course, this assumption need not be satisfied in reality. However, it is justified from a numerical point of view. In software implementations, it is common practice to place a lower bound, say  $10^{-50}$ , on transition probabilities to avoid numerical problems such as arithmetic underflows or divisions by zero [116, 117]. Also, the assumption is mathematically convenient. By the Perron-Frobenius theorem (cf. Appendix A.6), it guarantees that  $\tau_k^{\text{ca}}$  has a unique stationary distribution  $\underline{p} = [p^{(0)} \ p^{(1)} \ \dots \ p^{(M)}]^{\text{T}}$  and that  $\underline{p} > 0$ , i.e., all entries of  $\underline{p}$  are positive. We shall make use of this fact in the course of Chapter 4.



## Construction of the Augmented Dynamics

In this chapter, we construct an augmented dynamical system that jointly expresses the plant dynamics (3.1) and the actuator's buffering procedure (3.7) in a single model. This model then serves as the starting point for the control algorithms we present later in Chapters 5 and 6. More specifically, we construct a dynamical system of the form

$$\begin{aligned}\underline{\psi}_{k+1} &= \tilde{\mathbf{A}}^{(\theta_k)} \underline{\psi}_k + \tilde{\mathbf{B}}^{(\theta_k)} \underline{U}_k + \tilde{\underline{w}}_k, \\ \underline{y}_k &= [\mathbf{C} \ \mathbf{0}] \underline{\psi}_k + \underline{v}_k,\end{aligned}\tag{4.1}$$

with  $\underline{\psi}_k$  a suitably augmented state and  $\theta_k$  a discrete-valued variable, referred to as the *mode* or *operation mode* of the system (4.1). Dynamical systems, whose state variables decompose into a continuous- and a discrete-valued subset, are known in the literature as *hybrid systems* [118–120]. In a hybrid system, the dynamics of the continuous-valued state (here, in (4.1), the actual system and input matrices  $\tilde{\mathbf{A}}^{(\theta_k)}$  and  $\tilde{\mathbf{B}}^{(\theta_k)}$ ) are dependent on the discrete-valued state, i.e., the mode  $\theta_k$  causes “switchings” between different system dynamics. In networked control, switchings of the plant dynamics naturally appear due to packet delays and losses. For instance, the plant switches from closed-loop operation to open-loop operation once the actuator has to fall back to the default control input as described in the previous chapter. Thus, it is not surprising that hybrid system models are very often chosen for the synthesis and analysis of the networked control and estimation algorithms, e.g., [33–35, 65, 116, 121–126], to name only a few. All these models have in common that both the number of operation modes, i.e., the number of values that  $\theta_k$  can attain, and the switchings between the modes, i.e., the dynamics of  $\theta_k$ , are given by the nature of the network effects.

Specifically for sequence-based control of linear plants with dynamics (3.1) and the buffering procedure (3.7), a hybrid system of the form (4.1) was derived in the doctoral dissertation [123]. There, the author showed that  $\theta_k$  is a Markov chain with state space  $\{0, 1, \dots, N\}$ , rendering (4.1) a *Markov jump linear system (MJLS)*. A MJLS is a particular type of hybrid system where a Markov chain governs the switchings between different linear dynamics [127]. In the doctoral dissertation [123], however,  $\tau_k^{\text{ca}}$  was assumed to be an independent and stationary process, i.e., it was assumed that the  $p_k^{(i)}$  were independent of each other for all  $k, i$  and required that  $\underline{p}_k = \underline{p}_{k'}$  for all  $k \neq k'$ .

	$k-5$	$k-4$	$k-3$	$k-2$	$k-1$	$k$	$k+1$	$k+2$	$k+3$	$k+4$
$\underline{U}_k$						$\underline{u}_{k k}$	$\underline{u}_{k+1 k}$	$\underline{u}_{k+2 k}$	$\underline{u}_{k+3 k}$	$\underline{u}_{k+4 k}$
$\underline{U}_{k-1}$					$\underline{u}_{k-1 k-1}$	$\underline{u}_{k k-1}$	$\underline{u}_{k+1 k-1}$	$\underline{u}_{k+2 k-1}$	$\underline{u}_{k+3 k-1}$	
$\underline{U}_{k-2}$				$\underline{u}_{k-2 k-2}$	$\underline{u}_{k-1 k-2}$	$\underline{u}_{k k-2}$	$\underline{u}_{k+1 k-2}$	$\underline{u}_{k+2 k-2}$		
$\underline{U}_{k-3}$			$\underline{u}_{k-3 k-3}$	$\underline{u}_{k-2 k-3}$	$\underline{u}_{k-1 k-3}$	$\underline{u}_{k k-3}$	$\underline{u}_{k+1 k-3}$			
$\underline{U}_{k-4}$		$\underline{u}_{k-4 k-4}$	$\underline{u}_{k-3 k-4}$	$\underline{u}_{k-2 k-4}$	$\underline{u}_{k-1 k-4}$	$\underline{u}_{k k-4}$				
$\underline{U}_{k-5}$	$\underline{u}_{k-5 k-5}$	$\underline{u}_{k-4 k-5}$	$\underline{u}_{k-3 k-5}$	$\underline{u}_{k-2 k-5}$	$\underline{u}_{k-1 k-5}$					

**Figure 4.1:** The possible inputs (without the default input  $\underline{u}_k^{\text{df}}$ ) at time  $k$  for  $N = 5$  are highlighted by the colored rectangle.

In this chapter, we show that the same MJLS also arises when this assumption is relaxed. First, we perform the state augmentation that is necessary to obtain the augmented dynamical system (4.1) in Section 4.1. We then in Section 4.2 prove that  $\theta_k$  forms a Markov chain whenever  $\tau_k^{\text{ca}}$  is an arbitrary independent process as specified in Section 3.2.1. Then, in Section 4.3, we present a method that allows us to use (4.1) also in case  $\tau_k^{\text{ca}}$  is a Markov chain with the properties introduced in Section 3.2.2. Finally, we conclude this chapter in Section 4.4 with a brief discussion of the modeling approach chosen in this thesis.

In the following, whenever the context is clear, we will use the term “state” for both the plant state  $\underline{x}_k$ , as defined by (3.1) in Chapter 3, and the continuous-valued state  $\underline{\psi}_k$  of the augmented dynamics (4.1). Also, with a slight abuse of terminology, “mode” and “operation mode” will not only refer to the discrete-valued state  $\theta_k$  of the augmented dynamics, but also to the corresponding dynamics of  $\underline{\psi}_k$ . For example, the statement “The system is in mode  $j$ ” means that  $\theta_k = j$  and that  $\underline{\psi}_k$  evolves according to the dynamics  $\underline{\psi}_{k+1} = \tilde{\mathbf{A}}^{(j)}\underline{\psi}_k + \tilde{\mathbf{B}}^{(j)}\underline{U}_k + \tilde{\mathbf{w}}_k$ . Similarly, the statement “The MJLS has  $m$  modes” means that the Markov chain governing the switchings has  $m$  different states.

*In parts, this chapter is based on results presented in our publication [286].*

## 4.1 Definition of the Augmented State

At any time  $k$ , the applied plant input  $\underline{u}_k$  is either provided by the buffered control sequence  $\underline{U}_k^{\text{bf}}$  or is the default input  $\underline{u}_k^{\text{df}}$ . The latter can only happen if no control sequence has yet arrived at the actuator or when  $\underline{U}_k^{\text{bf}}$  has no more applicable inputs available. This is precisely the case when  $a_k(\underline{U}_k^{\text{bf}}) \geq N$ . Thus,  $\underline{u}_k$  must be part of one of the control sequences  $\underline{U}_{k-N+1}, \underline{U}_{k-N+2}, \dots, \underline{U}_k$ , or is the default input  $\underline{u}_k^{\text{df}}$ , i.e., we have

$$\underline{u}_k = \begin{cases} \underline{u}_{k|k-i} & a_k(\underline{U}_k^{\text{bf}}) = i \\ \underline{u}_k^{\text{df}} & a_k(\underline{U}_k^{\text{bf}}) \geq N \end{cases}, \quad (4.2)$$

where  $i \in \{0, 1, \dots, N-1\}$ . We visualize this observation in Fig. 4.1 for  $N = 5$ , where the five possible inputs at time  $k$ , excluding the default input, are highlighted. Eq. (4.2) has an intuitive interpretation: The plant has  $N+1$  possible operation modes, each of which is determined by the age of the buffered sequence, and each of which causes the application of a different control input. Hence, denoting the operation mode by  $\theta_k$ , we can identify

$$\theta_k = \begin{cases} i & \underline{u}_k = \underline{u}_{k|k-i} \\ N & \underline{u}_k = \underline{u}_k^{\text{df}} \end{cases}, \quad (4.3)$$

with  $i \in \{0, 1, \dots, N-1\}$ . Together with the vector  $\underline{\eta}_k$  that contains all inputs from *past* control sequences that are still applicable at time  $k$  or later

$$\underline{\eta}_k = \begin{bmatrix} \left[ \underline{u}_{k|k-1}^T \quad \underline{u}_{k+1|k-1}^T \quad \cdots \quad \underline{u}_{k+N-3|k-1}^T \quad \underline{u}_{k+N-2|k-1}^T \right]^T \\ \left[ \underline{u}_{k|k-2}^T \quad \underline{u}_{k+1|k-2}^T \quad \cdots \quad \underline{u}_{k+N-3|k-2}^T \right]^T \\ \vdots \\ \left[ \underline{u}_{k|k-N+2}^T \quad \underline{u}_{k+1|k-N+2}^T \right]^T \\ \underline{u}_{k|k-N+1} \end{bmatrix} \in \mathbb{R}^{\frac{(N-1)Nn_u}{2}}, \quad (4.4)$$

the actual plant input  $\underline{u}_k$  is expressed as

$$\underline{u}_k = \mathbf{H}^{(\theta_k)} \underline{\eta}_k + \mathbf{J}^{(\theta_k)} \underline{U}_k, \quad (4.5)$$

with  $\mathbf{H}^{(\theta_k)} \in \mathbb{R}^{n_u \times \frac{(N-1)Nn_u}{2}}$ ,  $\mathbf{J}^{(\theta_k)} \in \mathbb{R}^{n_u \times Nn_u}$  defined as

$$\begin{aligned} \mathbf{H}^{(\theta_k)} &= [\mathbb{1}_{\{\theta_k=1\}} \mathbf{I}_{n_u} \quad \mathbf{0}_{n_u \times (N-2)n_u} \quad \mathbb{1}_{\{\theta_k=2\}} \mathbf{I}_{n_u} \quad \mathbf{0}_{n_u \times (N-3)n_u} \quad \cdots \quad \mathbb{1}_{\{\theta_k=N-1\}} \mathbf{I}_{n_u}], \\ \mathbf{J}^{(\theta_k)} &= [\mathbb{1}_{\{\theta_k=0\}} \mathbf{I}_{n_u} \quad \mathbf{0}_{n_u \times (N-1)n_u}], \end{aligned} \quad (4.6)$$

and where  $\mathbb{1}_{\{a\}}$  is the indicator function of the predicate  $a$ , i.e.,  $\mathbb{1}_a = 1$  if  $a$  is true, and 0 otherwise. For  $N = 5$ , the elements comprising  $\underline{\eta}_k$  and  $\underline{\eta}_{k+1}$  are visualized in Fig. 4.2. The dynamics of  $\underline{\eta}_k$  is linear and time-invariant and given by

$$\underline{\eta}_{k+1} = \mathbf{F} \underline{\eta}_k + \mathbf{G} \underline{U}_k, \quad (4.7)$$

with  $\mathbf{F} \in \mathbb{R}^{\frac{(N-1)Nn_u}{2} \times \frac{(N-1)Nn_u}{2}}$  and  $\mathbf{G} \in \mathbb{R}^{\frac{(N-1)Nn_u}{2} \times Nn_u}$  as per

$$\mathbf{F} = \begin{array}{c} \text{\#columns:} \\ \mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} & \mathbf{0} \end{bmatrix} \\ \text{\#rows:} \end{array} \begin{array}{l} n_u \quad (N-2)n_u \quad n_u \quad (N-3)n_u \quad \cdots \quad n_u \quad n_u \\ (N-1)n_u \\ (N-2)n_u \\ (N-3)n_u \\ \vdots \\ n_u \end{array}, \quad (4.8)$$

and

$$\mathbf{G} = \begin{array}{c} \text{\#columns:} \\ \mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ \text{\#rows:} \end{array} \begin{array}{l} n_u \quad (N-1)n_u \\ (N-1)n_u \\ \frac{(N-1)(N-2)n_u}{2} \end{array}. \quad (4.9)$$

The purpose of  $\mathbf{F}$  is to remove obsolete entries from  $\underline{\eta}_k$ , whereas  $\mathbf{G}$  adds the relevant entries from the new sequence  $\underline{U}_k$ . This is illustrated in Fig. 4.2 for  $N = 5$ . For this example, we can deduce from the figure that  $\underline{\eta}_k \in \mathbb{R}^{10n_u}$  and, accordingly,

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{3n_u} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{2n_u} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{n_u} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{10n_u \times 10n_u}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{4n_u} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{10n_u \times 5n_u}.$$



augmented dynamical system if the actuator uses a hold-input strategy (i.e.,  $\underline{u}_k^{\text{df}} = \underline{u}_{k-1}$ ). The main difference compared to the zero-input strategy  $\underline{u}_k^{\text{df}} = \underline{0}$  is that the default input is time-varying with dynamics  $\underline{u}_{k+1}^{\text{df}} = \mathbb{1}_{\{\theta_k=N\}}\underline{u}_k^{\text{df}} + \mathbf{H}^{(\theta_k)}\underline{\eta}_k + \mathbf{J}^{(\theta_k)}\underline{U}_k$  and unknown to the controller. To handle this, however, it is enough to add  $\underline{u}_k^{\text{df}}$  to the augmented state  $\underline{\psi}_k$  and to alter  $\tilde{\mathbf{A}}^{(\theta_k)}$ ,  $\tilde{\mathbf{B}}^{(\theta_k)}$ , and  $\tilde{\underline{w}}_k$  in (4.1) accordingly [123].

## 4.2 Dynamics of $\theta_k$ : Independent Packet Delays and Losses

The following result proves that  $\theta_k$  is a Markov chain provided  $\tau_k^{\text{ca}}$  is an independent process as specified in Section 3.2.1, so that the augmented dynamics (4.1) becomes an MJLS with  $N + 1$  modes.

### Theorem 4.1:

Let  $\tau_k^{\text{ca}}$  be an independent process as defined in Section 3.2.1. Then,  $\theta_k$  forms a time-inhomogeneous Markov chain with state space  $\{0, 1, \dots, N\}$  and transition probabilities  $t_{k,i,j} = \text{P}[\theta_{k+1} = j | \theta_k = i]$  given by

$$t_{k,i,j} = \begin{cases} p_{k+1}^{(0)} & j = 0 \\ \left(1 - p_{k+1}^{(0)}\right) \prod_{m=0}^{i-1} \left(1 - q_k^{(m)}\right) & j = i + 1 \\ 0 & j > i + 1 \\ q_k^{(j-1)} \left(1 - p_{k+1}^{(0)}\right) \prod_{m=0}^{j-2} \left(1 - q_k^{(m)}\right) & 1 \leq j \leq i \leq N - 1 \\ \left(1 - p_{k+1}^{(0)}\right) \prod_{m=0}^{N-2} \left(1 - q_k^{(m)}\right) & i = j = N \end{cases},$$

with  $p_k^{(i)} = \text{P}[\tau_k^{\text{ca}} = i]$  and where  $q_k^{(j)}$  is the conditional probability that  $\underline{U}_{k-j}$  arrives at time  $k + 1$  given that it has not been received up to time  $k$

$$q_k^{(j)} = \text{P}[\tau_{k-j}^{\text{ca}} = j + 1 | \tau_{k-j}^{\text{ca}} > j] = \frac{p_{k-j}^{(j+1)}}{1 - \sum_{m=0}^j p_{k-j}^{(m)}}. \quad (4.11)$$

*Proof.* The proof is given in Appendix B.1. □

### Remark 4.1:

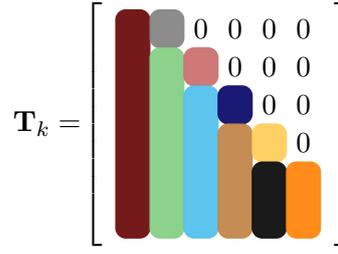
If  $\tau_k^{\text{ca}}$  is stationary, then  $\theta_k$  becomes a time-homogeneous Markov chain. The corresponding time-invariant transition probabilities  $t_{ij}$  were derived in [123, Lemma 3.1]. The same result is obtained by setting  $p_k^{(i)} = p^{(i)}$  for all  $k$  in Theorem 4.1.

The transition matrix  $\mathbf{T}_k$  with entries  $t_{k,i,j}$  allows to express the temporal evolution of  $\theta_k$  in terms of the probability distribution  $\underline{\mu}_k = [\mu_k^{(0)} \ \mu_k^{(1)} \ \dots \ \mu_k^{(N)}]^\text{T}$  according to

$$\underline{\mu}_{k+1}^\text{T} = \underline{\mu}_k^\text{T} \mathbf{T}_k, \quad (4.12)$$

with  $\mu_k^{(i)} = \text{P}[\theta_k = i]$  the mode probabilities.

A closer look at Theorem 4.1 reveals dependencies between the transition probabilities which impose a special structure on the transition matrix  $\mathbf{T}_k$ . This structure is illustrated for  $N = 5$  in Fig. 4.3, where



**Figure 4.3:** Illustration of the dependencies between the entries of  $\mathbf{T}_k$  for  $N = 5$ . Equal entries are represented by the same color.

equal entries are represented by the same color. We can conclude from the illustration that i)  $\mathbf{T}_k$  is a lower Hessenberg matrix, ii) the subdiagonal entries in each column are equal to the column's diagonal entry, and that iii) the last two rows are equal. Together with the constraint that each row must sum to one, these properties imply that the elements of the last row completely determine  $\mathbf{T}_k$ . For instance, using the colors from Fig. 4.3, we have  $\text{grey} = \text{green} + \text{red} = 1 - \text{dark red}$  and similarly  $\text{dark blue} = \text{brown} + \text{black} + \text{orange}$ . We will exploit this observation later in Chapter 6 where we develop a method to tackle uncertainties in the network model  $\tau_k^{\text{ca}}$ .

### 4.3 Dynamics of $\theta_k$ : Correlated Packet Delays and Losses

We now move our attention to the case of correlated packet delays and losses, modeled by a time-homogeneous Markov chain with states  $\{0, 1, \dots, M\}$  and transition matrix  $\mathbf{P}^{\text{ca}}$  as introduced in Section 3.2.2. Unfortunately,  $\theta_k$  forms no longer a Markov chain if  $\tau_k^{\text{ca}}$  is Markov chain.<sup>1</sup> The reason for this stems from the fact that by (4.3) and the actuator buffering procedure (3.7), the different values of  $\theta_k$  depend on the delays and losses experienced by a different number of control sequences  $\underline{U}_k, \dots, \underline{U}_{k-i}$ . For example, it holds  $\theta_k = 1 \Leftrightarrow \tau_k^{\text{ca}} > 0$  and  $\tau_{k-1}^{\text{ca}} \leq 1$ , but  $\theta_k = 2 \Leftrightarrow \tau_k^{\text{ca}} > 0, \tau_{k-1}^{\text{ca}} > 1$ , and  $\tau_{k-2}^{\text{ca}} \leq 2$ .

Fundamentally, the connection between  $\theta_k$  and the delays and losses is formalized by considering the expanded Markov chain  $\underline{\tau}_k^{\text{ca}} = (\tau_k^{\text{ca}}, \tau_{k-1}^{\text{ca}}, \dots, \tau_{k-(N-1)}^{\text{ca}})$ , where each state is an  $N$ -tuple of successive states of  $\tau_k^{\text{ca}}$ . Then,  $\theta_k$  induces a partitioning of the state space of  $\underline{\tau}_k^{\text{ca}}$  into  $N + 1$  disjoint clusters  $\mathcal{C}_j$  given by

$$\begin{aligned} \mathcal{C}_0 &= \{\underline{\tau}_k^{\text{ca}} \mid \tau_k^{\text{ca}} = 0\}, \\ \mathcal{C}_j &= \{\underline{\tau}_k^{\text{ca}} \mid \tau_k^{\text{ca}} > 0, \tau_{k-1}^{\text{ca}} > 1, \dots, \tau_{k-(j-1)}^{\text{ca}} > j-1, \tau_{k-j}^{\text{ca}} \leq j\}, \\ \mathcal{C}_N &= \{\underline{\tau}_k^{\text{ca}} \mid \tau_k^{\text{ca}} > 0, \tau_{k-1}^{\text{ca}} > 1, \dots, \tau_{k-(N-1)}^{\text{ca}} > N-1\}, \end{aligned} \quad (4.13)$$

with  $j = 1, \dots, N-1$ . For  $N = M = 3$  the resulting partitioning of the 64 states of  $\underline{\tau}_k^{\text{ca}}$  into four clusters is exemplified in Table 4.1. With (4.13), we identify

$$\theta_k = i \Leftrightarrow \underline{\tau}_k^{\text{ca}} \in \mathcal{C}_i, \quad (4.14)$$

i.e., modes express cluster membership and a transition from mode  $i$  to  $j$  corresponds to a transition from cluster  $\mathcal{C}_i$  to  $\mathcal{C}_j$ . Yet, the transition probabilities depend on the initial distribution of  $\tau_k^{\text{ca}}$  and on the membership history since

$$\begin{aligned} \mathbb{P}[\underline{\tau}_{k+1}^{\text{ca}} \in \mathcal{C}_0 \mid \underline{\tau}_k^{\text{ca}} \in \mathcal{C}_1] &= \mathbb{P}[\tau_{k+1}^{\text{ca}} = 0 \mid \tau_k^{\text{ca}} > 0], \\ \mathbb{P}[\underline{\tau}_{k+1}^{\text{ca}} \in \mathcal{C}_1 \mid \underline{\tau}_k^{\text{ca}} \in \mathcal{C}_1, \underline{\tau}_{k-1}^{\text{ca}} \in \mathcal{C}_1] &\neq \mathbb{P}[\underline{\tau}_{k+1}^{\text{ca}} \in \mathcal{C}_1 \mid \underline{\tau}_k^{\text{ca}} \in \mathcal{C}_1, \underline{\tau}_{k-1}^{\text{ca}} \in \mathcal{C}_0], \end{aligned}$$

<sup>1</sup>In fact,  $\theta_k$  forms a Markov chain if  $\mathbf{P}^{\text{ca}}$  has identical rows, meaning that  $\tau_{k+1}^{\text{ca}}$  is independent of  $\tau_k^{\text{ca}}$ . In such cases, however, the results presented in the sequel trivially hold.

**Table 4.1:** The partitioning of the state space of  $\tau_k^{\text{ca}}$  induced by  $\theta_k$  for  $N = M = 3$ .

Cluster	$\tau_k^{\text{ca}} = (\tau_k^{\text{ca}}, \tau_{k-1}^{\text{ca}}, \tau_{k-2}^{\text{ca}})$								$\theta_k$
$\mathcal{C}_0$	(0, 0, 0)	(0, 0, 1)	(0, 0, 2)	(0, 0, 3)	(0, 1, 0)	(0, 1, 1)	(0, 1, 2)	(0, 1, 3)	0
	(0, 2, 0)	(0, 2, 1)	(0, 2, 2)	(0, 2, 3)	(0, 3, 0)	(0, 3, 1)	(0, 3, 2)	(0, 3, 3)	
$\mathcal{C}_1$	(1, 0, 0)	(1, 0, 1)	(1, 0, 2)	(1, 0, 3)	(1, 1, 0)	(1, 1, 1)	(1, 1, 2)	(1, 1, 3)	1
	(2, 0, 0)	(2, 0, 1)	(2, 0, 2)	(2, 0, 3)	(2, 1, 0)	(2, 1, 1)	(2, 1, 2)	(2, 1, 3)	
	(3, 0, 0)	(3, 0, 1)	(3, 0, 2)	(3, 0, 3)	(3, 1, 0)	(3, 1, 1)	(3, 1, 2)	(3, 1, 3)	
$\mathcal{C}_2$	(1, 2, 0)	(1, 2, 1)	(1, 2, 2)	(1, 3, 0)	(1, 3, 1)	(1, 3, 2)	(2, 2, 0)	(2, 2, 1)	2
	(2, 2, 2)	(2, 3, 0)	(2, 3, 1)	(2, 3, 2)	(3, 2, 0)	(3, 2, 1)	(3, 2, 2)	(3, 3, 0)	
	(3, 3, 1)	(3, 3, 2)							
$\mathcal{C}_3$	(1, 2, 3)	(1, 3, 3)	(2, 2, 3)	(2, 3, 3)	(3, 2, 3)	(3, 3, 3)			3

implying that the clusters do not form a Markov chain. In mathematical terms, the reason for this is that the expanded chain  $\tau_k^{\text{ca}}$  is not *lumpable* with respect to the partition  $\{\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_N\}$  [128, 129]. Thus,  $\theta_k$  cannot be a Markov chain and the dynamics (4.1) is not a MJLS.

If we replace  $\theta_k$  in (4.5) by  $\tau_k^{\text{ca}}$  according to

$$\underline{u}_k = \check{\mathbf{H}}^{(\tau_k^{\text{ca}})} \underline{\eta}_k + \check{\mathbf{J}}^{(\tau_k^{\text{ca}})} \underline{U}_k,$$

with

$$\begin{aligned} \check{\mathbf{H}}^{(\tau_k^{\text{ca}})} &= \left[ \mathbf{1}_{\{\tau_k^{\text{ca}} \in \mathcal{C}_1\}} \mathbf{I}_{n_u} \mathbf{0}_{n_u \times (N-2)n_u} \mathbf{1}_{\{\tau_k^{\text{ca}} \in \mathcal{C}_2\}} \mathbf{I}_{n_u} \mathbf{0}_{n_u \times (N-3)n_u} \cdots \mathbf{1}_{\{\tau_k^{\text{ca}} \in \mathcal{C}_{N-1}\}} \mathbf{I}_{n_u} \right], \\ \check{\mathbf{J}}^{(\tau_k^{\text{ca}})} &= \left[ \mathbf{1}_{\{\tau_k^{\text{ca}} \in \mathcal{C}_0\}} \mathbf{I}_{n_u} \mathbf{0}_{n_u \times (N-1)n_u} \right], \end{aligned}$$

we arrive at the slightly different MJLS

$$\begin{aligned} \underline{\psi}_{k+1} &= \check{\mathbf{A}}^{(\tau_k^{\text{ca}})} \underline{\psi}_k + \check{\mathbf{B}}^{(\tau_k^{\text{ca}})} \underline{U}_k + \tilde{w}_k, \\ \underline{y}_k &= [\mathbf{C} \mathbf{0}] \underline{\psi}_k + v_k, \end{aligned} \quad (4.15)$$

with

$$\check{\mathbf{A}}^{(\tau_k^{\text{ca}})} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \check{\mathbf{H}}^{(\tau_k^{\text{ca}})} \\ \mathbf{0} & \mathbf{F} \end{bmatrix}, \quad \check{\mathbf{B}}^{(\tau_k^{\text{ca}})} = \begin{bmatrix} \mathbf{B} \check{\mathbf{J}}^{(\tau_k^{\text{ca}})} \\ \mathbf{G} \end{bmatrix}. \quad (4.16)$$

In (4.15), the switchings between the modes are directly governed by the network model, and, in principle, this model could replace (4.1). However, (4.15) has  $(M+1)^N$  modes, which is exponential in the sequence length, whereas (4.1) has only  $N+1$ . For example, for  $N = M = 4$ , (4.15) has already 625 modes but (4.1) has only five. The resulting computational burden might be prohibitively high even for small  $N$ , in particular with regards to the model predictive controllers we develop in Chapter 5. The practical usage of the MJLS (4.15) is thus limited. Inspection of (4.16) further exhibits that (4.15) is redundant because  $\check{\mathbf{H}}^{(\tau_k^{\text{ca}})}$  and  $\check{\mathbf{J}}^{(\tau_k^{\text{ca}})}$ , and, consequently,  $\check{\mathbf{A}}^{(\tau_k^{\text{ca}})}$  and  $\check{\mathbf{B}}^{(\tau_k^{\text{ca}})}$ , are equal for all values of  $\tau_k^{\text{ca}}$  belonging to the same cluster. As the above discussion showed, the cluster membership of  $\tau_k^{\text{ca}}$  entirely determines the switchings between the operation modes. Its actual value is only of secondary importance.

Based on this insight, the track we pursue to obtain a model of the form and complexity of (4.1) is to compute a Markov chain model for  $\theta_k$  that approximates the true mode dynamics, that is, the dynamics of the induced clusters  $\mathcal{C}_j$  in the expanded chain  $\tau_k^{\text{ca}}$ . Specifically, this model shall be such that the approximation error vanishes as  $k$  increases. For this approximation to be possible, the asymptotic behavior of  $\tau_k^{\text{ca}}$  must be well defined. The following lemma ensures that this is indeed the case. Before

we present it, let us first introduce a multi-index notation that we will use extensively in the remainder of this section. By definition,  $\underline{\tau}_k^{\text{ca}}$  has  $d = (M + 1)^N$  distinct states, each of which is an  $N$ -tuple of successive states of  $\tau_k^{\text{ca}}$ . We shall use  $N$ -dimensional multi-indices to enumerate the tuples, i.e., the notation

$$\underline{\tau}_k^{\text{ca}} = (i_0, i_1, \dots, i_{N-1}),$$

indicates the tuple composed of the successive states  $\tau_k^{\text{ca}} = i_0, \tau_{k-1}^{\text{ca}} = i_1, \dots, \tau_{k-(N-1)}^{\text{ca}} = i_{N-1}$ .

Also, let us recall that we assumed all transition probabilities  $p_{ij}^{\text{ca}}$  of  $\tau_k^{\text{ca}}$  to be positive, i.e.,  $\mathbf{P}^{\text{ca}} > 0$  (cf. Assumption 3.5). The Perron-Frobenius theorem (cf. Appendix A.6) then ensures that  $\tau_k^{\text{ca}}$  has a unique stationary distribution  $\underline{p} = [p^{(0)} \ p^{(1)} \ \dots \ p^{(M)}]^T$  and that  $\underline{p} > 0$ , i.e., all entries of  $\underline{p}$  are positive.

**Lemma 4.1:**

For every initial distribution, the expanded chain  $\underline{\tau}_k^{\text{ca}}$  converges to its unique stationary distribution  $\underline{\bar{p}}$  with elements  $\bar{p}^{(i_0, i_1, \dots, i_{N-1})}$  given by

$$\bar{p}^{(i_0, i_1, \dots, i_{N-1})} = p^{(i_{N-1})} p_{i_{N-1} i_{N-2}}^{\text{ca}} p_{i_{N-2} i_{N-3}}^{\text{ca}} \cdots p_{i_1 i_0}^{\text{ca}}, \quad (4.17)$$

with  $i_0, i_1, \dots, i_{N-1} \in \{0, \dots, M\}$ , and where  $\underline{p} = [p^{(0)} \ p^{(1)} \ \dots \ p^{(M)}]^T$  is the stationary distribution of  $\tau_k^{\text{ca}}$ .

*Proof.* The proof is provided in Appendix B.2. □

Denote the transition matrix of  $\underline{\tau}_k^{\text{ca}}$ , whose entries are given by (B.2), by  $\tilde{\mathbf{P}}^{\text{ca}}$ . Then, the above result implies (again by the Perron-Frobenius theorem) that the  $k$ -th powers of  $\tilde{\mathbf{P}}^{\text{ca}}$  converge to a matrix in which all rows are equal and given by  $\underline{\bar{p}}$ , that is,

$$\lim_{k \rightarrow \infty} \left( \tilde{\mathbf{P}}^{\text{ca}} \right)^k = \underline{\mathbb{1}}_d \underline{\bar{p}}^T, \quad (4.18)$$

with  $\underline{\mathbb{1}}_d$  the  $d$ -dimensional vector of ones.

Reducing the state space of a Markov chain is known as *Markov chain lumping* in the literature [128, 130]. Here, one attempts to aggregate states to obtain a simpler model with fewer states such that specific properties of the original chain are retained [131]. Applications of lumping are widespread and range from the natural sciences and finance, where Markov chains with huge state spaces are typical [132–136], over performance and network modeling [137, 138], to networked estimation and the model reduction of hybrid systems [139, 140]. We will use this concept to aggregate the  $(M + 1)^N$  states of  $\underline{\tau}_k^{\text{ca}}$  into a *lumped dynamics* with only  $N + 1$  states.

The basic idea to obtain the lumped dynamics for  $\theta_k$  can be explained as follows. For any initial distribution  $\tilde{\underline{p}}_0$  of  $\underline{\tau}_k^{\text{ca}}$ , the probability  $\mu_k^{(i)}$  that  $\underline{\tau}_k^{\text{ca}}$  is in cluster  $\mathcal{C}_i$ , and hence the probability that  $\theta_k = i$  (cf. (4.14)), can be computed by first propagating  $\tilde{\underline{p}}_0$  from the initial time step to time  $k$  to obtain  $\tilde{\underline{p}}_k$  and then adding up the probabilities of all states forming the cluster. Mathematically, this is expressed with  $\underline{\mu}_k = [\mu_k^{(0)} \ \mu_k^{(1)} \ \dots \ \mu_k^{(N)}]^T$  as

$$\underline{\mu}_k^T = \tilde{\underline{p}}_k^T \bar{\mathbf{V}} = \tilde{\underline{p}}_0^T \left( \tilde{\mathbf{P}}^{\text{ca}} \right)^k \bar{\mathbf{V}}, \quad (4.19)$$

where  $\bar{\mathbf{V}} \in \mathbb{R}^{d \times N+1}$  is the collection matrix with elements

$$\bar{v}_{(i_0, i_1, \dots, i_{N-1})j} = \begin{cases} 1 & (i_0, i_1, \dots, i_{N-1}) \in \mathcal{C}_j \\ 0 & (i_0, i_1, \dots, i_{N-1}) \notin \mathcal{C}_j \end{cases}. \quad (4.20)$$

That is, the  $j$ -th element of the  $(i_0, i_1, \dots, i_{N-1})$ -th row of  $\bar{\mathbf{V}}$  is one if  $\mathcal{I}_k^{\text{ca}} = (i_0, i_1, \dots, i_{N-1}) \in \mathcal{C}_j$  and zero otherwise. By Lemma 4.1 and (4.19), the unique stationary cluster probabilities  $\underline{\mu}$  are then

$$\underline{\mu}^{\text{T}} = \underline{\bar{p}}^{\text{T}} \bar{\mathbf{V}} = \lim_{k \rightarrow \infty} \tilde{\underline{p}}_0^{\text{T}} \left( \tilde{\mathbf{P}}^{\text{ca}} \right)^k \bar{\mathbf{V}}. \quad (4.21)$$

On the other hand, if  $\theta_k$  was a Markov chain with transition matrix  $\mathbf{T}$ , we would obtain the very same probabilities if we first added up the initial probabilities of all states forming a cluster and then propagated the resultant probabilities  $\underline{\mu}_0$  in time, i.e.,

$$\underline{\mu}_k^{\text{T}} = \underline{\mu}_0^{\text{T}} \mathbf{T}^k = \tilde{\underline{p}}_0^{\text{T}} \bar{\mathbf{V}} \mathbf{T}^k. \quad (4.22)$$

However, as we have seen, such  $\mathbf{T}$  does not exist, and any Markov chain model (4.22) can only approximate the true probabilities (4.19), that is, it holds  $\tilde{\underline{p}}_0^{\text{T}} \bar{\mathbf{V}} \mathbf{T}^k \neq \tilde{\underline{p}}_0^{\text{T}} \left( \tilde{\mathbf{P}}^{\text{ca}} \right)^k \bar{\mathbf{V}}$ .

The best we can do is to find an approximation such that the approximation error vanishes as  $k$  increases, i.e., such that  $\tilde{\underline{p}}_0^{\text{T}} \bar{\mathbf{V}} \mathbf{T}^k \rightarrow \underline{\mu}$  for  $k \rightarrow \infty$ . By (4.21) and (4.22) this necessitates that we must choose  $\mathbf{T}$  such that

$$\begin{aligned} \lim_{k \rightarrow \infty} \bar{\mathbf{V}} \mathbf{T}^k &\stackrel{!}{=} \lim_{k \rightarrow \infty} \left( \tilde{\mathbf{P}}^{\text{ca}} \right)^k \bar{\mathbf{V}} \\ &= \underline{\mathbb{1}}_d \underline{\bar{p}}^{\text{T}} \bar{\mathbf{V}}, \end{aligned} \quad (4.23)$$

where the second equality is due to (4.18). But by the definition of  $\bar{\mathbf{V}}$  (cf. (4.20)) we have  $\bar{\mathbf{V}} \underline{\mathbb{1}}_{N+1} = \underline{\mathbb{1}}_d$ , so that (4.23) can be rewritten by virtue of

$$\lim_{k \rightarrow \infty} \bar{\mathbf{V}} \mathbf{T}^k \stackrel{!}{=} \bar{\mathbf{V}} \underline{\mathbb{1}}_{N+1} \underline{\bar{p}}^{\text{T}} \bar{\mathbf{V}},$$

or, equivalently,

$$\lim_{k \rightarrow \infty} \mathbf{T}^k \stackrel{!}{=} \underline{\mathbb{1}}_{N+1} \underline{\bar{p}}^{\text{T}} \bar{\mathbf{V}},$$

because  $\bar{\mathbf{V}}$  has full rank since the clusters are disjoint.

Hence, we must choose  $\mathbf{T}$  such that its powers  $\mathbf{T}^k$  converge to a matrix in which all rows are given by  $\underline{\bar{p}}^{\text{T}} \bar{\mathbf{V}}$ . In other words, we must determine  $\mathbf{T}$  such that the lumped dynamics converges to its unique stationary distribution  $\bar{\mathbf{V}}^{\text{T}} \underline{\bar{p}}$ . As in [128, 141], this is achieved with a lumped dynamics of the form

$$\mathbf{T} = \bar{\mathbf{U}} \tilde{\mathbf{P}}^{\text{ca}} \bar{\mathbf{V}}, \quad (4.24)$$

with distribution matrix  $\bar{\mathbf{U}} \in \mathbb{R}^{N+1 \times d}$ . Specifically, we set the elements of  $\bar{\mathbf{U}}$  to

$$\bar{u}_{j(i_0, i_1, \dots, i_{N-1})} = \begin{cases} \frac{\bar{p}^{(i_0, i_1, \dots, i_{N-1})}}{\sum_{(l_0, l_1, \dots, l_{N-1}) \in \mathcal{C}_j} \bar{p}^{(l_0, l_1, \dots, l_{N-1})}} & (i_0, i_1, \dots, i_{N-1}) \in \mathcal{C}_j \\ 0 & (i_0, i_1, \dots, i_{N-1}) \notin \mathcal{C}_j \end{cases},$$

i.e., the  $(i_0, i_1, \dots, i_{N-1})$ -th element of the  $j$ -th row of  $\bar{\mathbf{U}}$  is the stationary distribution of  $\mathcal{I}_k^{\text{ca}}$  restricted to the states belonging to the same cluster  $\mathcal{C}_j$ , normalized such that the row sum is one. This yields the desired result, confirmed by the theorem below.

#### Theorem 4.2:

*The stationary distribution of the lumped dynamics (4.24) for  $\theta_k$  is unique and given by  $\bar{\mathbf{V}}^{\text{T}} \underline{\bar{p}}$ , where  $\underline{\bar{p}}$  is the stationary distribution (4.17) of  $\mathcal{I}_k^{\text{ca}}$  and  $\bar{\mathbf{V}}$  is given by (4.20). Moreover,*

$$\lim_{k \rightarrow \infty} \underline{\mu}_0^{\text{T}} \mathbf{T}^k = \lim_{k \rightarrow \infty} \underline{\mu}_0^{\text{T}} \left( \bar{\mathbf{U}} \tilde{\mathbf{P}}^{\text{ca}} \bar{\mathbf{V}} \right)^k = \underline{\bar{p}}^{\text{T}} \bar{\mathbf{V}}, \quad (4.25)$$

for any initial distribution  $\underline{\mu}_0$ .

*Proof.* We provide the proof of this result in Appendix B.3.  $\square$

For correlated packet delays and losses, we have thus established two things. First, we can use the lumped dynamics (4.24) for  $\theta_k$  to approximate the true augmented dynamics (4.15) by the MJLS (4.1) that was originally derived for independent packet delays and losses. This significantly reduces the number of operation modes to only  $N + 1$  and comes with a substantial reduction of computational complexity. Second, the lumped dynamics ensures that the approximation error vanishes in the long run, which means that for  $\underline{\psi}'_0 = \underline{\psi}_0$  and

$$\begin{aligned}\underline{\psi}_{k+1} &= \check{\mathbf{A}}^{(\tau_k^{\text{ca}})} \underline{\psi}_k + \check{\mathbf{B}}^{(\tau_k^{\text{ca}})} \underline{U}_k + \check{\underline{w}}_k, \\ \underline{\psi}'_{k+1} &= \check{\mathbf{A}}^{(\theta_k)} \underline{\psi}'_k + \check{\mathbf{B}}^{(\theta_k)} \underline{U}'_k + \check{\underline{w}}'_k,\end{aligned}$$

with  $\check{\mathbf{A}}^{(\tau_k^{\text{ca}})}$ ,  $\check{\mathbf{B}}^{(\tau_k^{\text{ca}})}$  as in (4.16) and  $\check{\mathbf{A}}^{(\theta_k)}$ ,  $\check{\mathbf{B}}^{(\theta_k)}$  as in (4.10), we have  $\underline{\psi}_k - \underline{\psi}'_k \rightarrow \underline{0}$  as  $k$  increases, provided  $\underline{U}_k \equiv \underline{U}'_k$  and  $\check{\underline{w}}_k \equiv \check{\underline{w}}'_k$ . Consequently, we can expect that control sequences computed by a control algorithm based on the dynamics (4.1) yield similar performance as those computed by the same algorithm based on (4.15).

As for independent packet delays and losses, the transition matrix  $\mathbf{T}$  of the lumped dynamics is always a lower Hessenberg matrix. However, the other properties discovered in Section 4.2 do not generalize. In particular,  $\mathbf{T}$  is no longer determined by its last row.

To conclude this section, we want to stress that the  $d$ -by- $d$  transition matrix  $\check{\mathbf{P}}^{\text{ca}}$  of the expanded chain  $\tau_k^{\text{ca}}$  is not needed explicitly to compute the lumped dynamics (4.24) because each nonzero entry equals an entry of  $\mathbf{P}^{\text{ca}}$  (cf. (B.2)). Also, the computation of the stationary distribution  $\bar{p}$  given by (4.17) is relatively cheap. If  $\bar{p}$  is available, only  $N$  multiplications per element are needed. The computation of  $\bar{p}$  itself has complexity  $\mathcal{O}(M^3)$  if a direct method is used or  $\mathcal{O}(M^2z)$  if the power method is used. Here,  $z$  is the number of iterations that depends on the magnitude of the second largest eigenvalue of  $\mathbf{P}^{\text{ca}}$  [142].

## 4.4 Discussion of the Modeling Approach

Over the years, Markov jump linear systems have proven to be a powerful tool for modeling and control in various applications [127, 143]. Successful applications range from networked control, as already mentioned at the beginning of this chapter, over target tracking [144–146] to control systems with component failures or abrupt environmental disturbances [147–151]. Thus, we can leverage a solid theoretical foundation which makes the augmented dynamics (4.1) very appealing for the design and analysis of sequence-based controllers. However, one might argue that the additional variables in (4.1),  $\underline{\eta}_k$  and  $\theta_k$ , are superfluous. Indeed, a way to express the plant dynamics (3.1) and the actuator's buffering procedure (3.7) without  $\underline{\eta}_k$  and  $\theta_k$  is [152]<sup>2</sup>

$$\begin{aligned}\underline{x}_{k+1} &= \mathbf{A}\underline{x}_k + \mathbf{B}\underline{u}_k + \underline{w}_k, \\ f_k^u(\underline{u}_k) &= \sum_{i=0}^{N-1} \omega_k^{(i)} \delta(\underline{u}_k - \underline{u}_{k|k-i}) + \omega_k^{(N)} \delta(\underline{u}_k - \underline{u}_k^{\text{df}}),\end{aligned}\tag{4.26}$$

where  $\delta(\cdot)$  denotes the Dirac delta function. The gist of this model is to understand the actual plant input  $\underline{u}_k$  as a random variable with probability density function (PDF)  $f_k^u$  rather than the output of an (artificial) dynamical system with state  $\underline{\eta}_k$  (cf. (4.5)).  $f_k^u$  is a Dirac mixture density with  $N + 1$

<sup>2</sup>For simplicity, we leave out the measurement equation.

components, because  $\underline{u}_k$  can attain only  $N + 1$  different values as per (4.2). Each component of the mixture corresponds to one of the possible control inputs, and the weighting factors  $\omega_k^{(i)}$  given by

$$\omega_k^{(i)} = \mathbb{P}[\underline{u}_k = \underline{u}_{k|k-i}], \quad \omega_k^{(N)} = \mathbb{P}[\underline{u}_k = \underline{u}_k^{\text{df}}], \quad (4.27)$$

denote the probability that the respective input is actually applied. These probabilities depend on the network model  $\tau_k^{\text{ca}}$  and are computed according to (cf. (4.13))

$$\begin{aligned} \omega_k^{(0)} &= \mathbb{P}[\tau_k^{\text{ca}} = 0], \\ \omega_k^{(i)} &= \mathbb{P}[\tau_k^{\text{ca}} > 0, \tau_{k-1}^{\text{ca}} > 1, \dots, \tau_{k-(i-1)}^{\text{ca}} > i-1, \tau_{k-i}^{\text{ca}} \leq i], \\ \omega_k^{(N)} &= \mathbb{P}[\tau_k^{\text{ca}} > 0, \tau_{k-1}^{\text{ca}} > 1, \dots, \tau_{k-(N-1)}^{\text{ca}} > N-1]. \end{aligned} \quad (4.28)$$

Hence, they express the dependence of the plant dynamics on the network. To calculate the weighting factors, no assumptions about  $\tau_k^{\text{ca}}$  are needed, it suffices that the probabilities  $\mathbb{P}[\tau_k^{\text{ca}} = i]$  are computable by the controller for all  $i = 0, 1, \dots, M$  at every time step  $k$ .

In this regard, (4.26) seems more general than the augmented dynamics (4.1), where the additional assumptions about  $\tau_k^{\text{ca}}$  were pivotal to obtain a dynamics for  $\theta_k$ . In (4.1), however, the dependence of the plant state on the network becomes directly visible by  $\theta_k$ . This explicit coupling facilitates the investigation of closed-loop properties such as stability and allows the application of (approximate) dynamic programming to obtain control laws that are optimal with regards to some cost function [153, 154]. Additionally, the actuator feedback (i.e., the ACKs) is simpler incorporated into the augmented dynamics (4.1) than into the Dirac mixture  $f_k^u$  because it directly exhibits past system modes. When, as in the example in Section 4.1, the controller infers from  $\text{ACK}_{k-2}$  that  $\underline{u}_{k-2} = \underline{u}_{k-2|k-5}$ , we get  $\theta_{k-2} = 3$  and thus  $\underline{\mu}_{k-2} = \underline{e}_{N+1}^{(3)}$ , where  $\underline{e}_n^{(i)}$  denotes the  $i$ -th standard basis vector in  $\mathbb{R}^n$ . By (4.12), the mode probabilities at time  $k$  then directly become  $\underline{\mu}_k = (\mathbf{T}^2)^T \underline{e}_{N+1}^{(3)}$ . Regarding (4.26), the weighting factors of the Dirac mixture  $f_k^u$  are no longer given by (4.28) if actuator feedback is available. For instance, given that  $\underline{u}_{k-2} = \underline{u}_{k-2|k-5}$  is known, they become

$$\begin{aligned} \omega_k^{(0)} &= \mathbb{P}[\tau_k^{\text{ca}} = 0], \\ \omega_k^{(i)} &= \mathbb{P}[\tau_k^{\text{ca}} > 0, \tau_{k-1}^{\text{ca}} > 1, \dots, \tau_{k-(i-1)}^{\text{ca}} > i-1, \tau_{k-i}^{\text{ca}} \leq i | \underline{u}_{k-2} = \underline{u}_{k-2|k-5}], \\ \omega_k^{(N)} &= \mathbb{P}[\tau_k^{\text{ca}} > 0, \tau_{k-1}^{\text{ca}} > 1, \dots, \tau_{k-(N-1)}^{\text{ca}} > N-1 | \underline{u}_{k-2} = \underline{u}_{k-2|k-5}]. \end{aligned}$$

In general, the probabilities on the right side of (4.28) must be conditioned on the information available from all received ACKs

$$\begin{aligned} \omega_k^{(0)} &= \mathbb{P}[\tau_k^{\text{ca}} = 0], \\ \omega_k^{(i)} &= \mathbb{P}[\tau_k^{\text{ca}} > 0, \tau_{k-1}^{\text{ca}} > 1, \dots, \tau_{k-(i-1)}^{\text{ca}} > i-1, \tau_{k-i}^{\text{ca}} \leq i | \mathcal{A}_0, \dots, \mathcal{A}_k], \\ \omega_k^{(N)} &= \mathbb{P}[\tau_k^{\text{ca}} > 0, \tau_{k-1}^{\text{ca}} > 1, \dots, \tau_{k-(N-1)}^{\text{ca}} > N-1 | \mathcal{A}_0, \dots, \mathcal{A}_k], \end{aligned}$$

which makes their computation not straightforward.

On the other hand, the augmented dynamics (4.1) might be less suited than (4.26) when it comes to the development of *event-triggered* and *self-triggered control* approaches. The general idea of these approaches is to minimize the amount of communication needed to guarantee a certain level of performance, and new control inputs are transmitted to the plant only if a predefined triggering condition is violated [155–157]. Not transmitting a control sequence at every time step, however, implies that the state space of  $\theta_k$  varies over time. For example, if the controller does not send the

sequence  $\underline{U}_{k-1}$ , then the modes  $\theta_{k-1} = 0, \theta_k = 1, \theta_{k+1} = 2, \dots, \theta_{k+N-2} = N - 1$  as such do not exist. Thus, the transition matrix of  $\theta_k$  is necessarily time-varying. For independent packet delays and losses, this poses no additional difficulties, and we can adapt Theorem 4.1 in the same way as Lemma 3.1 of [123] was adapted in [123, Chapter 5]. The resulting transition matrices, however, will have zero rows and columns, reflecting that the state space of  $\theta_k$  varies over time. However, the lumped dynamics for  $\theta_k$  derived in Section 4.3 for correlated delays and losses is time-homogeneous by construction. Hence, to take the time-varying nature into account, the controller must directly manipulate the mode probabilities. Continuing the above example, if  $\underline{U}_{k-1}$  is not sent, the controller must zero the corresponding mode probabilities at time  $k$  and later, i.e.,

$$\mu_{k-1}^{(0)} = \mu_k^{(1)} = \dots = \mu_{k+N-2}^{(N-1)} = 0,$$

and then renormalize  $\underline{\mu}_{k-1}, \underline{\mu}_k, \dots, \underline{\mu}_{k+N-2}$ . This manipulation, however, implies that typically  $\underline{\mu}_{k+1} \neq \mathbf{T}^T \underline{\mu}_k$ . Thus, it might be simpler to use (4.26) as a starting point for the design of event-triggered or self-triggered controllers and directly work with the input probabilities (4.27).

To conclude this section, we emphasize that neither of the two models (4.1) and (4.26) allows the computation of control sequences that are globally optimal with respect to some cost function. As already indicated at the beginning of Chapter 3, this is inherent to the considered setup and due to the potential loss of acknowledgment packets [90, 96, 98]. It is a well-known result that even for a quadratic cost function the optimal control sequence  $\underline{U}_k^*$  is generally a nonlinear function of the information  $\mathcal{I}_k$  available to the controller.<sup>3</sup> In particular, this implies that the optimal solution does not exhibit a separation between estimation and control, i.e., providing the controller with the optimal state estimate  $\hat{x}_k = \mathbb{E}\{x_k | \mathcal{I}_k\}$  rather than  $\mathcal{I}_k$  is always suboptimal [158]. The reason for this lies in the controller's imperfect knowledge of the buffered control sequences and consequently the applied control inputs. As we show in the next chapter, this results in a so-called *dual effect*: Each control sequence affects the plant state (namely, the desired control action) and also the controller's future uncertainty of the plant state (i.e., the future estimation error) [159]. The latter, however, also determines the quality of the computed control sequences – the controller makes better decisions when the estimation error is low [158]. Hence, an optimal control algorithm must achieve the control objective and excite the plant such that its uncertainty of the plant state is reduced. These tasks, however, are typically antagonistic.

## 4.5 Conclusions

In this chapter, we presented the first contribution of this thesis, which lays the groundwork for the sequence-based control algorithms that we will develop in Chapter 5 and Chapter 6. Our goal was to find a model that accurately reflects the influence of the shared usage of the available communication resources, which enables networked controllers to take changing communication conditions into account during the computation of new control sequences. To that end, we sought a model to describe the impact of the most relevant influencing factors on the control task, namely the occurring packet delays and losses. Moreover, we wanted a flexible model supporting the different probabilistic representations of the packet delays and losses provided by the CoCPN translator.

To achieve this goal, we built upon a result presented in [123], where it was shown that sequence-based control of linear plants with dynamics (3.1) can be addressed using an augmented dynamical system of the form (4.1). More precisely, it was proved that for control sequences of length  $N$ , (4.1) becomes a Markov jump linear system with  $N + 1$  operation modes if one assumes that the delays and losses that

<sup>3</sup>The available information set  $\mathcal{I}_k$  is made precise in the next chapter.

occur during the transmission of the control sequences be governed by an independent and stationary random process  $\tau_k^{ca}$ . Our key discovery leading to the results of this chapter was that the same MJLS also arises when this assumption is jettisoned and replaced by either of the two representations provided by the CoCPN translator. Starting from the definition of the augmented state  $\underline{\psi}_k$ , we first confirmed this for the case that  $\tau_k^{ca}$  is still an independent process but no longer stationary as specified in Section 3.2.1. However, confirming this for the second representation provided by the CoCPN translator, rendering  $\tau_k^{ca}$  a Markov chain with the properties discussed in Section 3.2.2, turned out to be trickier. In a first step, we showed that the resulting augmented dynamical system is a MJLS but with significantly more operation modes (cf. (4.15)). To obtain the dynamics (4.1), we then employed Markov chain lumping to aggregate the  $(M + 1)^N$  operation modes into  $N + 1$  modes.

In the course of our discussions, we also detailed that only a subset of the complete mode history is known to the controller due to the lack of reliable feedback from the actuator. This is a direct consequence of our decision to take delays and losses during the transmission of acknowledgment packets into account. While this decision allows for a more realistic treatment of the impact of the occurring communication imperfections compared to the vast body of related research, it precludes us from deriving optimal control laws even for quadratic cost functions. We shall demonstrate this issue in the next chapter, where we propose two novel approaches for sequence-based stochastic model predictive control.

We conclude this chapter by emphasizing a valuable by-product of our results. Existing implementations of sequence-based controllers that were originally designed under the assumption of independent packet delays and losses occurring with fixed probabilities, such as the optimal control law for trajectory tracking presented in [123], can be readily used in the setup considered in this thesis without adaptation or increase of computational complexity.



## Sequence-Based Stochastic Model Predictive Control

Model predictive control (MPC) is a popular control approach in many fields, such as process control, path planning, autonomous driving, or supply chain management [160–163]. MPC translates the control objective into an optimization problem so that disturbances and uncertainties appearing in real-world applications can be explicitly integrated into the computation of the control inputs. In traditional MPC, one typically assumes unknown but bounded disturbances and performs worst-case analyses, which leads to robust but conservative results, and neglects the often probabilistic nature of the occurring uncertainties. To overcome these limitations, *stochastic model predictive control (SMPC)* has gained much attention in recent years, both from a theoretical and a practical point of view. Here, the general idea is to use techniques that have emanated from stochastic programming and stochastic optimal control for a consistent treatment of all uncertainties to exploit their statistical descriptions [164, 165].

Given the randomness of packet delays and losses, networked control tasks can inherently be addressed by SMPC. It is thus not surprising that many research results have been published in recent years. Focus, however, has been laid on packet losses rather than on delays. In [166] and [167], the authors address SMPC for linear plants where the control inputs are subject to Bernoulli losses, i.e., get lost according to a Bernoulli process.<sup>1</sup> Due to the immediate availability of ACKs, i.e., ACKs are delivered failure-free and without delays, the posed stochastic control problem is an easy-to-solve quadratic program. A similar setup is considered in [113], where the packet losses are governed by a Markov chain. In these works, however, state feedback is assumed, that is, the plant state  $x_k$  is known to the controller at every time step. This assumption is left out in [168]. There, the controller uses a Kalman filter to estimate  $x_k$  from noisy measurements that are sent over an unreliable network, which drops them according to a Bernoulli process. Again, however, failure-free delivery of ACKs from the actuator is assumed. Both Bernoulli and Markovian packet losses are covered by the SMPC formulation developed in [112]. Targeting unreliable communication in wireless environments, the authors modeled communication reliability via an underlying network state process with a finite state space.

---

<sup>1</sup>In other words, packet losses are independent of each other and occur with fixed probability  $p$ .

Sequence-based SMPC of a linear plant subject to Bernoulli losses was investigated in [63]. With arguments similar to those we used in Chapter 4, the authors derived an augmented dynamical system. Specifically for control sequences with  $N$  elements, they derived an MJLS with  $N + 1$  modes with mode transition probabilities solely determined by the packet loss probabilities. However, the authors assumed a TCP-like connection between controller and actuator. This assumption enabled them to find an analytical solution to the underlying control problem. Failure-free transmission of ACKs from the actuator was also assumed in [169], where the authors modeled packet delays and losses by an independent process with fixed delay and loss probabilities. This modeling approach was also used in [67]. A case study, where sequence-based SMPC is used in an application from process control, is presented in [170].

In this chapter, we deal with sequence-based SMPC for the NCS described in Chapter 3. Compared to the works discussed above, this demands that we address two additional challenges. First, because the plant state  $\underline{x}_k$  is not available, the controller must rely on the information provided by the received measurements, which are corrupted by stochastic uncertainties themselves. Control problems of this type are referred to as problems of *imperfect information* and are generally harder to solve than state feedback problems [153]. Second, the lack of reliable actuator feedback results in a dual effect, which forbids the analytical determination of optimal control laws and even makes their numerical computation intractable. We present two suboptimal approaches in the course of this chapter. The first approach draws inspiration from the connection between hybrid systems and *multiple model* estimation algorithms. We use this connection to derive a control algorithm that combines the output of multiple controllers, each of which corresponds to one operation mode of the augmented dynamics (4.1). The second approach tackles the control problem from a different perspective. Here, we re-express the underlying cost function in terms of the second moment of the plant state. Based on this reformulation, we derive an upper bound of the cost function and then develop an iterative procedure to minimize this bound in the neighborhood of a reference trajectory.

To facilitate the exposition, we present the results of this chapter under the assumption that  $\theta_k$  is a time-homogeneous Markov chain. Hence, throughout this chapter, we will always denote the mode transition probabilities by  $t_{ij}$ . We recall from Section 3.2 that the controller cannot anticipate how the information provided by the CoCPN translator, i.e., either the delay and loss probabilities or their transition probabilities, might change in the future. To predict the evolution of the plant state over a fixed number of time steps, it is thus reasonable to assume that they remain constant. This in turn implies that the mode transition probabilities  $t_{k,ij}, t_{k+1,ij}, \dots$  become time-invariant anyhow even if they are computed according to Theorem 4.1.

We start with a definition of the underlying stochastic optimal control problem in Section 5.1. Then, in Section 5.2 we show that the optimal solution to this problem is intractable. Sections 5.3 and 5.4 constitute the main contribution of this chapter and present the two proposed control approaches. Finally, we conclude the chapter with an evaluation in Section 5.5, where we compare the proposed approaches in different control tasks with two state-of-the-art approaches from the literature.

*This chapter is based on results presented in our publications [287, 288, 292, 296].*

## 5.1 Problem Formulation

In the considered NCS, the true plant state  $\underline{x}_k$  is not known to the controller. Instead, the controller must calculate the control sequences based on the received measurements, the received ACKs, and the prior knowledge about the initial plant state  $\underline{x}_0$ . More precisely, the information available to the

controller at time  $k$  is called *information set* and given by

$$\begin{aligned}\mathcal{I}_0 &= \{\hat{\mathbf{x}}_0, \mathbf{X}_0\}, \\ \mathcal{I}_k &= \mathcal{I}_{k-1} \cup \{\underline{U}_{k-1}, \mathcal{A}_k, \mathcal{Y}_k\}, \quad k = 1, 2, \dots\end{aligned}\quad (5.1)$$

**Remark 5.1:**

The plant dynamics (3.1), the network models  $\tau_k^{\text{ca}}$ ,  $\tau_k^{\text{ac}}$ , and  $\tau_k^{\text{sc}}$ , the initial conditions  $\theta_0 = N$  and  $\eta_0 = \underline{0}$ , and the actuator behavior, i.e., its buffering procedure and its default input strategy, are also known to the controller. For brevity, we did not add these pieces of information to the sets  $\mathcal{I}_k$ .

In contrast to the traditional deterministic MPC, in SMPC one typically does not solve the underlying optimal control problem directly with regards to control actions [171]. Rather, to account for the uncertainties, one is interested in *control laws*, functions  $\underline{\pi}_k$  that map the available information  $\mathcal{I}_k$  into the space of control actions [165, 172].

**Definition 5.1:**

A control sequence  $\underline{U}_k$  is admissible at time  $k$  if it is a function of the information set  $\mathcal{I}_k$ , i.e., if we can write  $\underline{U}_k = \underline{\pi}_k(\mathcal{I}_k)$  for some function  $\underline{\pi}_k$ . We then call  $\underline{\pi}_k$  a control law.

At every time step  $k$ , we seek control laws  $\underline{\pi}_k^*, \underline{\pi}_{k+1}^*, \dots, \underline{\pi}_{k+K-1}^*$  that minimize the cost function

$$\mathcal{J}(\underline{\pi}_{k:k+K-1}) = \underset{\substack{\underline{x}_k, \\ \tau_{k:k+K-1}^{\text{ca}}, \\ \tau_{k:k+K-1}^{\text{sc}}, \\ \tau_{k:k+K-1}^{\text{ac}}, \\ \underline{w}_{k:k+K-1}, \\ \underline{u}_{k:k+K-1}}}{\mathbb{E}} \left\{ \underline{x}_{k+K}^T \mathbf{Q}_K \underline{x}_{k+K} + \sum_{t=0}^{K-1} \underline{x}_{k+t}^T \mathbf{Q}_t \underline{x}_{k+t} + \underline{u}_{k+t}^T \mathbf{R}_t \underline{u}_{k+t} \mid \mathcal{I}_k \right\}, \quad (5.2)$$

where  $K \in \mathbb{N}$  is the horizon length, and  $\mathbf{Q}_t \in \mathbb{R}^{n_x \times n_x}$  and  $\mathbf{R}_t \in \mathbb{R}^{n_u \times n_u}$  are weighting matrices such that  $\mathbf{Q}_t \succcurlyeq 0$  for  $t = 0, 1, \dots, K$  and  $\mathbf{R}_t \succ 0$  for  $t = 0, 1, \dots, K-1$ . In (5.2), we used the abbreviation  $a_{i:j}$  to denote a sequence  $a_i, a_{i+1}, \dots, a_j$  of elements of the same kind and the expectation on the right side is with respect to all occurring random variables as indicated by the subscript. Note that it is equivalent to take the expectation with respect to  $\underline{x}_{k:k+K}$  and  $\underline{u}_{k:k+K-1}$ , which depend on the random variables in (5.2).

As we minimize the cost function (5.2) with regards to control laws rather than control sequences  $\underline{U}_k, \underline{U}_{k+1}, \dots, \underline{U}_{k+K-1}$ , the minimization need not be carried out repeatedly at every time step. Instead, it suffices to determine the minimizing control laws  $\underline{\pi}_0^*, \underline{\pi}_1^*, \dots, \underline{\pi}_{K-1}^*$  offline. Online, during operation, the receding horizon principle then demands that the controller evaluates  $\underline{\pi}_0^*$  and transmits the resulting control sequence  $\underline{U}_k = \underline{\pi}_0^*(\mathcal{I}_k)$  to the plant. To obtain  $\underline{\pi}_0^*, \underline{\pi}_1^*, \dots, \underline{\pi}_{K-1}^*$ , we must solve the *stochastic optimal control problem*

$$\begin{aligned} & \min_{\underline{\pi}_{0:K-1}} && \mathcal{J}(\underline{\pi}_{0:K-1}) \\ \text{subject to} &&& \underline{x}_{t+1} = \mathbf{A}\underline{x}_t + \mathbf{B}\underline{u}_t + \underline{w}_t, && t = 0, \dots, K-1, \\ &&& \underline{y}_t = \mathbf{C}\underline{x}_t + \underline{v}_t, && t = 0, \dots, K, \\ &&& \underline{u}_t = \begin{cases} \underline{u}_{t|t-i} & a_t(\underline{U}_t^{\text{bf}}) = i \\ \underline{u}_t^{\text{df}} & a_t(\underline{U}_t^{\text{bf}}) \geq N \end{cases}, && t = 0, \dots, K-1, \\ &&& \underline{U}_t = \underline{\pi}_t(\mathcal{I}_t), && t = 0, \dots, K-1. \end{aligned}\quad (5.3)$$

**Algorithm 5.1** Conceptual Algorithm for Sequence-Based SMPC

- 
- 1: Obtain  $\underline{\pi}_0^*$  by solving the stochastic optimal control problem (5.3) // *Offline*
  - 2: **for**  $k = 0, 1 \dots$  **do**
  - 3:     Create information set  $\mathcal{I}_k$  according to (5.1)
  - 4:     Evaluate first optimal control law,  $\underline{U}_k = \underline{\pi}_0^*(\mathcal{I}_k)$
  - 5:     Transmit  $\underline{U}_k$  to the actuator
  - 6: **end for**
- 

The conceptual algorithm for sequence-based SMPC is summarized in Algorithm 5.1. To bring (5.3) in a more convenient form, we reformulate the cost function (5.2) in terms of the augmented dynamics (4.1) derived in the previous chapter. To that end, we define

$$\begin{aligned}\tilde{\mathbf{Q}}_K &= \begin{bmatrix} \mathbf{Q}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n_\psi \times n_\psi}, \\ \tilde{\mathbf{Q}}_t^{(\theta_t)} &= \begin{bmatrix} \mathbf{Q}_t & \mathbf{0} \\ \mathbf{0} & (\mathbf{H}^{(\theta_t)})^\top \mathbf{R}_t \mathbf{H}^{(\theta_t)} \end{bmatrix} \in \mathbb{R}^{n_\psi \times n_\psi}, \\ \tilde{\mathbf{R}}_t^{(\theta_t)} &= (\mathbf{J}^{(\theta_t)})^\top \mathbf{R}_t \mathbf{J}^{(\theta_t)} \in \mathbb{R}^{Nn_u \times Nn_u},\end{aligned}$$

for  $t = 0, 1, \dots, K-1$ . Observing that (4.6) implies  $\mathbf{H}^{(i)} \neq \mathbf{0} \Rightarrow \mathbf{J}^{(i)} = \mathbf{0}$  and  $\mathbf{J}^{(i)} \neq \mathbf{0} \Rightarrow \mathbf{H}^{(i)} = \mathbf{0}$  for all  $i = 0, 1, \dots, N$ , we may write the cost function as [123, Appendix A.1]

$$\mathcal{J}(\underline{\pi}_{0:K-1}) = \mathbb{E}_{\substack{\underline{\psi}_{0:K}, \\ \theta_{0:K-1}}} \left\{ \underline{\psi}_K^\top \tilde{\mathbf{Q}}_K \underline{\psi}_K + \sum_{t=0}^{K-1} \underline{\psi}_t^\top \tilde{\mathbf{Q}}_t^{(\theta_t)} \underline{\psi}_t + \underline{U}_t^\top \tilde{\mathbf{R}}_t^{(\theta_t)} \underline{U}_t \mid \mathcal{I}_0 \right\}, \quad (5.4)$$

with  $\underline{U}_t = \underline{\pi}_t(\mathcal{I}_t)$ . Although (5.4) and (5.2) are equivalent, we emphasize that the weighting matrix for the control actions in (5.2),  $\mathbf{R}_t$ , is positive definite, whereas its counterpart in (5.4),  $\tilde{\mathbf{R}}_t^{(\theta_t)}$ , is only positive semidefinite because  $\mathbf{J}^{(\theta_t)}$  does not have full column rank. This will pose a technical difficulty in Sections 5.2 to 5.4, as it leads to underdetermined systems of linear equations with infinitely many solutions.

With the reformulated cost function and the augmented dynamics (4.1), the stochastic optimal control problem to solve becomes

$$\begin{aligned}\min_{\underline{\pi}_{0:K-1}} \quad & \mathcal{J}(\underline{\pi}_{0:K-1}) \\ \text{subject to} \quad & \underline{\psi}_{t+1} = \tilde{\mathbf{A}}^{(\theta_t)} \underline{\psi}_t + \tilde{\mathbf{B}}^{(\theta_t)} \underline{U}_t + \tilde{\mathbf{w}}_t, \quad t = 0, \dots, K-1, \\ & \underline{y}_t = [\mathbf{C} \ \mathbf{0}] \underline{\psi}_t + \underline{v}_t, \quad t = 0, \dots, K, \\ & \underline{U}_t = \underline{\pi}_t(\mathcal{I}_t), \quad t = 0, \dots, K-1,\end{aligned} \quad (5.5)$$

with  $\mathcal{J}(\underline{\pi}_{0:K-1})$  given by (5.4). To solve this optimization problem, we can exploit *Bellman's principle of optimality*. Roughly speaking, this principle states that the optimal solution  $\min \mathcal{J}(\underline{\pi}_{0:K-1})$  is obtained by concatenating the solutions of “tail subproblems” of increasing time length [153]. Based on this principle, the *dynamic programming (DP)* technique translates the optimization problem (5.5) into the backward recursion

$$\begin{aligned}\mathcal{V}_K(\mathcal{I}_K) &= \mathbb{E}_{\underline{\psi}_K} \left\{ \underline{\psi}_K^\top \tilde{\mathbf{Q}}_K \underline{\psi}_K \mid \mathcal{I}_K \right\}, \\ \mathcal{V}_t(\mathcal{I}_t) &= \min_{\substack{\underline{U}_t \\ \underline{\psi}_t, \theta_t, \\ \tilde{\mathbf{w}}_t, \\ \mathcal{Y}_{t+1}, \\ \mathcal{A}_{t+1}}} \mathbb{E} \left\{ \underline{\psi}_t^\top \tilde{\mathbf{Q}}_t^{(\theta_t)} \underline{\psi}_t + \underline{U}_t^\top \tilde{\mathbf{R}}_t^{(\theta_t)} \underline{U}_t + \mathcal{V}_{t+1}(\mathcal{I}_{t+1}) \mid \mathcal{I}_t \right\},\end{aligned} \quad (5.6)$$

for  $t = 0, 1, \dots, K - 1$ , with  $\mathcal{V}_t$  the *cost-to-go* at stage  $t$  of the optimization horizon. Thus, starting at the terminal stage of the optimization horizon, the optimal control laws  $\underline{\pi}_0^*, \dots, \underline{\pi}_{K-1}^*$  are computed backward in time and at each stage given by

$$\underline{\pi}_t^*(\mathcal{I}_t) = \arg \min_{\substack{\underline{\psi}_t, \theta_t, \\ \underline{\tilde{w}}_t, \\ \mathcal{Y}_{t+1}, \\ \mathcal{A}_{t+1}}} \mathbb{E} \left\{ \underline{\psi}_t^T \tilde{\mathbf{Q}}_t^{(\theta_t)} \underline{\psi}_t + \underline{U}_t^T \tilde{\mathbf{R}}_t^{(\theta_t)} \underline{U}_t + \mathcal{V}_{t+1}(\mathcal{I}_{t+1}) \mid \mathcal{I}_t \right\}. \quad (5.7)$$

In particular, the optimal cost equals the cost-to-go at the initial stage, i.e., it holds

$$\min \mathcal{J} = \mathcal{J}(\underline{\pi}_{0:K-1}^*) = \mathcal{V}_0(\mathcal{I}_0).$$

Eq. (5.7) implies that the optimal control laws are necessarily *closed-loop feedback* laws because they inherently anticipate the influence of future measurements during the decision-making [158].

## 5.2 Intractability of the Optimal Solution

The DP recursion (5.6) is conceptually simple but cannot be solved analytically. Even worse, the numerical evaluation of the cost-to-go becomes computationally intractable already at stage  $K - 2$ , although  $\mathcal{V}_{K-1}$ , and hence  $\underline{\pi}_{K-1}^*$ , can be expressed in closed-form. To demonstrate this issue, we evaluate the cost-to-go at stage  $K - 1$ , which are given by

$$\mathcal{V}_{K-1} = \min_{\underline{U}_{K-1}} \mathbb{E} \left\{ \underline{\psi}_{K-1}^T \tilde{\mathbf{Q}}_{K-1}^{(\theta_{K-1})} \underline{\psi}_{K-1} + \underline{U}_{K-1}^T \tilde{\mathbf{R}}_{K-1}^{(\theta_{K-1})} \underline{U}_{K-1} + \mathcal{V}_K \mid \mathcal{I}_{K-1} \right\},$$

where we dropped the subscripts below the expectation and the arguments of  $\mathcal{V}_{K-1}$  and  $\mathcal{V}_K$  for notational convenience. Plugging in the expression for  $\mathcal{V}_K$  yields

$$\begin{aligned} \mathcal{V}_{K-1} &= \min_{\underline{U}_{K-1}} \mathbb{E} \left\{ \underline{\psi}_{K-1}^T \tilde{\mathbf{Q}}_{K-1}^{(\theta_{K-1})} \underline{\psi}_{K-1} + \underline{U}_{K-1}^T \tilde{\mathbf{R}}_{K-1}^{(\theta_{K-1})} \underline{U}_{K-1} + \mathbb{E} \left\{ \underline{\psi}_K^T \tilde{\mathbf{Q}}_K \underline{\psi}_K \mid \mathcal{I}_K \right\} \mid \mathcal{I}_{K-1} \right\} \\ &= \min_{\underline{U}_{K-1}} \mathbb{E} \left\{ \underline{\psi}_{K-1}^T \tilde{\mathbf{Q}}_{K-1}^{(\theta_{K-1})} \underline{\psi}_{K-1} + \underline{U}_{K-1}^T \tilde{\mathbf{R}}_{K-1}^{(\theta_{K-1})} \underline{U}_{K-1} + \underline{\psi}_K^T \tilde{\mathbf{Q}}_K \underline{\psi}_K \mid \mathcal{I}_{K-1} \right\}, \end{aligned}$$

where we used the law of total expectation.<sup>2</sup> Using the augmented dynamics (4.1), we obtain

$$\begin{aligned} \mathcal{V}_{K-1} &= \mathbb{E} \left\{ \underline{\psi}_{K-1}^T \left( \tilde{\mathbf{Q}}_{K-1}^{(\theta_{K-1})} + \left( \tilde{\mathbf{A}}^{(\theta_{K-1})} \right)^T \tilde{\mathbf{Q}}_K \tilde{\mathbf{A}}^{(\theta_{K-1})} \right) \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1} \right\} \\ &\quad + \mathbb{E} \left\{ \tilde{\underline{w}}_{K-1}^T \tilde{\mathbf{Q}}_K \tilde{\underline{w}}_{K-1} \mid \mathcal{I}_{K-1} \right\} \\ &\quad + \min_{\underline{U}_{K-1}} \left[ \underline{U}_{K-1}^T \mathbb{E} \left\{ \tilde{\mathbf{R}}_{K-1}^{(\theta_{K-1})} + \left( \tilde{\mathbf{B}}^{(\theta_{K-1})} \right)^T \tilde{\mathbf{Q}}_K \tilde{\mathbf{B}}^{(\theta_{K-1})} \mid \mathcal{I}_{K-1} \right\} \underline{U}_{K-1} \right. \\ &\quad \left. + 2 \mathbb{E} \left\{ \underline{\psi}_{K-1}^T \left( \tilde{\mathbf{A}}^{(\theta_{K-1})} \right)^T \tilde{\mathbf{Q}}_K \tilde{\mathbf{B}}^{(\theta_{K-1})} \mid \mathcal{I}_{K-1} \right\} \underline{U}_{K-1} \right], \end{aligned} \quad (5.8)$$

since  $\mathbb{E} \{ \tilde{\underline{w}}_{K-1} \} = \mathbf{0}$ , and  $\tilde{\underline{w}}_{K-1}$  and  $\underline{\psi}_{K-1}$  are independent. The term in square brackets is bounded, quadratic, and convex with regards to  $\underline{U}_{K-1}$ . Setting its derivative to zero yields the necessary and sufficient minimality condition

$$\begin{aligned} &\mathbb{E} \left\{ \tilde{\mathbf{R}}_{K-1}^{(\theta_{K-1})} + \left( \tilde{\mathbf{B}}^{(\theta_{K-1})} \right)^T \tilde{\mathbf{Q}}_K \tilde{\mathbf{B}}^{(\theta_{K-1})} \mid \mathcal{I}_{K-1} \right\} \underline{U}_{K-1} \\ &\stackrel{!}{=} - \mathbb{E} \left\{ \left( \tilde{\mathbf{B}}^{(\theta_{K-1})} \right)^T \tilde{\mathbf{Q}}_K \tilde{\mathbf{A}}^{(\theta_{K-1})} \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1} \right\}, \end{aligned} \quad (5.9)$$

<sup>2</sup>Applied to our case, the law of total expectation implies that  $\mathbb{E} \left\{ \mathbb{E} \left\{ g(\underline{\psi}_{t+1}) \mid \mathcal{I}_{t+1} \right\} \mid \mathcal{I}_t \right\} = \mathbb{E} \left\{ g(\underline{\psi}_{t+1}) \mid \mathcal{I}_t \right\}$  for any measurable function  $g$  because  $\mathcal{I}_t \subseteq \mathcal{I}_{t+1}$  [173, Lemma 1c].

which is a system of linear equations. The matrix on the left side of (5.9) is only positive and thus rank deficient, which implies that the system of equations has infinitely many solutions. The reason is that the minimization problem is somewhat ill-posed because  $\underline{U}_{K-1}$  contains entries to be applied after the end of the optimization horizon, namely the entries  $\underline{u}_{K|K-1}, \underline{u}_{K+1|K-1}, \dots, \underline{u}_{K+N-2|K-1}$ . These do not contribute to the cost-to-go and, hence, can be chosen arbitrarily. One particular solution of (5.9) is the *minimum norm solution* given by

$$\underline{U}_{K-1}^+ = -\mathbf{L}_{K-1}^\dagger \mathbb{E} \left\{ \left( \tilde{\mathbf{B}}^{(\theta_{K-1})} \right)^\top \tilde{\mathbf{Q}}_K \tilde{\mathbf{A}}^{(\theta_{K-1})} \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1} \right\}, \quad (5.10)$$

where  $\mathbf{L}_{K-1}$  denotes the matrix on the left side of (5.9), i.e.,

$$\mathbf{L}_{K-1} = \mathbb{E} \left\{ \tilde{\mathbf{R}}_{K-1}^{(\theta_{K-1})} + \left( \tilde{\mathbf{B}}^{(\theta_{K-1})} \right)^\top \tilde{\mathbf{Q}}_K \tilde{\mathbf{B}}^{(\theta_{K-1})} \mid \mathcal{I}_{K-1} \right\}, \quad (5.11)$$

and with  $\mathbf{L}_{K-1}^\dagger$  its *Moore-Penrose pseudoinverse* (cf. Appendix A.5).

Plugging (5.10) into (5.8) then leads to

$$\begin{aligned} \mathcal{V}_{K-1} &= \mathbb{E} \left\{ \underline{\psi}_{K-1}^\top \left( \tilde{\mathbf{Q}}_{K-1}^{(\theta_{K-1})} + \left( \tilde{\mathbf{A}}^{(\theta_{K-1})} \right)^\top \tilde{\mathbf{Q}}_K \tilde{\mathbf{A}}^{(\theta_{K-1})} \right) \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1} \right\} \\ &\quad - \left( \mathbb{E} \left\{ \left( \tilde{\mathbf{B}}^{(\theta_{K-1})} \right)^\top \tilde{\mathbf{Q}}_K \tilde{\mathbf{A}}^{(\theta_{K-1})} \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1} \right\} \right)^\top \\ &\quad \cdot \mathbf{L}_{K-1}^\dagger \mathbb{E} \left\{ \left( \tilde{\mathbf{B}}^{(\theta_{K-1})} \right)^\top \tilde{\mathbf{Q}}_K \tilde{\mathbf{A}}^{(\theta_{K-1})} \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1} \right\} \\ &\quad + \mathbb{E} \left\{ \tilde{\underline{w}}_{K-1}^\top \tilde{\mathbf{Q}}_K \tilde{\underline{w}}_{K-1} \mid \mathcal{I}_{K-1} \right\}, \end{aligned}$$

for the cost-to-go at stage  $K - 1$ . Using the law of total probability,  $\mathcal{V}_{K-1}$  can be expressed as

$$\begin{aligned} \mathcal{V}_{K-1} &= \sum_{i=0}^N \mu_{K-1}^{(i)} \mathbb{E} \left\{ \underline{\psi}_{K-1}^\top \left( \tilde{\mathbf{Q}}_{K-1}^{(i)} + \left( \tilde{\mathbf{A}}^{(i)} \right)^\top \tilde{\mathbf{Q}}_K \tilde{\mathbf{A}}^{(i)} \right) \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\} \\ &\quad - \left( \sum_{i=0}^N \mu_{K-1}^{(i)} \left( \tilde{\mathbf{B}}^{(i)} \right)^\top \tilde{\mathbf{Q}}_K \tilde{\mathbf{A}}^{(i)} \mathbb{E} \left\{ \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\} \right)^\top \\ &\quad \cdot \mathbf{L}_{K-1}^\dagger \left( \sum_{i=0}^N \mu_{K-1}^{(i)} \left( \tilde{\mathbf{B}}^{(i)} \right)^\top \tilde{\mathbf{Q}}_K \tilde{\mathbf{A}}^{(i)} \mathbb{E} \left\{ \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\} \right) \\ &\quad + \mathbb{E} \left\{ \tilde{\underline{w}}_{K-1}^\top \tilde{\mathbf{Q}}_K \tilde{\underline{w}}_{K-1} \mid \mathcal{I}_{K-1} \right\}, \end{aligned}$$

where  $\mu_{K-1}^{(i)} = \mathbb{P}[\theta_{K-1} = i \mid \mathcal{I}_{K-1}]$  is the mode probability. With the definitions

$$\tilde{\underline{e}}_{K-1}^{(i)} = \underline{\psi}_{K-1} - \mathbb{E} \left\{ \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\}, \quad (5.12)$$

$$\mathbf{D}_{K-1}^{(i)} = \tilde{\mathbf{Q}}_{K-1}^{(i)} + \left( \tilde{\mathbf{A}}^{(i)} \right)^\top \tilde{\mathbf{Q}}_K \tilde{\mathbf{A}}^{(i)} - \mathbf{M}_{K-1}^{(i)}, \quad (5.13)$$

$$\mathbf{M}_{K-1}^{(i)} = \left( \tilde{\mathbf{B}}^{(i)} \right)^\top \tilde{\mathbf{Q}}_K \tilde{\mathbf{B}}^{(i)} \mathbf{L}_{K-1}^\dagger \left( \tilde{\mathbf{B}}^{(i)} \right)^\top \tilde{\mathbf{Q}}_K \tilde{\mathbf{A}}^{(i)}, \quad (5.14)$$

for  $i = 0, 1, \dots, N$ , we may write this more compactly as

$$\begin{aligned} \mathcal{V}_{K-1} &= \sum_{i=0}^N \mu_{K-1}^{(i)} \mathbb{E} \left\{ \underline{\psi}_{K-1}^T \mathbf{D}_{K-1}^{(i)} \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\} \\ &+ \sum_{i=0}^N \mu_{K-1}^{(i)} \mathbb{E} \left\{ \left( \tilde{\underline{\epsilon}}_{K-1}^{(i)} \right)^T \mathbf{M}_{K-1}^{(i)} \tilde{\underline{\epsilon}}_{K-1}^{(i)} \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\} \\ &+ \mathbb{E} \left\{ \tilde{\underline{w}}_{K-1}^T \tilde{\mathbf{Q}}_K \tilde{\underline{w}}_{K-1} \mid \mathcal{I}_{K-1} \right\}. \end{aligned} \quad (5.15)$$

We see from (5.15) that the cost-to-go at stage  $K-1$  are the sum of  $N+1$  functions, one for each mode of the augmented dynamics. These functions are quadratic with regards to  $\underline{\psi}_{K-1}$  and each contributes to the overall cost-to-go according to the mode probability  $\mu_{K-1}^{(i)}$ . The conditional expectation on the right of (5.12) is the mode-conditioned estimate of the augmented state  $\underline{\psi}_{K-1}$ . Consequently,  $\tilde{\underline{\epsilon}}_{K-1}^{(i)}$  is the mode-conditioned estimation error and the second sum in (5.15) quantifies the contribution of the estimation error to the cost-to-go.

Similarly, we can rewrite (5.10) by evaluating the expectation with regards to  $\theta_{K-1}$  to reveal that the optimal control law  $\underline{\pi}_{K-1}^*$  is the weighted combination of  $N+1$  mode-conditioned control laws  $\underline{\pi}_{K-1}^{*(i)}$  according to

$$\underline{\pi}_{K-1}^* (\mathcal{I}_{K-1}) = -\mathbf{L}_{K-1}^\dagger \sum_{i=0}^N \mu_{K-1}^{(i)} \underline{\pi}_{K-1}^{*(i)} (\mathcal{I}_{K-1}), \quad (5.16)$$

where  $\underline{\pi}_{K-1}^{*(i)}$  is linear in the mode-conditioned state estimate and given by

$$\underline{\pi}_{K-1}^{*(i)} (\mathcal{I}_{K-1}) = \left( \tilde{\mathbf{B}}^{(i)} \right)^T \tilde{\mathbf{Q}}_K \tilde{\mathbf{A}}^{(i)} \mathbb{E} \left\{ \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\}.$$

Having obtained  $\mathcal{V}_{K-1}$ , the cost-to-go at stage  $K-2$  are given by

$$\begin{aligned} \mathcal{V}_{K-2} &= \min_{\underline{U}_{K-2}} \mathbb{E} \left\{ \underline{\psi}_{K-2}^T \tilde{\mathbf{Q}}_{K-2}^{(\theta_{K-2})} \underline{\psi}_{K-2} + \underline{U}_{K-2}^T \tilde{\mathbf{R}}_{K-2}^{(\theta_{K-2})} \underline{U}_{K-2} + \mathcal{V}_{K-1} \mid \mathcal{I}_{K-2} \right\} \\ &= \mathbb{E} \left\{ \underline{\psi}_{K-2}^T \tilde{\mathbf{Q}}_{K-2}^{(\theta_{K-2})} \underline{\psi}_{K-2} \mid \mathcal{I}_{K-2} \right\} + \mathbb{E} \left\{ \tilde{\underline{w}}_{K-1}^T \tilde{\mathbf{Q}}_K \tilde{\underline{w}}_{K-1} \mid \mathcal{I}_{K-2} \right\} \\ &+ \min_{\underline{U}_{K-2}} \left[ \mathbb{E} \left\{ \underline{U}_{K-2}^T \tilde{\mathbf{R}}_{K-2}^{(\theta_{K-2})} \underline{U}_{K-2} \mid \mathcal{I}_{K-2} \right\} \right. \\ &\left. + \sum_{i=0}^N \mathbb{P}[\theta_{K-1} = i \mid \mathcal{I}_{K-2}] \mathbb{E} \left\{ \underline{\psi}_{K-1}^T \mathbf{D}_{K-1}^{(i)} \underline{\psi}_{K-1} + \left( \tilde{\underline{\epsilon}}_{K-1}^{(i)} \right)^T \mathbf{M}_{K-1}^{(i)} \tilde{\underline{\epsilon}}_{K-1}^{(i)} \mid \mathcal{I}_{K-2} \right\} \right]. \end{aligned} \quad (5.17)$$

However, we cannot further simplify the right side. In particular, the term penalizing the estimation error cannot be excluded from the minimization because it depends on  $\underline{U}_{K-2}$  as the following lemma shows.

**Lemma 5.1:**

For every time step  $k$ , the estimation error  $\tilde{\underline{\epsilon}}_k = \underline{\psi}_k - \mathbb{E} \left\{ \underline{\psi}_k \mid \mathcal{I}_k \right\}$  is a function of the computed control sequences  $\underline{U}_0, \dots, \underline{U}_{k-1}$ .

*Proof.* The proof is given in Appendix C.1. □

This observation is a manifestation of the dual effect that we already mentioned in Section 4.4: The control sequence  $\underline{U}_{K-2}$  affects the plant state (namely, the desired control action) and also the future

estimation error  $\tilde{\underline{e}}_{K-1}$ . At stage  $K - 1$ , the controller achieves no cost savings by reducing the estimation error since no subsequent control inputs will be computed. At all other stages, however, it might be more beneficial to generate control sequences that excite the plant such that the estimation error is reduced, as this yields lower cost-to-go at later stages.<sup>3</sup> The dual effect is also reflected by the matrix  $\mathbf{L}_{K-1}^\dagger$  in the expressions for  $\mathbf{D}_{K-1}^{(i)}$  and  $\mathbf{M}_{K-1}^{(i)}$ , (5.13) and (5.14).  $\mathbf{L}_{K-1}$ , given by (5.11), is dependent on all modes so that  $\mathbf{L}_{K-1}^\dagger$  nonlinearly couples  $\mathbf{D}_{K-1}^{(i)}$  and  $\mathbf{D}_{K-1}^{(j)}$ , and  $\mathbf{M}_{K-1}^{(i)}$  and  $\mathbf{M}_{K-1}^{(j)}$ , respectively. This coupling results from the fact that the controller's knowledge of the structure of the augmented dynamics (i.e., its mode) is not perfect but limited to the probabilities  $\mu_{K-1}^{(i)}$ . An unpleasant consequence of Lemma 5.1 is that we cannot further simplify the expectations  $\mathbb{E}\left\{\underline{\psi}_{K-1}^\top \mathbf{D}_{K-1}^{(i)} \underline{\psi}_{K-1} + \left(\tilde{\underline{e}}_{K-1}^{(i)}\right)^\top \mathbf{M}_{K-1}^{(i)} \tilde{\underline{e}}_{K-1}^{(i)} \mid \mathcal{I}_{K-2}\right\}$  in the sum in (5.17), because evaluating them demands the computation of products that contain  $\mathbf{L}_{K-1}^\dagger$ , i.e., the pseudoinverse of a conditional expectation. This leads to non-convex minimization problems so that the DP recursion (5.6) becomes intractable and prevents a numerical determination of the optimal control laws. This is even true for the state feedback case [175, Chapter 3]. Approximate algorithms for the optimal control of MJLS with imperfect mode observation have thus been actively researched over the years. Yet, focus has been laid on state feedback [148, 176–179] and static output feedback, where noise-free measurements of the plant state are available [149, 180–182].

To conclude this section, we note that any practical implementation of the optimal control law (5.10), or equivalently (5.16), is necessarily suboptimal. Here, the reason is that the computation of the *minimum mean square error (MMSE) estimate*

$$\hat{\underline{\psi}}_k = \mathbb{E}\left\{\underline{\psi}_k \mid \mathcal{I}_k\right\} = \sum_{i=0}^N \mu_k^{(i)} \mathbb{E}\left\{\underline{\psi}_k \mid \mathcal{I}_k, \theta_k = i\right\},$$

is intractable [183, 184]. As the modes  $\theta_{0:k-1}$  of the MJLS (4.1) – or, in other words, the actual plant inputs  $\underline{u}_{0:k-1}$  in (3.1) – are not perfectly known, the number of hypotheses increases exponentially in time, and so does the computational and memory complexity of the MMSE estimator [125]. Consequently, a variety of approximations have been proposed. The proposed approaches range from linear MMSE estimators [185–187] to estimators that keep only a fixed number hypotheses by applying some merging strategy [145, 188]. Among the latter, *multiple model* estimators have gained much attraction. Here, the general idea is to run a bank of Kalman filters, one for each mode of the MJLS, in combination with a method to estimate the mode probabilities  $\mu_k^{(i)}$ . For the augmented dynamics (4.1), a multiple model estimator thus consists of  $N + 1$  Kalman filters. The state estimate is then maintained in the form of a mixture distribution with  $N + 1$  components. Each component is a Gaussian with mean and covariance

$$\hat{\underline{\psi}}_k^{(i)} = \mathbb{E}\left\{\underline{\psi}_k \mid \mathcal{I}_k, \theta_k = i\right\}, \quad \Sigma_k^{(i)} = \mathbb{E}\left\{\tilde{\underline{e}}_k^{(i)} \left(\tilde{\underline{e}}_k^{(i)}\right)^\top \mid \mathcal{I}_k, \theta_k = i\right\}, \quad (5.18)$$

for  $i = 0, 1, \dots, N$ , which is provided by one of the mode-conditioned Kalman filters, and weighted according to the estimated mode probability  $\mu_k^{(i)}$ . Eq. (5.18) exhibits a connection between multiple model estimation and the stochastic optimal control problem (5.3), because both  $\hat{\underline{\psi}}_{K-1}^{(i)}$  and  $\Sigma_{K-1}^{(i)}$  naturally appear in the cost-to-go (5.15) and (5.17).<sup>4</sup>

<sup>3</sup>Using control inputs for uncertainty reduction is often called *probing* in the literature [159, 174].

<sup>4</sup>The covariance of the estimation error  $\Sigma_{K-1}^{(i)}$  appears if we rewrite  $\mathbb{E}\left\{\left(\tilde{\underline{e}}_{K-1}^{(i)}\right)^\top \mathbf{M}_{K-1}^{(i)} \tilde{\underline{e}}_{K-1}^{(i)}\right\}$  as  $\text{tr}[\mathbf{M}_{K-1}^{(i)} \Sigma_{K-1}^{(i)}]$ .

## 5.3 Multiple Model Based SMPC

The previous section showed that any tractable solution of the stochastic optimal control problem (5.3) is necessarily approximate. In this section, we present a computationally cheap algorithm for sequence-based SMPC that exploits the Gaussian mixture representation of the state estimate provided by a multiple model estimator. Using ideas akin to those in [189], this enables us to write the cost-to-go  $\mathcal{V}_t$  at each stage  $t$  of the DP recursion as a weighted combination of  $N + 1$  quadratic functions. These functions are linearly coupled by matrices  $\mathbf{P}_t^{(i)}$  that are propagated backward according to mode-conditioned Riccati-like equations. However, relying on the particular representation of the state estimate alone is not enough as it does not eliminate the dual effect. Thus, to obtain the desired representation of the cost-to-go, we must assume that the estimation error is independent of previous control actions. This assumption allows us to exclude the portion of the cost-to-go related to the estimation error from the minimization and removes the nonlinear coupling introduced by the matrix  $\mathbf{L}^\dagger$  (cf. (5.11)). At each stage, the resulting control law  $\pi_t^*$  is then a weighted combination of  $N + 1$  linear mode-dependent control laws, i.e., of the form (5.16).

In the following, we first derive the control laws by carrying out the DP recursion in Section 5.3.1. Then, in Section 5.3.2, we present a conceptual algorithm for the associated multiple model estimation. Finally, the proposed algorithm for sequence-based SMPC is summarized in Section 5.3.3.

### 5.3.1 Derivation of the Control Laws

We start with recalling the DP recursion (5.6)

$$\begin{aligned}\mathcal{V}_K &= \mathbb{E}\left\{\underline{\psi}_K^\top \tilde{\mathbf{Q}}_K \underline{\psi}_K \mid \mathcal{I}_K\right\}, \\ \mathcal{V}_t &= \min_{\underline{U}_t} \mathbb{E}\left\{\underline{\psi}_t^\top \tilde{\mathbf{Q}}_t^{(\theta_t)} \underline{\psi}_t + \underline{U}_t^\top \tilde{\mathbf{R}}_t^{(\theta_t)} \underline{U}_t + \mathcal{V}_{t+1} \mid \mathcal{I}_t\right\},\end{aligned}$$

for  $t = 0, 1, \dots, K-1$ , where we again omitted the subscripts below the expectation and the arguments of the cost-to-go for notational convenience. Our goal is to express the cost-to-go  $\mathcal{V}_t$  at each stage  $t$  as a weighted sum of quadratic functions according to

$$\mathcal{V}_t \stackrel{!}{=} \sum_{i=0}^N \mu_t^{(i)} \left[ \mathbb{E}\left\{\underline{\psi}_t^\top \hat{\mathbf{P}}_t^{(i)} \underline{\psi}_t + \left(\tilde{\underline{e}}_t^{(i)}\right)^\top \hat{\mathbf{S}}_t^{(i)} \tilde{\underline{e}}_t^{(i)} \mid \mathcal{I}_t, \theta_t = i\right\} + \alpha_t^{(i)} \right], \quad (5.19)$$

with  $\hat{\mathbf{P}}_t^{(i)}, \hat{\mathbf{S}}_t^{(i)} \succcurlyeq 0$ ,  $\alpha_t^{(i)} \geq 0$ ,  $\tilde{\underline{e}}_t^{(i)}$  the mode-conditioned estimation error as per (5.12), and where  $\mu_t^{(i)} = \mathbb{P}[\theta_t = i \mid \mathcal{I}_t]$  is the mode probability. Note that (5.19) trivially holds at the terminal stage  $K$  with  $\hat{\mathbf{P}}_K^{(i)} = \tilde{\mathbf{Q}}_K$ ,  $\hat{\mathbf{S}}_K^{(i)} = \mathbf{0}$ , and  $\alpha_K^{(i)} = 0$  for  $i = 0, 1, \dots, N$ . Similarly, we observed in the previous section that this representation also holds for the cost-to-go  $\mathcal{V}_{K-1}$  at stage  $K-1$  (cf. (5.15)). For all other stages, however, the representation (5.19) is only an approximation.

To obtain this approximation, we first define the *mode-conditioned cost-to-go*  $\mathcal{V}_t^{(i)}$  by means of the recursion

$$\begin{aligned}\mathcal{V}_K^{(i)} &= \mathbb{E}\left\{\underline{\psi}_K^\top \tilde{\mathbf{Q}}_K \underline{\psi}_K \mid \mathcal{I}_K, \theta_K = i\right\}, \\ \mathcal{V}_t^{(i)} &= \min_{\underline{U}_t} \mathbb{E}\left\{\underline{\psi}_t^\top \tilde{\mathbf{Q}}_t^{(i)} \underline{\psi}_t + \underline{U}_t^\top \tilde{\mathbf{R}}_t^{(i)} \underline{U}_t + \mathcal{V}_{t+1} \mid \mathcal{I}_t, \theta_t = i\right\},\end{aligned} \quad (5.20)$$

for  $t = 0, 1, \dots, K-1$  and  $i = 0, 1, \dots, N$ .

Now, for the time being, let us assume that for  $t = 0, 1, \dots, K$ , we can express  $\mathcal{V}_t^{(i)}$  as the quadratic function

$$\mathcal{V}_t^{(i)} \stackrel{!}{=} \mathbb{E} \left\{ \underline{\psi}_t^T \mathbf{P}_t^{(i)} \underline{\psi}_t + \left( \tilde{\underline{e}}_t^{(i)} \right)^T \mathbf{S}_t^{(i)} \tilde{\underline{e}}_t^{(i)} \mid \mathcal{I}_t, \theta_t = i \right\} + \alpha_t^{(i)}, \quad (5.21)$$

with  $\mathbf{P}_t^{(i)}, \mathbf{S}_t^{(i)} \succcurlyeq 0$ . Note that the mode-conditioned cost-to-go (5.21) are similar but not equal to functions on the right of (5.19). As we shall see soon, there is a slight but important difference between the matrices  $\widehat{\mathbf{S}}_t^{(i)}$  and  $\mathbf{S}_t^{(i)}$  and, likewise, between  $\widehat{\mathbf{P}}_t^{(i)}$  and  $\mathbf{P}_t^{(i)}$ . Note also that the representation (5.21) holds true at the terminal stage  $K$  when we set  $\mathbf{P}_K^{(i)} = \tilde{\mathbf{Q}}_K, \mathbf{S}_K^{(i)} = \mathbf{0}$ , and  $\alpha_K^{(i)} = 0$  for  $i = 0, 1, \dots, N$ . For all other stages, however, (5.21) is only an approximation.

Then, we use the law of total probability to write the recurring expression  $\mathbb{E}\{\mathcal{V}_{t+1} \mid \mathcal{I}_t\}$  in the DP recursion (5.6) as

$$\mathbb{E}\{\mathcal{V}_{t+1} \mid \mathcal{I}_t\} = \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E}\{\mathcal{V}_{t+1} \mid \mathcal{I}_t, \theta_{t+1} = r\},$$

for  $t = 0, 1, \dots, K-1$ . Similar to [189], we then assume that we can approximate the sum on the right by replacing  $\mathcal{V}_{t+1}$  inside the expectation by the mode-conditioned cost-to-go  $\mathcal{V}_{t+1}^{(r)}$  according to

$$\sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E}\{\mathcal{V}_{t+1} \mid \mathcal{I}_t, \theta_{t+1} = r\} \stackrel{!}{\approx} \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E}\{\mathcal{V}_{t+1}^{(r)} \mid \mathcal{I}_t, \theta_{t+1} = r\}.$$

With this assumption, we obtain, using the representation (5.21),

$$\begin{aligned} \mathbb{E}\{\mathcal{V}_{t+1} \mid \mathcal{I}_t\} &= \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E}\{\mathcal{V}_{t+1} \mid \mathcal{I}_t, \theta_{t+1} = r\} \\ &\approx \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E}\{\mathcal{V}_{t+1}^{(r)} \mid \mathcal{I}_t, \theta_{t+1} = r\} \\ &= \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E}\left\{ \mathbb{E}\left\{ \underline{\psi}_{t+1}^T \mathbf{P}_{t+1}^{(r)} \underline{\psi}_{t+1} \mid \mathcal{I}_{t+1}, \theta_{t+1} = r \right\} \mid \mathcal{I}_t, \theta_{t+1} = r \right\} \\ &\quad + \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E}\left\{ \mathbb{E}\left\{ \left( \tilde{\underline{e}}_{t+1}^{(r)} \right)^T \mathbf{S}_{t+1}^{(r)} \tilde{\underline{e}}_{t+1}^{(r)} \mid \mathcal{I}_{t+1}, \theta_{t+1} = r \right\} \mid \mathcal{I}_t, \theta_{t+1} = r \right\} \\ &\quad + \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E}\left\{ \mathbb{E}\left\{ \alpha_{t+1}^{(r)} \mid \mathcal{I}_{t+1}, \theta_{t+1} = r \right\} \mid \mathcal{I}_t, \theta_{t+1} = r \right\}, \end{aligned}$$

which, by the law of total expectation, yields

$$\begin{aligned} \mathbb{E}\{\mathcal{V}_{t+1} \mid \mathcal{I}_t\} &\approx \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \\ &\quad \cdot \mathbb{E}\left\{ \underline{\psi}_{t+1}^T \mathbf{P}_{t+1}^{(r)} \underline{\psi}_{t+1} + \left( \tilde{\underline{e}}_{t+1}^{(r)} \right)^T \mathbf{S}_{t+1}^{(r)} \tilde{\underline{e}}_{t+1}^{(r)} + \alpha_{t+1}^{(r)} \mid \mathcal{I}_t, \theta_{t+1} = r \right\}. \end{aligned}$$

Plugging this approximation back into the expression for the cost-to-go  $\mathcal{V}_t$  gives

$$\begin{aligned} \mathcal{V}_t &= \min_{\underline{U}_t} \mathbb{E} \left\{ \underline{\psi}_t^T \tilde{\mathbf{Q}}_t^{(\theta_t)} \underline{\psi}_t + \underline{U}_t^T \tilde{\mathbf{R}}_t^{(\theta_t)} \underline{U}_t + \mathcal{V}_{t+1} \mid \mathcal{I}_t \right\} \\ &\approx \mathbb{E} \left\{ \underline{\psi}_t^T \tilde{\mathbf{Q}}_t^{(\theta_t)} \underline{\psi}_t \mid \mathcal{I}_t \right\} + \min_{\underline{U}_t} \left[ \mathbb{E} \left\{ \underline{U}_t^T \tilde{\mathbf{R}}_t^{(\theta_t)} \underline{U}_t \mid \mathcal{I}_t \right\} \right. \\ &\quad \left. + \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E} \left\{ \underline{\psi}_{t+1}^T \mathbf{P}_{t+1}^{(r)} \underline{\psi}_{t+1} + \left( \tilde{\underline{e}}_{t+1}^{(r)} \right)^T \mathbf{S}_{t+1}^{(r)} \tilde{\underline{e}}_{t+1}^{(r)} + \alpha_{t+1}^{(r)} \mid \mathcal{I}_t, \theta_{t+1} = r \right\} \right], \end{aligned}$$

which, under the assumption that  $\tilde{\underline{e}}_{t+1}^{(r)}$ ,  $\mathbf{S}_{t+1}^{(r)}$ , and  $\alpha_{t+1}^{(r)}$  be independent of  $\underline{U}_t$ , becomes

$$\begin{aligned} \mathcal{V}_t &\approx \mathbb{E} \left\{ \underline{\psi}_t^T \tilde{\mathbf{Q}}_t^{(\theta_t)} \underline{\psi}_t \mid \mathcal{I}_t \right\} + \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E} \left\{ \left( \tilde{\underline{e}}_{t+1}^{(r)} \right)^T \mathbf{S}_{t+1}^{(r)} \tilde{\underline{e}}_{t+1}^{(r)} + \alpha_{t+1}^{(r)} \mid \mathcal{I}_t, \theta_{t+1} = r \right\} \\ &\quad + \min_{\underline{U}_t} \left[ \underline{U}_t^T \mathbb{E} \left\{ \tilde{\mathbf{R}}_t^{(\theta_t)} \mid \mathcal{I}_t \right\} \underline{U}_t + \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E} \left\{ \underline{\psi}_{t+1}^T \mathbf{P}_{t+1}^{(r)} \underline{\psi}_{t+1} \mid \mathcal{I}_t, \theta_{t+1} = r \right\} \right]. \end{aligned}$$

Note that the assumption that the portion of the cost related to the estimation error  $\tilde{\underline{e}}_{t+1}^{(r)}$  be independent of  $\underline{U}_t$  contradicts Lemma 5.1, implying that the impact of the dual effect is neglected. Using the augmented dynamics (4.1), we obtain

$$\begin{aligned} \mathcal{V}_t &\approx \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E} \left\{ \underline{\psi}_t^T \left( \tilde{\mathbf{Q}}_t^{(\theta_t)} + \left( \tilde{\mathbf{A}}^{(\theta_t)} \right)^T \mathbf{P}_{t+1}^{(r)} \tilde{\mathbf{A}}^{(\theta_t)} \right) \underline{\psi}_t \mid \mathcal{I}_t, \theta_{t+1} = r \right\} \\ &\quad + \min_{\underline{U}_t} \left[ \underline{U}_t^T \left( \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E} \left\{ \tilde{\mathbf{R}}_t^{(\theta_t)} + \left( \tilde{\mathbf{B}}^{(\theta_t)} \right)^T \mathbf{P}_{t+1}^{(r)} \tilde{\mathbf{B}}^{(\theta_t)} \mid \mathcal{I}_t, \theta_{t+1} = r \right\} \right) \underline{U}_t \right. \\ &\quad \left. + 2 \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E} \left\{ \underline{\psi}_t^T \left( \tilde{\mathbf{A}}^{(\theta_t)} \right)^T \mathbf{P}_{t+1}^{(r)} \tilde{\mathbf{B}}^{(\theta_t)} \mid \mathcal{I}_t, \theta_{t+1} = r \right\} \underline{U}_t \right] + \beta_t, \end{aligned}$$

where

$$\beta_t = \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E} \left\{ \tilde{\underline{w}}_t^T \mathbf{P}_{t+1}^{(r)} \tilde{\underline{w}}_t + \left( \tilde{\underline{e}}_{t+1}^{(r)} \right)^T \mathbf{S}_{t+1}^{(r)} \tilde{\underline{e}}_{t+1}^{(r)} + \alpha_{t+1}^{(r)} \mid \mathcal{I}_t, \theta_{t+1} = r \right\}.$$

The expression to minimize is quadratic and convex with regards to  $\underline{U}_t$ . Hence, the necessary and sufficient condition for a minimizer is

$$\begin{aligned} &\left( \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E} \left\{ \tilde{\mathbf{R}}_t^{(\theta_t)} + \left( \tilde{\mathbf{B}}^{(\theta_t)} \right)^T \mathbf{P}_{t+1}^{(r)} \tilde{\mathbf{B}}^{(\theta_t)} \mid \mathcal{I}_t, \theta_{t+1} = r \right\} \right) \underline{U}_t \\ &\stackrel{!}{=} - \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E} \left\{ \left( \tilde{\mathbf{B}}^{(\theta_t)} \right)^T \mathbf{P}_{t+1}^{(r)} \tilde{\mathbf{A}}^{(\theta_t)} \underline{\psi}_t \mid \mathcal{I}_t, \theta_{t+1} = r \right\}. \end{aligned} \tag{5.22}$$

As in (5.9), the matrix on the left side of (5.22) is rank deficient, implying the existence of infinitely many solutions. We again pick the minimum norm solution

$$\begin{aligned} \underline{U}_t^+ &= - \left( \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E} \left\{ \tilde{\mathbf{R}}_t^{(\theta_t)} + \left( \tilde{\mathbf{B}}^{(\theta_t)} \right)^T \mathbf{P}_{t+1}^{(r)} \tilde{\mathbf{B}}^{(\theta_t)} \mid \mathcal{I}_t, \theta_{t+1} = r \right\} \right)^\dagger \\ &\quad \cdot \left( \sum_{r=0}^N \mathbb{P}[\theta_{t+1} = r \mid \mathcal{I}_t] \mathbb{E} \left\{ \left( \tilde{\mathbf{B}}^{(\theta_t)} \right)^T \mathbf{P}_{t+1}^{(r)} \tilde{\mathbf{A}}^{(\theta_t)} \underline{\psi}_t \mid \mathcal{I}_t, \theta_{t+1} = r \right\} \right), \end{aligned}$$

which, after evaluating the expectations with regards to  $\theta_t$  becomes

$$\begin{aligned} \underline{U}_t^+ = & - \left( \sum_{i=0}^N \sum_{r=0}^N t_{ir} \mu_t^{(i)} \left[ \tilde{\mathbf{R}}_t^{(i)} + \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{P}_{t+1}^{(r)} \tilde{\mathbf{B}}^{(i)} \right] \right)^\dagger \\ & \cdot \left( \sum_{i=0}^N \sum_{r=0}^N t_{ir} \mu_t^{(i)} \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{P}_{t+1}^{(r)} \tilde{\mathbf{A}}^{(i)} \hat{\underline{\psi}}_t^{(i)} \right). \end{aligned}$$

Since  $\tilde{\mathbf{R}}_t^{(\theta_t)}$  is only nonzero for  $\theta_t = 0$  due to (4.6), we may write this as

$$\underline{U}_t^+ = - \left( \mu_t^{(0)} \tilde{\mathbf{R}}_t^{(0)} + \left[ \sum_{i=0}^N \mu_t^{(i)} \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_t^{(i)} \tilde{\mathbf{B}}^{(i)} \right] \right)^\dagger \left[ \sum_{i=0}^N \mu_t^{(i)} \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_t^{(i)} \tilde{\mathbf{A}}^{(i)} \hat{\underline{\psi}}_t^{(i)} \right], \quad (5.23)$$

with

$$\mathbf{Y}_t^{(i)} = \sum_{r=0}^N t_{ir} \mathbf{P}_{t+1}^{(r)}. \quad (5.24)$$

Using this result, we define

$$\begin{aligned} \hat{\mathbf{P}}_t^{(i)} &= \tilde{\mathbf{Q}}_t^{(i)} + \left( \tilde{\mathbf{A}}^{(i)} \right)^T \mathbf{Y}_t^{(i)} \tilde{\mathbf{A}}^{(i)} - \hat{\mathbf{S}}_t^{(i)}, \\ \hat{\mathbf{S}}_t^{(i)} &= \left( \tilde{\mathbf{A}}^{(i)} \right)^T \mathbf{Y}_t^{(i)} \tilde{\mathbf{B}}^{(i)} \left( \mu_t^{(0)} \tilde{\mathbf{R}}_t^{(0)} + \left[ \sum_{j=0}^N \mu_t^{(j)} \left( \tilde{\mathbf{B}}^{(j)} \right)^T \mathbf{Y}_t^{(j)} \tilde{\mathbf{B}}^{(j)} \right] \right)^\dagger \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_t^{(i)} \tilde{\mathbf{A}}^{(i)}, \end{aligned} \quad (5.25)$$

and

$$\begin{aligned} \alpha_t^{(i)} &= \mathbb{E} \left\{ \tilde{\underline{w}}_t^T \mathbf{Y}_t^{(i)} \tilde{\underline{w}}_t \mid \mathcal{I}_t, \theta_t = i \right\} \\ &+ \sum_{r=0}^N t_{ir} \mathbb{E} \left\{ \mathbb{E} \left\{ \left( \tilde{\underline{e}}_{t+1}^{(r)} \right)^T \mathbf{S}_{t+1}^{(r)} \tilde{\underline{e}}_{t+1}^{(r)} \mid \mathcal{I}_{t+1}, \theta_{t+1} = r \right\} + \alpha_{t+1}^{(r)} \mid \mathcal{I}_t, \theta_t = i, \theta_{t+1} = r \right\}, \end{aligned}$$

which allow us to represent the cost-to-go according to (5.19), i.e., as desired, we can write  $\mathcal{V}_t$  as a weighted sum of  $N + 1$  quadratic functions

$$\begin{aligned} \mathcal{V}_t &= \min_{\underline{U}_t} \mathbb{E} \left\{ \underline{\psi}_t^T \tilde{\mathbf{Q}}_t^{(\theta_t)} \underline{\psi}_t + \underline{U}_t^T \tilde{\mathbf{R}}_t^{(\theta_t)} \underline{U}_t + \mathcal{V}_{t+1} \mid \mathcal{I}_t \right\} \\ &\approx \sum_{i=0}^N \mu_t^{(i)} \mathbb{E} \left\{ \underline{\psi}_t^T \hat{\mathbf{P}}_t^{(i)} \underline{\psi}_t + \left( \tilde{\underline{e}}_t^{(i)} \right)^T \hat{\mathbf{S}}_t^{(i)} \tilde{\underline{e}}_t^{(i)} + \alpha_t^{(i)} \mid \mathcal{I}_t, \theta_t = i \right\}. \end{aligned} \quad (5.26)$$

Observe that (5.26) depends on  $\mathbf{P}_{t+1}^{(r)}$ ,  $\mathbf{S}_{t+1}^{(r)}$ , and  $\alpha_{t+1}^{(r)}$  that appear in the quadratic representation (5.21) of the mode-conditioned cost-to-go  $\mathcal{V}_{t+1}^{(r)}$ .

To obtain recursive expressions for  $\mathbf{P}_t^{(i)}$ ,  $\mathbf{S}_t^{(i)}$ , and  $\alpha_t^{(i)}$ , we first note that the representation (5.21) trivially holds at the terminal stage  $K$  with  $\mathbf{P}_K^{(i)} = \tilde{\mathbf{Q}}_K$ ,  $\mathbf{S}_K^{(i)} = \mathbf{0}$ , and  $\alpha_K^{(i)} = 0$  for  $i = 0, 1, \dots, N$ . At stage  $K - 1$ , the mode-conditioned cost-to-go  $\mathcal{V}_{K-1}^{(i)}$  is defined as

$$\mathcal{V}_{K-1}^{(i)} = \min_{\underline{U}_{K-1}} \mathbb{E} \left\{ \underline{\psi}_{K-1}^T \tilde{\mathbf{Q}}_{K-1}^{(i)} \underline{\psi}_{K-1} + \underline{U}_{K-1}^T \tilde{\mathbf{R}}_{K-1}^{(i)} \underline{U}_{K-1} + \mathcal{V}_K \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\}.$$

We then again use the law of total probability to approximate the expectation  $E\{\mathcal{V}_K \mid \mathcal{I}_{K-1}, \theta_{K-1} = i\}$  that appears on the right of  $\mathcal{V}_{K-1}^{(i)}$  by means of the sum of the mode-conditioned cost-to-go  $\mathcal{V}_K^{(r)}$

$$\begin{aligned} E\{\mathcal{V}_K \mid \mathcal{I}_{K-1}, \theta_{K-1} = i\} &= \sum_{r=0}^N t_{ir} E\{\mathcal{V}_K \mid \mathcal{I}_{K-1}, \theta_{K-1} = i, \theta_K = r\} \\ &\stackrel{!}{\approx} \sum_{r=0}^N t_{ir} E\{\mathcal{V}_K^{(r)} \mid \mathcal{I}_{K-1}, \theta_{K-1} = i, \theta_K = r\}. \end{aligned}$$

With the definition (5.20) for  $\mathcal{V}_K^{(r)}$ , this approximation then results in

$$\begin{aligned} \mathcal{V}_{K-1}^{(i)} &= \min_{\underline{U}_{K-1}} E\left\{ \underline{\psi}_{K-1}^T \tilde{\mathbf{Q}}_{K-1}^{(i)} \underline{\psi}_{K-1} + \underline{U}_{K-1}^T \tilde{\mathbf{R}}_{K-1}^{(i)} \underline{U}_{K-1} + \mathcal{V}_K \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\} \\ &\approx E\left\{ \underline{\psi}_{K-1}^T \tilde{\mathbf{Q}}_{K-1}^{(i)} \underline{\psi}_{K-1} \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\} \\ &\quad + \min_{\underline{U}_{K-1}} \left[ E\left\{ \underline{U}_{K-1}^T \tilde{\mathbf{R}}_{K-1}^{(i)} \underline{U}_{K-1} \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\} \right. \\ &\quad \left. + \sum_{r=0}^N t_{ir} \left[ E\left\{ \underline{\psi}_K^T \mathbf{P}_K^{(r)} \underline{\psi}_K \mid \mathcal{I}_{K-1}, \theta_{K-1} = i, \theta_K = r \right\} \right] \right]. \end{aligned}$$

Using the dynamics (4.1) and setting the derivate with regards to  $\underline{U}_{K-1}$  to zero yields the minimizer

$$\underline{U}_t^{+, (i)} = - \left( \tilde{\mathbf{R}}_{K-1}^{(i)} + \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_{K-1}^{(i)} \tilde{\mathbf{B}}^{(i)} \right)^\dagger \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_{K-1}^{(i)} \tilde{\mathbf{A}}^{(i)} \hat{\underline{\psi}}_{K-1}^{(i)},$$

with  $\mathbf{Y}_{K-1}^{(i)}$  as per (5.24). Plugging  $\underline{U}_t^{+, (i)}$  back into the expression for  $\mathcal{V}_{K-1}^{(i)}$  leads to the quadratic representation

$$\mathcal{V}_{K-1}^{(i)} \approx E\left\{ \underline{\psi}_{K-1}^T \mathbf{P}_{K-1}^{(i)} \underline{\psi}_{K-1} + \left( \tilde{\underline{\epsilon}}_{K-1}^{(i)} \right)^T \mathbf{S}_{K-1}^{(i)} \tilde{\underline{\epsilon}}_{K-1}^{(i)} \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\} + \alpha_{K-1}^{(i)},$$

where

$$\begin{aligned} \mathbf{P}_{K-1}^{(i)} &= \tilde{\mathbf{Q}}_{K-1}^{(i)} + \left( \tilde{\mathbf{A}}^{(i)} \right)^T \mathbf{Y}_{K-1}^{(i)} \tilde{\mathbf{A}}^{(i)} - \mathbf{S}_{K-1}^{(i)}, \\ \mathbf{S}_{K-1}^{(i)} &= \left( \tilde{\mathbf{A}}^{(i)} \right)^T \mathbf{Y}_{K-1}^{(i)} \tilde{\mathbf{B}}^{(i)} \left( \tilde{\mathbf{R}}_{K-1}^{(i)} + \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_{K-1}^{(i)} \tilde{\mathbf{B}}^{(i)} \right)^\dagger \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_{K-1}^{(i)} \tilde{\mathbf{A}}^{(i)}, \\ \alpha_{K-1}^{(i)} &= E\left\{ \tilde{\underline{w}}_{K-1}^T \mathbf{Y}_{K-1}^{(i)} \tilde{\underline{w}}_{K-1} \mid \mathcal{I}_{K-1}, \theta_{K-1} = i \right\}. \end{aligned}$$

Observe that  $\mathbf{P}_{K-1}^{(i)}$  only depends on  $\mathbf{Y}_{K-1}^{(i)}$ , whereas  $\hat{\mathbf{P}}_{K-1}^{(i)}$  in (5.25) depends on all  $\mathbf{Y}_{K-1}^{(j)}$  via the sum in the pseudoinverse in the expression for  $\hat{\mathbf{S}}_{K-1}^{(i)}$ .

Proceeding now to stage  $K - 2$ , we can use the same reasoning to obtain an approximation of the mode-conditioned cost-to-go  $\mathcal{V}_{K-2}^{(i)}$  according to

$$\begin{aligned} \mathcal{V}_{K-2}^{(i)} &= \min_{\underline{U}_{K-2}} \mathbb{E} \left\{ \underline{\psi}_{K-2}^T \tilde{\mathbf{Q}}_{K-2}^{(i)} \underline{\psi}_{K-2} + \underline{U}_{K-2}^T \tilde{\mathbf{R}}_{K-2}^{(i)} \underline{U}_{K-2} + \mathcal{V}_{K-1} \mid \mathcal{I}_{K-2}, \theta_{K-2} = i \right\} \\ &\approx \mathbb{E} \left\{ \underline{\psi}_{K-2}^T \tilde{\mathbf{Q}}_{K-2}^{(i)} \underline{\psi}_{K-2} \mid \mathcal{I}_{K-2}, \theta_{K-2} = i \right\} + \min_{\underline{U}_{K-2}} \left[ \underline{U}_{K-2}^T \tilde{\mathbf{R}}_{K-2}^{(i)} \underline{U}_{K-2} \right. \\ &\quad \left. + \sum_{r=0}^N t_{ir} \mathbb{E} \left\{ \mathbb{E} \left\{ \underline{\psi}_{K-1}^T \mathbf{P}_{K-1}^{(r)} \underline{\psi}_{K-1} + \left( \tilde{\underline{\epsilon}}_{K-1}^{(r)} \right)^T \mathbf{S}_{K-1}^{(r)} \tilde{\underline{\epsilon}}_{K-1}^{(r)} \mid \mathcal{I}_{K-1}, \theta_{K-1} = r \right\} \right. \right. \\ &\quad \left. \left. \mid \mathcal{I}_{K-2}, \theta_{K-2} = i, \theta_{K-1} = r \right\} \right] + \sum_{r=0}^N t_{ir} \mathbb{E} \left\{ \alpha_{K-1}^{(i)} \mid \mathcal{I}_{K-2}, \theta_{K-2} = i, \theta_{K-1} = r \right\}. \end{aligned}$$

Assuming again that the estimation error  $\tilde{\underline{\epsilon}}_{K-1}^{(r)}$  does not depend on  $\underline{U}_{K-2}$  allows us to exclude it from the minimization, which we carry out to arrive at

$$\mathcal{V}_{K-2}^{(i)} \approx \mathbb{E} \left\{ \underline{\psi}_{K-2}^T \mathbf{P}_{K-2}^{(i)} \underline{\psi}_{K-2} + \left( \tilde{\underline{\epsilon}}_{K-2}^{(i)} \right)^T \mathbf{S}_{K-2}^{(i)} \tilde{\underline{\epsilon}}_{K-2}^{(i)} \mid \mathcal{I}_{K-2}, \theta_{K-2} = i \right\} + \alpha_{K-2}^{(i)}, \quad (5.27)$$

with

$$\begin{aligned} \mathbf{P}_{K-2}^{(i)} &= \tilde{\mathbf{Q}}_{K-2}^{(i)} + \left( \tilde{\mathbf{A}}^{(i)} \right)^T \mathbf{Y}_{K-2}^{(i)} \tilde{\mathbf{A}}^{(i)} - \mathbf{S}_{K-2}^{(i)}, \\ \mathbf{S}_{K-2}^{(i)} &= \left( \tilde{\mathbf{A}}^{(i)} \right)^T \mathbf{Y}_{K-2}^{(i)} \tilde{\mathbf{B}}^{(i)} \left( \tilde{\mathbf{R}}_{K-2}^{(i)} + \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_{K-2}^{(i)} \tilde{\mathbf{B}}^{(i)} \right)^\dagger \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_{K-2}^{(i)} \tilde{\mathbf{A}}^{(i)}, \\ \mathbf{Y}_{K-2}^{(i)} &= \sum_{r=0}^N t_{ir} \mathbf{P}_{K-1}^{(r)}, \end{aligned}$$

and

$$\begin{aligned} \alpha_{K-2}^{(i)} &= \mathbb{E} \left\{ \tilde{\underline{w}}_{K-2}^T \mathbf{Y}_{K-2}^{(i)} \tilde{\underline{w}}_{K-2} \mid \mathcal{I}_{K-2}, \theta_{K-2} = i \right\} + \sum_{r=0}^N t_{ir} \mathbb{E} \left\{ \alpha_{K-1}^{(i)} \mid \mathcal{I}_{K-2}, \theta_{K-2} = i, \theta_{K-1} = r \right\} \\ &\quad + \sum_{r=0}^N t_{ir} \mathbb{E} \left\{ \mathbb{E} \left\{ \left( \tilde{\underline{\epsilon}}_{K-1}^{(r)} \right)^T \mathbf{S}_{K-1}^{(r)} \tilde{\underline{\epsilon}}_{K-1}^{(r)} \mid \mathcal{I}_{K-1}, \theta_{K-1} = r \right\} \mid \mathcal{I}_{K-2}, \theta_{K-2} = i, \theta_{K-1} = r \right\}. \end{aligned}$$

Eq. (5.27) shows that  $\mathcal{V}_{K-2}^{(i)}$  is again of the form (5.21). Thus, we can establish the quadratic representation of the mode-conditioned cost-to-go for each stage  $t$  by inductive reasoning under the assumption that  $\tilde{\underline{\epsilon}}_{t+1}^{(i)}$  and  $\alpha_{t+1}^{(i)}$  be independent of  $\underline{U}_t$ . For  $t = 0, 1, \dots, K - 1$ ,  $\mathbf{P}_t^{(i)}$ ,  $\mathbf{S}_t^{(i)}$ , and  $\alpha_t^{(i)}$  are then given by the backward recursions

$$\begin{aligned} \mathbf{P}_t^{(i)} &= \tilde{\mathbf{Q}}_t^{(i)} + \left( \tilde{\mathbf{A}}^{(i)} \right)^T \mathbf{Y}_t^{(i)} \tilde{\mathbf{A}}^{(i)} - \mathbf{S}_t^{(i)}, \\ \mathbf{S}_t^{(i)} &= \left( \tilde{\mathbf{A}}^{(i)} \right)^T \mathbf{Y}_t^{(i)} \tilde{\mathbf{B}}^{(i)} \left( \tilde{\mathbf{R}}_t^{(i)} + \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_t^{(i)} \tilde{\mathbf{B}}^{(i)} \right)^\dagger \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_t^{(i)} \tilde{\mathbf{A}}^{(i)}, \\ \alpha_t^{(i)} &= \mathbb{E} \left\{ \tilde{\underline{w}}_t^T \mathbf{Y}_t^{(i)} \tilde{\underline{w}}_t \mid \mathcal{I}_t, \theta_t = i \right\} + \sum_{r=0}^N t_{ir} \mathbb{E} \left\{ \alpha_{t+1}^{(i)} \mid \mathcal{I}_t, \theta_t = i, \theta_{t+1} = r \right\} \\ &\quad + \sum_{r=0}^N t_{ir} \mathbb{E} \left\{ \mathbb{E} \left\{ \left( \tilde{\underline{\epsilon}}_{t+1}^{(r)} \right)^T \mathbf{S}_{t+1}^{(r)} \tilde{\underline{\epsilon}}_{t+1}^{(r)} \mid \mathcal{I}_{t+1}, \theta_{t+1} = r \right\} \mid \mathcal{I}_t, \theta_t = i, \theta_{t+1} = r \right\}, \end{aligned} \quad (5.28)$$

that are initialized with  $\mathbf{P}_K^{(i)} = \tilde{\mathbf{Q}}_K$ ,  $\mathbf{S}_K^{(i)} = \mathbf{0}$ , and  $\alpha_K^{(i)} = 0$ , and where

$$\mathbf{Y}_t^{(i)} = \sum_{r=0}^N t_{ir} \mathbf{P}_{t+1}^{(r)}, \quad (5.29)$$

for  $i = 0, 1, \dots, N$ . Note that the Riccati-like equations (5.28) that govern the evolution of  $\mathbf{P}_t^{(i)}$  are linearly coupled via (5.29). This can be interpreted as mixing or interaction of the results from the previous stage  $t + 1$ .

Hence, we conclude that (5.19) holds for all  $t$  so that the cost-to-go at each stage are approximated by a weighted sum of mode-conditioned quadratic functions. As per (5.23), with regards to this approximation the optimal control law is then

$$\underline{\pi}_t^* \left( \underline{\mu}_t, \hat{\underline{\psi}}_t^{(0)}, \dots, \hat{\underline{\psi}}_t^{(N)} \right) = - \left( \mu_t^{(0)} \tilde{\mathbf{R}}_t^{(0)} + \left[ \sum_{i=0}^N \mu_t^{(i)} \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_t^{(i)} \tilde{\mathbf{B}}^{(i)} \right] \right)^\dagger \sum_{i=0}^N \mu_t^{(i)} \underline{\pi}_t^{*(i)} \left( \hat{\underline{\psi}}_t^{(i)} \right), \quad (5.30)$$

for  $t = 0, 1, \dots, K - 1$ , with  $\underline{\pi}_t^{*(i)}$  given by

$$\underline{\pi}_t^{*(i)} \left( \hat{\underline{\psi}}_t^{(i)} \right) = \left( \tilde{\mathbf{B}}^{(i)} \right)^T \mathbf{Y}_t^{(i)} \tilde{\mathbf{A}}^{(i)} \hat{\underline{\psi}}_t^{(i)},$$

and  $\mathbf{Y}_t^{(i)}$  as per (5.29) for  $i = 0, 1, \dots, N$ . That is, the optimal control law consists of the weighted combination of the output of  $N + 1$  mode-conditioned control laws. Although each  $\underline{\pi}_t^{*(i)}$  is linear in the mode-conditioned state estimate  $\hat{\underline{\psi}}_t^{(i)}$ ,  $\underline{\pi}_t^*$  is highly nonlinear because the involved pseudoinverse depends on the estimated mode probabilities  $\mu_t^{(i)}$ .

Note that the derived control law is a function of  $\hat{\underline{\psi}}_t^{(i)}$  and  $\underline{\mu}_t$  only and not of the information set  $\mathcal{I}_t$ . Thus, an estimator that provides the controller with these quantities can be designed separately. As we pointed out at the end of the previous section, both  $\hat{\underline{\psi}}_t^{(i)}$  and  $\underline{\mu}_t$  are naturally provided by a multiple model estimator. A tailored estimator that can process delayed and out-of-sequence measurements and incorporate the information supplied by received ACKs is presented next.

### 5.3.2 Tailoring of a Multiple Model Estimator

Among the class of multiple model estimators, the *interacting multiple model (IMM) filter* has gained much popularity because it exhibits a good tradeoff between estimation quality and computational complexity [184]. In essence, the IMM filter consists of a bank of Kalman filters, which are individually reinitialized at each time step by “mixing” all mode-conditioned estimates from the previous time step. After a measurement is processed, the mode probabilities are updated according to the mode-conditioned measurement likelihoods. The mixing step is a particular feature of the IMM filter and not shared with other multiple model approaches. In [190] it is shown that such a mixing step is also part of the optimal yet intractable estimator, which mathematically justifies the widespread success of the IMM filter. Another merit of the IMM filter is its wide usage in target tracking applications, where delayed and out-of-sequence arrivals of measurement are typical [144, 145, 191]. Consequently, it has received some attraction also in the scope of networked control [126, 192, 193].

To handle arbitrarily delayed and out-of-sequence measurements, it was proposed to use retrodiction techniques in [191]. Retrodiction requires that  $\tilde{\mathbf{A}}^{(\theta_k)}$  in (4.1) be invertible for all modes, which, however, is never the case since  $\mathbf{F}$  (cf. (4.8)) is nilpotent and thus necessarily singular. As an alternative to retrodiction, the filter presented in [193] is equipped with a buffer to store received

measurements and estimates from previous time steps. Once a delayed measurement is received, the filter stores it in the buffer and updates the current state estimate using the buffered data. In addition to being simple, this approach is well suited to deal with multiple measurements that arrive at the same time step as they can be processed one after another. Moreover, it is easily extended to integrate the mode observations inferred from the received ACKs from the actuator. We introduce this extension in the following.

Since our network model  $\tau_k^{\text{sc}}$  allows unbounded measurement delays, it would require infinite memory to guarantee that all successfully transmitted measurements can be stored and eventually processed [194]. Thus, for a feasible solution with finite and fixed-size memory, the assumption below is necessary.

**Assumption 5.1:**

*Measurements with a delay larger than  $L \in \mathbb{N} \cup \{0\}$  time steps are discarded upon reception.*

Discarding measurements is always suboptimal so that the value of  $L$  must be chosen carefully by the designer and traded off against a potential reduction of storage and computational complexity. To determine an appropriate value, one should also consider that the lack of measurements – be it due to too many packet losses or due to a too small buffer – can lead to an unbounded estimation error covariance and eventually to the instability of the filter [195–197].

As previously stated, an IMM filter for the augmented dynamics (4.1) maintains the state estimate in the form of a Gaussian mixture with  $N + 1$  components with mean  $\hat{\underline{\psi}}_k^{(i)}$  and covariance  $\underline{\Sigma}_k^{(i)}$  as given by (5.18). The component weights are determined by the estimated mode probability distribution  $\underline{\mu}_k$ . We already demonstrated in Section 4.4 that once we know that  $\theta_k = i$  this probability distribution reduces to the  $(i + 1)$ -th standard basis vector

$$\underline{\mu}_k = \underline{e}_{N+1}^{(i+1)}. \quad (5.31)$$

At time  $k$ , integrating a mode observation, say from time  $k' < k$ , then consists of updating  $\underline{\mu}_{k'}$  according to (5.31), followed by a recomputation of the state estimates from time  $k' + 1$  to  $k$  using all the available measurements and mode observations. One cycle of the resulting IMM filter is given in Algorithm 5.2. For a detailed description of the IMM-specific mixing step in lines 7 to 14 and the update of the mode probabilities in line 30 based on the mode-conditioned measurement likelihoods  $l^{(i)}$  refer to, for instance, [184, 190]. The initial conditions for the mode-conditioned Kalman filters are

$$\hat{\underline{\psi}}_0^{(i)} = \begin{bmatrix} \hat{\underline{x}}_0 \\ \underline{0} \end{bmatrix}, \quad \underline{\Sigma}_0^{(i)} = \begin{bmatrix} \mathbf{X}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

for  $i = 0, 1, \dots, N$ , and the initial mode probability distribution is set to  $\underline{\mu}_0 = \underline{e}_{N+1}^{(N+1)}$  to reflect that the buffer at the actuator is initially empty. During the first time steps  $k' = 0, 1, \dots, L$ , the input arguments  $\hat{\underline{\psi}}_{k-(L+1)}^{(i)}$ ,  $\underline{\Sigma}_{k-(L+1)}^{(i)}$ , and  $\underline{\mu}_{k-(L+1)}^{(i)}$  are not yet existent. Instead, we call the algorithm with the initial conditions  $\hat{\underline{\psi}}_0^{(i)}$ ,  $\underline{\Sigma}_0^{(i)}$ , and  $\underline{\mu}_0^{(i)}$ , and the outermost loop runs from  $r = k' - 1$  to 0.

The memory footprint of the algorithm increases linearly with the buffer length. Specifically, it is necessary to allocate memory for the mode observations  $\theta_{k-(L+1)}, \dots, \theta_{k-1}$ , the control sequences  $\underline{U}_{k-(L+1)}, \dots, \underline{U}_{k-1}$ , the measurements  $\underline{y}_{k-L}, \dots, \underline{y}_k$ , and the Gaussian mixture representing the estimate at time  $k - (L + 1)$ . For the latter,  $N + 1$  mode-conditioned means  $\hat{\underline{\psi}}_{k-(L+1)}^{(i)}$  and covariance matrices  $\underline{\Sigma}_{k-(L+1)}^{(i)}$ , and the estimated mode distribution  $\underline{\mu}_{k-(L+1)}$  must be stored.

**Algorithm 5.2** One Cycle of the Proposed IMM Filter

**Input:**  $\mathcal{A}_k, \mathcal{Y}_k, \hat{\underline{\psi}}_{k-(L+1)}^{(i)}, \underline{\Sigma}_{k-(L+1)}^{(i)}, \mu_{k-(L+1)}^{(i)}$

**Output:**  $\hat{\underline{\psi}}_k^{(i)}, \underline{\Sigma}_k^{(i)}, \underline{\mu}_k$

```

1: Store mode observations inferred from  $\mathcal{A}_k$  // As described in Section 4.1
2: Store measurements  $\mathcal{Y}_k$ 
3: for  $r = L$  to 0 do
4:   if mode  $\theta_{k-(r+1)}$  is available then
5:     Update  $\underline{\mu}_{k-(r+1)}$  according to (5.31)
6:   end if
   // Mode-conditioned reinitialization (Mixing step)
7:   for  $i = 0$  to  $N$  do
8:      $\bar{\mu}^{(i)} = \sum_j t_{ji} \mu_{k-(r+1)}^{(j)}$  // Mode probability prediction
9:     for  $j = 0$  to  $N$  do
10:       $\mu^{j|i} = t_{ji} \frac{\mu_{k-(r+1)}^{(j)}}{\bar{\mu}^{(i)}}$  // Mixing weight between modes  $i$  and  $j$ 
11:    end for
    // Mixing estimate for mode  $i$ 
12:     $\hat{\underline{\psi}}_{\text{m}}^{(i)} = \sum_j \mu^{j|i} \hat{\underline{\psi}}_{k-(r+1)}^{(j)}$ 
13:     $\underline{\Sigma}_{\text{m}}^{(i)} = \sum_j \mu^{j|i} \left[ \underline{\Sigma}_{k-(r+1)}^{(j)} + \left( \hat{\underline{\psi}}_{\text{m}}^{(i)} - \hat{\underline{\psi}}_{k-(r+1)}^{(j)} \right) \left( \hat{\underline{\psi}}_{\text{m}}^{(i)} - \hat{\underline{\psi}}_{k-(r+1)}^{(j)} \right)^{\text{T}} \right]$ 
14:  end for
  // Mode-conditioned prediction step
15:  for  $i = 0$  to  $N$  do
16:     $\hat{\underline{\psi}}_{k-r}^{(i)} = \tilde{\mathbf{A}}^{(i)} \hat{\underline{\psi}}_{\text{m}}^{(i)} + \tilde{\mathbf{B}}^{(i)} \underline{U}_{k-(r+1)}$ 
17:     $\underline{\Sigma}_{k-r}^{(i)} = \tilde{\mathbf{A}}^{(i)} \underline{\Sigma}_{\text{m}}^{(i)} \left( \tilde{\mathbf{A}}^{(i)} \right)^{\text{T}} + \tilde{\mathbf{W}}$ 
18:  end for
19:  if  $\underline{y}_{k-r}$  is available then
20:    // Mode-conditioned measurement update
21:    for  $i = 0$  to  $N$  do
22:       $\underline{z} = \underline{y}_{k-r} - [\mathbf{C} \ \mathbf{0}] \hat{\underline{\psi}}_{k-r}^{(i)}$  // Innovation
23:       $\underline{\Sigma}^z = [\mathbf{C} \ \mathbf{0}] \underline{\Sigma}_{k-r}^{(i)} [\mathbf{C}^{\text{T}} \ \mathbf{0}]^{\text{T}} + \mathbf{V}$  // Innovation covariance matrix
24:       $\mathbf{K} = \underline{\Sigma}_{k-r}^{(i)} \mathbf{C}^{\text{T}} (\underline{\Sigma}^z)^{-1}$ 
25:       $\hat{\underline{\psi}}_{k-r}^{(i)} = \hat{\underline{\psi}}_{k-r}^{(i)} + \mathbf{K} \underline{z}$ 
26:       $\underline{\Sigma}_{k-r}^{(i)} = \underline{\Sigma}_{k-r}^{(i)} - \mathbf{K} \underline{\Sigma}^z (\mathbf{K})^{\text{T}}$ 
27:       $l^{(i)} = \frac{1}{\sqrt{\det[2\pi \underline{\Sigma}^z]}} \exp \left\{ -\frac{1}{2} \underline{z}^{\text{T}} (\underline{\Sigma}^z)^{-1} \underline{z} \right\}$  // Measurement likelihood
28:    end for
29:    for  $i = 0$  to  $N$  do
30:       $\mu_{k-r}^{(i)} = \frac{\bar{\mu}^{(i)} l^{(i)}}{\sum_j \bar{\mu}^{(j)} l^{(j)}}$  // Mode probability update
31:    end for
32:  else
33:    for  $i = 0$  to  $N$  do
34:       $\mu_{k-r}^{(i)} = \bar{\mu}^{(i)}$  // Mode probability update
35:    end for
36:  end if
37: end for
38: return  $\hat{\underline{\psi}}_k^{(i)}, \underline{\Sigma}_k^{(i)}, \underline{\mu}_k$ 

```

**Algorithm 5.3** Conceptual Algorithm for the Proposed Multiple Model Based SMPC

- 
- 1: Compute  $\mathbf{P}_1^{(0)}, \dots, \mathbf{P}_1^{(N)}$  by means of the backward recursion (5.28) initialized with  $\mathbf{P}_K^{(i)} = \tilde{\mathbf{Q}}_K$
  - 2: Compute  $\mathbf{Y}_0^{(0)}, \dots, \mathbf{Y}_0^{(N)}$  using  $\mathbf{P}_1^{(0)}, \dots, \mathbf{P}_1^{(N)}$  in (5.29)
  - 3: **for**  $k = 0, 1 \dots$  **do**
  - 4:     Estimate  $\hat{\psi}_k^{(i)}, \Sigma_k^{(i)}$ , and  $\underline{\mu}_k$  with the IMM filter (Algorithm 5.2)
  - 5:     Compute  $\underline{U}_k = \underline{\pi}_0^* \left( \underline{\mu}_k, \hat{\psi}_k^{(0)}, \dots, \hat{\psi}_k^{(N)} \right)$  using  $\mathbf{Y}_0^{(0)}, \dots, \mathbf{Y}_0^{(N)}$  in (5.30)
  - 6:     Transmit  $\underline{U}_k$  to the actuator
  - 7: **end for**
- 

**5.3.3 Summary of the Proposed SMPC Algorithm**

Combining the results of the previous two sections yields the proposed multiple model approach for sequence-based SMPC. The resulting conceptual algorithm is summarized in Algorithm 5.3. The algorithm is straightforward to implement and has fixed, moderate computational complexity at each time step. In particular, for the computation of the control sequence  $\underline{U}_k$  no costly iterative optimization is required at runtime. Moreover, the algorithm is inherently parallelizable. The mode-conditioned Kalman filters can run concurrently, and, similarly, the mode-conditioned control laws in (5.30) can be evaluated in parallel. We note that the matrices  $\mathbf{P}_1^{(i)}$  and  $\mathbf{Y}_0^{(i)}$  need to be recomputed once the controller updates the mode transition probabilities due to an update of the network model provided by the CoCPN translator. We also note that we can replace any occurrence of  $t_{ji}$  in the IMM filter algorithm by  $t_{k-r,ji}$  if  $\theta_k$  is a time-inhomogeneous Markov chain.

To conclude this section, we emphasize that the assumption that  $\hat{e}_{t+1}^{(r)}, \mathbf{S}_{t+1}^{(r)}$ , and  $\alpha_{t+1}^{(r)}$  were independent of  $\underline{U}_t$  was crucial to eliminate the dual effect, and, thereby, to obtain a tractable solution. Algorithm 5.3 is thus an *open-loop feedback* control approach because the impact of future measurements on the decision-making is not taken into account. However, as the impact of the dual effect on the cost increases with the considered horizon length  $K$ , one can expect only a slight loss of performance for small  $K$ . We illustrate this with simulation results in Section 5.5.

**5.4 SMPC Based on Local Approximation of the Cost-to-Go**

The two main building blocks of the control algorithm presented in Section 5.3 are both nonlinear with regards to the available information  $\mathcal{I}_k$ . The IMM filter because the mixing step at the beginning of each iteration and the mode probability update at the end of each iteration are nonlinear functions of the mode-conditioned state estimates and the estimated mode probabilities. The control law  $\underline{\pi}_0^*$  because the pseudoinverse in (5.30) is a function of all estimated mode probabilities. To obtain a tractable solution, we approximated the cost-to-go at each stage of the DP recursion by imposing a particular representation. This approximation of the cost-to-go is *global* because the imposed representation is valid for all possible  $\mathcal{I}_k$ , or, more precisely, for all possible state estimates, i.e., for all possible Gaussian mixtures provided by the IMM filter. For the approximated cost-to-go, we then obtained the optimal control law by disregarding the impact of the dual effect on the cost.

In this section, we present an approach for sequence-based SMPC with opposite properties. The underlying control law is a linear function of the available information  $\mathcal{I}_k$ , the approximation of the cost-to-go is only *local*, and, most importantly, the dual effect is taken into account. The gist of the approach is summarized as follows. At every step, a given estimate of the plant state is propagated forward over the optimization horizon using given control laws  $\underline{\pi}_0, \dots, \underline{\pi}_{K-1}$ . This “forward pass” yields a reference trajectory, i.e., a sequence of predicted state estimates. Then, the DP recursion is

carried out backward over the horizon, and at every stage  $t$  an upper bound of the cost-to-go, evaluated at the predicted state estimate, is obtained. This “backward pass” yields improved control laws  $\underline{\pi}_0, \dots, \underline{\pi}_{K-1}$ , which, in turn lead to a better reference trajectory in the next forward pass. Both steps, forward and backward pass, are repeated until convergence of the cost (5.2). As we wish to consider the impact of the dual effect on the cost, we must anticipate future measurements during the forward pass because they affect the quality of the predicted state estimates. This is challenging since i) the number of available measurements depends on the delays and losses during their transmission to the controller and ii) the values that will be measured in the future cannot be known in advance.

Similar to [124], we address the first challenge by an additional state augmentation in Section 5.4.1, leading to a hybrid system with a second discrete-valued variable  $\underline{\gamma}_k$  that encodes the measurement availability. Based on this state augmentation, we then tackle the second challenge in Section 5.4.2 and reformulate the cost function (5.2) assuming linear control laws  $\underline{\pi}_0, \dots, \underline{\pi}_{K-1}$ . In contrast to the reformulation (5.4), the key idea here is to express the cost in terms of the second moment of the augmented state rather than in terms of the augmented state directly. This is a common strategy in literature because the dynamics of the second moment is linear and depends only on the availability of measurements and *not* on the measured values [127, 177, 180, 198]. Predicting a given second moment in time thus inherently anticipates future measurement information. Equipped with these prerequisites, we then derive the proposed iterative method for the computation of the control laws in Section 5.4.3. Finally, Section 5.4.4 summarizes the resulting algorithm for sequence-based SMPC.

#### 5.4.1 Definition of the Augmented Dynamical System

Assumption 5.1 implies that it suffices to consider the availability of the measurements  $\underline{y}_{k-L}, \dots, \underline{y}_k$ . Whether  $\underline{y}_{k-l}$  is available to the controller for processing at time  $k$  is encoded by the binary random variable  $\gamma_{k|k-l} \in \{0, 1\}$  defined as

$$\gamma_{k|k-l} = \begin{cases} 1 & \underline{y}_{k-l} \in \mathcal{Y}_k \\ 0 & \underline{y}_{k-l} \notin \mathcal{Y}_k \end{cases}, \quad (5.32)$$

for  $l = 0, 1, \dots, L$ . Note that  $\gamma_{k|k-l} = 0$  means that  $\underline{y}_{k-l}$  has already arrived at an earlier time step or has not yet arrived at all. In particular, (5.32) states that  $\gamma_{k|k-l} = 1 \Rightarrow \gamma_{k'|k-l} = 0$  for  $k' > k$  so that each measurement is processed at most once. Based on (5.32), we then introduce the augmented measurement

$$\bar{\underline{y}}_k = \left[ \tilde{\underline{y}}_{k|k}^T \tilde{\underline{y}}_{k|k-1}^T \cdots \tilde{\underline{y}}_{k|k-L}^T \right]^T \in \mathbb{R}^{(L+1)n_y},$$

with

$$\tilde{\underline{y}}_{k|k-l} = \gamma_{k|k-l} \underline{y}_{k-l} = \gamma_{k|k-l} \mathbf{C} \underline{x}_{k-l} + \gamma_{k|k-l} \underline{v}_{k-l}, \quad (5.33)$$

for  $l = 0, 1, \dots, L$ . Note that if  $\underline{y}_{k-l}$  is not available ( $\gamma_{k|k-l} = 0$ ), the corresponding element  $\tilde{\underline{y}}_{k|k-l}$  in  $\bar{\underline{y}}_k$  becomes  $\underline{0}$ . On the other hand, in case the measurement  $\underline{y}_{k-l} = \underline{0}$  is available, we always have  $\gamma_{k|k-l} = 1$ . Hence, the two situations with  $\tilde{\underline{y}}_{k|k-l} = \underline{0}$  can be distinguished by the controller. In addition to the current state  $\underline{x}_k$ ,  $\bar{\underline{y}}_k$  also affects the past states  $\underline{x}_{k-1}, \dots, \underline{x}_{k-L}$ , so that it is natural to define an augmented state  $\underline{\xi}_k$  according to

$$\underline{\xi}_k = \left[ \underline{x}_k^T \underline{x}_{k-1}^T \cdots \underline{x}_{k-L}^T \bar{\underline{y}}_k^T \right]^T \in \mathbb{R}^{n_\xi},$$

with  $\underline{\eta}_k$  as defined in (4.4) and where  $n_\xi = (L+1)n_x + \frac{(N-1)Nn_u}{2}$ .

With  $\underline{\bar{w}}_k = [\underline{w}_k^T \ \underline{0} \ \dots \ \underline{0}]^T$ , (3.1), and (4.7), we get

$$\underline{\xi}_{k+1} = \overline{\mathbf{A}}_k^{(\theta_k)} \underline{\xi}_k + \overline{\mathbf{B}}_k^{(\theta_k)} \underline{U}_k + \underline{\bar{w}}_k, \quad (5.34)$$

for the dynamics of  $\underline{\xi}_k$ , where  $\overline{\mathbf{A}}_k^{(\theta_k)} \in \mathbb{R}^{n_\xi \times n_\xi}$  and  $\overline{\mathbf{B}}_k^{(\theta_k)} \in \mathbb{R}^{n_\xi \times Nn_u}$  are given given by

$$\overline{\mathbf{A}}_k^{(\theta_k)} = \begin{array}{c} \text{\#columns:} \\ \begin{bmatrix} n_x & n_x & \dots & n_x & \frac{(N-1)Nn_u}{2} \\ \mathbf{A} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{BH}^{(\theta_k)} \\ \mathbf{I} & \mathbf{0} & \ddots & \vdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{F} \end{bmatrix} \\ \text{\#rows:} \end{array}, \quad \overline{\mathbf{B}}_k^{(\theta_k)} = \begin{array}{c} \text{\#columns:} \\ \begin{bmatrix} Nn_u \\ \mathbf{BJ}^{(\theta_k)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{G} \end{bmatrix} \\ \text{\#rows:} \end{array},$$

with  $\mathbf{H}^{(\theta_k)}$  and  $\mathbf{J}^{(\theta_k)}$  as per (4.6), and  $\mathbf{F}$  and  $\mathbf{G}$  according to (4.8) and (4.9). The noise  $\underline{\bar{w}}_k$  is zero-mean and white with covariance matrix

$$\overline{\mathbf{W}} = \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n_\xi \times n_\xi}.$$

Similarly, with the binary vector

$$\underline{\gamma}_k = [\gamma_{k|k} \ \gamma_{k|k-1} \ \dots \ \gamma_{k|k-L}]^T \in \{0, 1\}^{L+1}, \quad (5.35)$$

we then define the augmented measurement equation

$$\underline{\bar{y}}_k = \overline{\mathbf{S}}^{(\underline{\gamma}_k)} \overline{\mathbf{C}} \underline{\xi}_k + \overline{\mathbf{S}}^{(\underline{\gamma}_k)} \underline{\bar{v}}_k, \quad (5.36)$$

with  $\underline{\bar{v}}_k = [\underline{v}_k^T \ \underline{v}_{k-1}^T \ \dots \ \underline{v}_{k-L}^T]^T$ , and where  $\overline{\mathbf{S}}^{(\underline{\gamma}_k)} \in \mathbb{R}^{(L+1)n_y \times (L+1)n_y}$  and  $\overline{\mathbf{C}} \in \mathbb{R}^{(L+1)n_y \times n_\xi}$  are given by

$$\overline{\mathbf{S}}^{(\underline{\gamma}_k)} = \begin{bmatrix} \gamma_{k|k} \mathbf{I}_{n_y} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \gamma_{k|k-1} \mathbf{I}_{n_y} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \gamma_{k|k-L} \mathbf{I}_{n_y} \end{bmatrix}, \quad \overline{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & \ddots & & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{C} & \mathbf{0} \end{bmatrix}.$$

The augmented measurement noise  $\underline{\bar{v}}_k$  is zero-mean and white with covariance matrix

$$\overline{\mathbf{V}} = \begin{bmatrix} \mathbf{V} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{V} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{V} \end{bmatrix} \in \mathbb{R}^{(L+1)n_y \times (L+1)n_y}.$$

We emphasize that although  $\underline{\gamma}_k$  is a random quantity, its realization is always known to the controller because  $\mathcal{Y}_k \in \mathcal{I}_k$ .

The combination of (5.34) and (5.36) yields the augmented dynamical system

$$\begin{aligned} \underline{\xi}_{k+1} &= \overline{\mathbf{A}}_k^{(\theta_k)} \underline{\xi}_k + \overline{\mathbf{B}}_k^{(\theta_k)} \underline{U}_k + \underline{\bar{w}}_k, \\ \underline{\bar{y}}_k &= \overline{\mathbf{S}}^{(\underline{\gamma}_k)} \overline{\mathbf{C}} \underline{\xi}_k + \overline{\mathbf{S}}^{(\underline{\gamma}_k)} \underline{\bar{v}}_k. \end{aligned} \quad (5.37)$$

Eq. (5.37) describes a hybrid system with two independent operation modes  $\theta_k$  and  $\underline{\gamma}_k$ . As before,  $\theta_k$  is a Markov chain with state space  $\{0, 1, \dots, N\}$  and only a subset of its history  $\theta_0, \dots, \theta_k$  is known to the controller. In contrast, the state space of  $\underline{\gamma}_k$  consists of  $2^{L+1}$  distinct states and its history  $\underline{\gamma}_0, \dots, \underline{\gamma}_k$  is completely known. Nothing can be said about the properties of the stochastic process  $\underline{\gamma}_k$  without further assumptions regarding the network model  $\tau_k^{\text{sc}}$ . However, we can predict the availability of measurements at future time steps  $k' > k$  by virtue of

$$\begin{aligned} \text{P}[\gamma_{k'|k'-l} = 1 | \mathcal{I}_k] &= \text{P}[\tau_{k'-l}^{\text{sc}} = l | \mathcal{I}_k], \\ \text{P}[\gamma_{k'|k'-l} = 0 | \mathcal{I}_k] &= \text{P}[\tau_{k'-l}^{\text{sc}} \neq l | \mathcal{I}_k] = 1 - \text{P}[\tau_{k'-l}^{\text{sc}} = l | \mathcal{I}_k], \end{aligned}$$

for  $l = 0, 1, \dots, L$ . For example, if at time  $k$  the measurement  $\underline{y}_{k-2}$  has not been received yet<sup>5</sup>, we get for its availability at time  $k+1$

$$\begin{aligned} \text{P}[\gamma_{k+1|k-2} = 1 | \mathcal{I}_k] &= \text{P}[\tau_{k-2}^{\text{sc}} = 3 | \tau_{k-2}^{\text{sc}} > 2], \\ \text{P}[\gamma_{k+1|k-2} = 0 | \mathcal{I}_k] &= \text{P}[\tau_{k-2}^{\text{sc}} > 3 | \tau_{k-2}^{\text{sc}} > 2]. \end{aligned}$$

Similarly, if at time  $k$  the measurement  $\underline{y}_{k-1}$  is available, it holds

$$\begin{aligned} \text{P}[\gamma_{k'|k-1} = 1 | \mathcal{I}_k] &= 0, \\ \text{P}[\gamma_{k'|k-1} = 0 | \mathcal{I}_k] &= 1. \end{aligned}$$

for  $k' > k$ . Using (5.35), the probability of a future mode realization  $\underline{\gamma}_{k'} = (i_0, i_1, \dots, i_L)$ , where  $i_0, i_1, \dots, i_L \in \{0, 1\}$ , is then given by

$$\text{P}[\underline{\gamma}_{k'} = (i_0, i_1, \dots, i_L) | \mathcal{I}_k] = \text{P}[\gamma_{k'|k'} = i_0, \gamma_{k'|k'-1} = i_1, \dots, \gamma_{k'|k'-L} = i_L | \mathcal{I}_k]. \quad (5.38)$$

### 5.4.2 Reformulation of the Cost Function

With the augmented dynamical system (5.37) derived in the previous section, the availability of measurements can be anticipated during the propagation of a state estimate over the optimization horizon. Hence, the dual effect is taken into account during computation of the control laws  $\underline{\pi}_0, \dots, \underline{\pi}_{K-1}$ . To remain computationally tractable, we restrict our attention to control laws with a fixed parametrization. More precisely, we demand that each  $\underline{\pi}_t$  be a linear function of the available information  $\mathcal{I}_t$ . That is, at each stage of the optimization horizon we seek to find a linear control law of the form

$$\begin{aligned} \hat{\underline{\xi}}_{t+1} &= \hat{\mathbf{A}}_t \hat{\underline{\xi}}_t + \hat{\mathbf{B}}_t \underline{U}_t + \mathbf{K}_t \left( \underline{\bar{y}}_t - \hat{\mathbf{S}}_t \overline{\mathbf{C}} \hat{\underline{\xi}}_t \right), \\ \underline{U}_t &= \mathbf{L}_t \hat{\underline{\xi}}_t, \end{aligned} \quad (5.39)$$

for  $t = 0, 1, \dots, K-1$ , where  $\hat{\underline{\xi}}_t \in \mathbb{R}^{n_\xi}$  is the controller's estimate of the augmented state  $\underline{\xi}_t$  and with  $\hat{\mathbf{A}}_t \in \mathbb{R}^{n_\xi \times n_\xi}$ ,  $\hat{\mathbf{B}}_t \in \mathbb{R}^{n_\xi \times N n_u}$ , and  $\hat{\mathbf{S}}_t \in \mathbb{R}^{(L+1)n_y \times (L+1)n_y}$  given by<sup>6</sup>

$$\hat{\mathbf{A}}_t = \mathbb{E}_{\theta_t} \left\{ \overline{\mathbf{A}}_t^{(\theta_t)} \mid \mathcal{I}_0 \right\}, \quad \hat{\mathbf{B}}_t = \mathbb{E}_{\theta_t} \left\{ \overline{\mathbf{B}}_t^{(\theta_t)} \mid \mathcal{I}_0 \right\}, \quad \hat{\mathbf{S}}_t = \mathbb{E}_{\underline{\gamma}_t} \left\{ \overline{\mathbf{S}}^{(\underline{\gamma}_t)} \mid \mathcal{I}_0 \right\}. \quad (5.40)$$

The parameters of the control law  $\mathbf{K}_t \in \mathbb{R}^{n_\xi \times (L+1)n_y}$  and  $\mathbf{L}_t \in \mathbb{R}^{N n_u \times n_\xi}$  shall minimize the cost (5.2), i.e., they shall be such that  $(\mathbf{K}_0, \mathbf{L}_0), \dots, (\mathbf{K}_{K-1}, \mathbf{L}_{K-1})$  minimizes (5.2) among all possible sequences of parameters  $(\mathbf{K}'_0, \mathbf{L}'_0), \dots, (\mathbf{K}'_{K-1}, \mathbf{L}'_{K-1})$ . Note that  $\mathbf{K}_t$  plays the role of an

<sup>5</sup>In other words,  $\underline{y}_{k-2}$  is not contained in  $\mathcal{Y}_{k-2}, \mathcal{Y}_{k-1}$ , or  $\mathcal{Y}_k$  and, thus, not part of  $\mathcal{I}_k$ .

<sup>6</sup>In the following,  $\mathcal{I}_0$  always denotes the information set at the initial stage  $t = 0$  of the optimization horizon.

estimator gain in (5.39). Hence, we do not separate the design of a state estimator from the design of a controller, as we did in Section 5.3.

The expectations in (5.40) can be evaluated according to

$$\hat{\mathbf{A}}_t = \mathbb{E}\left\{\overline{\mathbf{A}}_t^{(\theta_t)} \mid \mathcal{I}_0\right\} = \sum_{i=0}^N \mu_t^{(i)} \overline{\mathbf{A}}_t^{(i)}, \quad (5.41)$$

$$\hat{\mathbf{B}}_t = \mathbb{E}\left\{\overline{\mathbf{B}}_t^{(\theta_t)} \mid \mathcal{I}_0\right\} = \sum_{i=0}^N \mu_t^{(i)} \overline{\mathbf{B}}_t^{(i)}, \quad (5.42)$$

$$\hat{\mathbf{S}}_t = \mathbb{E}\left\{\overline{\mathbf{S}}^{(\gamma_t)} \mid \mathcal{I}_0\right\} = \sum_{\substack{(i_0, i_1, \dots, i_L) \\ i_0, i_1, \dots, i_L \in \{0, 1\}}} \mathbb{P}[\gamma_t = (i_0, i_1, \dots, i_L) \mid \mathcal{I}_0] \overline{\mathbf{S}}^{(i_0, i_1, \dots, i_L)}, \quad (5.43)$$

with  $\mu_t^{(i)} = \mathbb{P}[\theta_t = i \mid \mathcal{I}_0]$  and  $\mathbb{P}[\gamma_t = (i_0, i_1, \dots, i_L) \mid \mathcal{I}_0]$  as in (5.38). Note that the controller knows the realization  $\gamma_0 = (i_0, i_1, \dots, i_L)$  at the initial stage of the optimization horizon, so that  $\hat{\mathbf{S}}_0 = \overline{\mathbf{S}}^{(i_0, i_1, \dots, i_L)}$ . Thus, using  $\hat{\mathbf{S}}_t$  in place of  $\overline{\mathbf{S}}^{(\gamma_t)}$  in the control law (5.39) seems unnecessary. However, propagating the state estimate  $\hat{\underline{x}}_0$  forward over the horizon with  $\overline{\mathbf{S}}^{(\gamma_t)}$  would require to consider all possible paths of the evolution of  $\gamma_0$  which is impractical because the number of paths grows exponentially with  $t$ . As will become clear soon, using  $\hat{\mathbf{S}}_t$  instead of  $\overline{\mathbf{S}}^{(\gamma_t)}$  ensures that the optimization problem remains tractable.

Propagating a given state estimate over the horizon by means of (5.39) involves an innovation term  $\overline{\underline{y}}_t - \hat{\mathbf{S}}_t \overline{\underline{\xi}}_t$  at each stage. Computing the innovation for future stages  $t + i$  is, however, not possible since the measurement vector  $\overline{\underline{y}}_{t+i}$  is not yet known at stage  $t$ . Thus, (5.39) cannot be used directly to obtain control laws that consider the impact of future measurement information. The first step towards a solution of this problem is the construction of the closed-loop dynamics, which, by combining the augmented dynamics (5.37) and the control law (5.39), is

$$\tilde{\underline{x}}_{t+1} = \mathbf{\Gamma}_t^{(\theta_t)} \tilde{\underline{x}}_t + \underline{o}_t, \quad (5.44)$$

for  $t = 0, 1, \dots, K - 1$ , with  $\tilde{\underline{x}}_t \in \mathbb{R}^{2n_\xi}$ ,  $\mathbf{\Gamma}_t^{(\theta_t)} \in \mathbb{R}^{2n_\xi \times 2n_\xi}$ , and  $\underline{o}_t \in \mathbb{R}^{2n_\xi}$  given by

$$\tilde{\underline{x}}_t = \begin{bmatrix} \overline{\underline{\xi}}_t \\ \underline{\xi}_t \end{bmatrix}, \quad \mathbf{\Gamma}_t^{(\theta_t)} = \begin{bmatrix} \overline{\mathbf{A}}_t^{(\theta_t)} & \overline{\mathbf{B}}_t^{(\theta_t)} \mathbf{L}_t \\ \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} & \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \end{bmatrix}, \quad \underline{o}_t = \begin{bmatrix} \overline{\underline{w}}_t \\ \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\underline{v}}_t \end{bmatrix}. \quad (5.45)$$

Then, we decompose the second moment of the closed-loop state  $\tilde{\mathbf{X}}_t = \mathbb{E}_{\tilde{\underline{x}}_t} \{\tilde{\underline{x}}_t \tilde{\underline{x}}_t^T \mid \mathcal{I}_0\}$  into  $N + 1$  matrices by means of the indicator function according to

$$\tilde{\mathbf{X}}_t^{(i)} = \mathbb{E}_{\tilde{\underline{x}}_t, \theta_t} \{\tilde{\underline{x}}_t \tilde{\underline{x}}_t^T \mathbb{1}_{\{\theta_t = i\}} \mid \mathcal{I}_0\}, \quad (5.46)$$

for  $i = 0, 1, \dots, N$ , so that it holds [127, p. 31]

$$\tilde{\mathbf{X}}_t = \sum_{i=0}^N \tilde{\mathbf{X}}_t^{(i)}. \quad (5.47)$$

Note that  $\tilde{\mathbf{X}}_t^{(i)}$  is closely related to the mode-conditioned second moment  $\mathbb{E}_{\tilde{\underline{x}}_t} \{\tilde{\underline{x}}_t \tilde{\underline{x}}_t^T \mid \mathcal{I}_0, \theta_t = i\}$  since evaluation of the expectation in (5.46) reveals that

$$\tilde{\mathbf{X}}_t^{(i)} = \mu_t^{(i)} \mathbb{E}_{\tilde{\underline{x}}_t} \{\tilde{\underline{x}}_t \tilde{\underline{x}}_t^T \mid \mathcal{I}_0, \theta_t = i\}.$$

Thus, (5.47) is a compact formulation of the law of total expectation.

Let  $\mathbf{N}_t$  denote the covariance matrix of  $\underline{o}_t$  given by

$$\mathbf{N}_t = \begin{bmatrix} \overline{\mathbf{W}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{V}} (\mathbf{K}_t \hat{\mathbf{S}}_t)^\top \end{bmatrix} \in \mathbb{R}^{2n_\xi \times 2n_\xi}. \quad (5.48)$$

Regarding the dynamics of  $\tilde{\mathbf{X}}_t^{(i)}$ , we then have the following result.

**Lemma 5.2:**

For  $t = 0, 1, \dots, K-1$ , it holds for all  $j = 0, 1, \dots, N$

$$\tilde{\mathbf{X}}_{t+1}^{(j)} = \sum_{i=0}^N t_{ij} \left( \mathbf{\Gamma}_t^{(i)} \tilde{\mathbf{X}}_t^{(i)} (\mathbf{\Gamma}_t^{(i)})^\top + \mu_t^{(i)} \mathbf{N}_t \right). \quad (5.49)$$

*Proof.* The proof involves straightforward calculations and can be found in Appendix C.2.  $\square$

Note that the dynamics (5.49), and hence also the dynamics of the second moment  $\tilde{\mathbf{X}}_t$ , is linear and only dependent on the availability of measurements (via  $\hat{\mathbf{S}}_t$ ) and *not* on the measured values. We can observe the same for the dynamics of the covariance matrix. This is a consequence of the linearity of the control laws (5.39) and in stark contrast to the approach presented in Section 5.3. There, the dynamics of the estimation error covariance matrix was nonlinear and dependent on the measured values due to the mixing step and the mode probability update of the IMM filter (cf. Algorithm 5.2).

For the reformulation of the cost function (5.2) in terms of the second moment, we first define

$$\begin{aligned} \overline{\mathbf{J}}_t^{(\theta_t)} &= \left( \mathbf{J}^{(\theta_t)} \mathbf{L}_t \right)^\top \mathbf{R}_t \mathbf{J}^{(\theta_t)} \mathbf{L}_t \in \mathbb{R}^{n_\xi \times n_\xi}, \\ \overline{\mathbf{H}}_t^{(\theta_t)} &= \left( \mathbf{H}^{(\theta_t)} \right)^\top \mathbf{R}_t \mathbf{H}^{(\theta_t)} \in \mathbb{R}^{\frac{(N-1)Nn_u}{2} \times \frac{(N-1)Nn_u}{2}}, \end{aligned}$$

and

$$\begin{aligned} \overline{\mathbf{Q}}_K &= \begin{bmatrix} \mathbf{Q}_K & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n_\xi \times n_\xi}, & \overline{\mathbf{Q}}_t^{(\theta_t)} &= \begin{bmatrix} \mathbf{Q}_t & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \overline{\mathbf{H}}_t^{(\theta_t)} \end{bmatrix} \in \mathbb{R}^{n_\xi \times n_\xi}, \\ \hat{\mathbf{Q}}_K &= \begin{bmatrix} \overline{\mathbf{Q}}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2n_\xi \times 2n_\xi}, & \hat{\mathbf{Q}}_t^{(\theta_t)} &= \begin{bmatrix} \overline{\mathbf{Q}}_t^{(\theta_t)} & \mathbf{0} \\ \mathbf{0} & \overline{\mathbf{J}}_t^{(\theta_t)} \end{bmatrix} \in \mathbb{R}^{2n_\xi \times 2n_\xi}, \end{aligned} \quad (5.50)$$

for  $t = 0, 1, \dots, K-1$ . Then, we write the cost function in terms of the closed-loop state  $\tilde{\mathbf{x}}_t$  as

$$\mathcal{J}(\mathbf{K}_{0:K-1}, \mathbf{L}_{0:K-1}) = \mathbb{E}_{\substack{\tilde{\mathbf{x}}_{0:K}, \\ \theta_{0:K-1}}} \left\{ \tilde{\mathbf{x}}_K^\top \hat{\mathbf{Q}}_K \tilde{\mathbf{x}}_K + \sum_{t=0}^{K-1} \tilde{\mathbf{x}}_t^\top \hat{\mathbf{Q}}_t^{(\theta_t)} \tilde{\mathbf{x}}_t \mid \mathcal{I}_0 \right\}. \quad (5.51)$$

For the expectation on the right side of (5.51) we may use that

$$\begin{aligned}
\mathbb{E}_{\tilde{\mathbf{x}}_K} \left\{ \tilde{\mathbf{x}}_K^T \hat{\mathbf{Q}}_K \tilde{\mathbf{x}}_K \mid \mathcal{I}_0 \right\} &= \sum_{i=0}^N \mathbb{E}_{\tilde{\mathbf{x}}_K, \theta_K} \left\{ \tilde{\mathbf{x}}_K^T \hat{\mathbf{Q}}_K \tilde{\mathbf{x}}_K \mathbb{1}_{\{\theta_K=i\}} \mid \mathcal{I}_0 \right\} \\
&= \sum_{i=0}^N \mathbb{E}_{\tilde{\mathbf{x}}_K, \theta_K} \left\{ \text{tr} \left[ \tilde{\mathbf{x}}_K^T \hat{\mathbf{Q}}_K \tilde{\mathbf{x}}_K \mathbb{1}_{\{\theta_K=i\}} \right] \mid \mathcal{I}_0 \right\} \\
&= \sum_{i=0}^N \text{tr} \left[ \mathbb{E}_{\tilde{\mathbf{x}}_K, \theta_K} \left\{ \hat{\mathbf{Q}}_K \tilde{\mathbf{x}}_K \tilde{\mathbf{x}}_K^T \mathbb{1}_{\{\theta_K=i\}} \mid \mathcal{I}_0 \right\} \right] \\
&= \sum_{i=0}^N \text{tr} \left[ \hat{\mathbf{Q}}_K \tilde{\mathbf{X}}_K^{(i)} \right],
\end{aligned}$$

and, similarly, for  $t = 0, 1, \dots, K-1$

$$\mathbb{E}_{\tilde{\mathbf{x}}_t, \theta_t} \left\{ \tilde{\mathbf{x}}_t^T \hat{\mathbf{Q}}_t^{(\theta_t)} \tilde{\mathbf{x}}_t \mid \mathcal{I}_0 \right\} = \sum_{i=0}^N \mathbb{E}_{\tilde{\mathbf{x}}_t, \theta_t} \left\{ \tilde{\mathbf{x}}_t^T \hat{\mathbf{Q}}_t^{(i)} \tilde{\mathbf{x}}_t \mathbb{1}_{\{\theta_t=i\}} \mid \mathcal{I}_0 \right\} = \sum_{i=0}^N \text{tr} \left[ \hat{\mathbf{Q}}_t^{(i)} \tilde{\mathbf{X}}_t^{(i)} \right],$$

which yields

$$\mathcal{J}(\mathbf{K}_{0:K-1}, \mathbf{L}_{0:K-1}) = \sum_{i=0}^N \text{tr} \left[ \hat{\mathbf{Q}}_K \tilde{\mathbf{X}}_K^{(i)} \right] + \sum_{t=0}^{K-1} \sum_{i=0}^N \text{tr} \left[ \hat{\mathbf{Q}}_t^{(i)} \tilde{\mathbf{X}}_t^{(i)} \right]. \quad (5.52)$$

For a given sequence of control law parameters  $(\mathbf{K}_0, \mathbf{L}_0), \dots, (\mathbf{K}_{K-1}, \mathbf{L}_{K-1})$ , (5.52) expresses the cost in terms of the second moment of the closed-loop state. Minimizing (5.52) subject to the second moment dynamics (5.49) leads to closed-loop feedback laws because the influence of future measurement information is considered. Similarly, the impact of the control law parameters  $(\mathbf{K}_t, \mathbf{L}_t)$  on future state estimates, expressed by means of the second moments  $\tilde{\mathbf{X}}_{t+1}^{(i)}, \tilde{\mathbf{X}}_{t+2}^{(i)}, \dots, \tilde{\mathbf{X}}_{t+K}^{(i)}$  is taken into account. Note that using  $\bar{\mathbf{S}}^{(\gamma_t)}$  in place of  $\hat{\mathbf{S}}_t$  in the control law (5.39) would have made the closed-loop dynamics and the second moment dynamics dependent on  $\underline{\gamma}_t$ . This, in turn, would have led to a decomposition of the second moment into  $(N+1)2^{L+1}$  matrices  $\tilde{\mathbf{X}}_t^{(i, j_0, j_1, \dots, j_L)}$ , rendering the minimization of (5.52) intractable even for small  $L$ .

We can formulate the cost function (5.52) more conveniently as a backward recursion similar to the DP recursion (5.6). Before we do so, let us first introduce some notation to simplify the exposition. Define the space

$$\mathbb{H}^n \triangleq \left\{ \left( \mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)} \right) \mid \mathbf{X}^{(i)} \in \mathbb{R}^{n \times n} \right\},$$

that is composed of all  $(N+1)$ -tuples of  $n$ -by- $n$  matrices.  $\mathbb{H}^n$  is a vector space with addition and scalar multiplication defined as

$$\begin{aligned}
\mathcal{X} + \mathcal{Y} &\triangleq \left( \mathbf{X}^{(0)} + \mathbf{Y}^{(0)}, \mathbf{X}^{(1)} + \mathbf{Y}^{(1)}, \dots, \mathbf{X}^{(N)} + \mathbf{Y}^{(N)} \right), \\
\alpha \mathcal{X} &\triangleq \left( \alpha \mathbf{X}^{(0)}, \alpha \mathbf{X}^{(1)}, \dots, \alpha \mathbf{X}^{(N)} \right),
\end{aligned}$$

for  $\mathcal{X} = (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)})$ ,  $\mathcal{Y} = (\mathbf{Y}^{(0)}, \mathbf{Y}^{(1)}, \dots, \mathbf{Y}^{(N)}) \in \mathbb{H}^n$ , and  $\alpha \in \mathbb{R}$ , i.e., the usual addition and scalar multiplication on  $\mathbb{R}^{n \times n}$  is applied elementwise.<sup>7</sup> On  $\mathbb{H}^n$ , define the family of

<sup>7</sup>More precisely,  $\mathbb{H}^n$  is a Banach space (cf. Appendix A.7). We will make use of this fact in Chapter 6.

mappings  $\mathcal{E}^{(i)}: \mathbb{H}^n \mapsto \mathbb{R}^{n \times n}$  according to

$$\mathcal{E}^{(i)}(\mathcal{X}) \triangleq \sum_{j=0}^{N+1} t_{ij} \mathbf{X}^{(j)},$$

for any  $\mathcal{X} = (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)}) \in \mathbb{H}^n$  for  $i = 0, 1, \dots, N$ . With this definition, we can evaluate the cost (5.52) for a given sequence of control law parameters recursively according to the following theorem.

**Theorem 5.1:**

Fix a sequence of control law parameters  $(\mathbf{K}_0, \mathbf{L}_0), \dots, (\mathbf{K}_{K-1}, \mathbf{L}_{K-1})$ . Then, the cost-to-go at each stage  $t = 0, 1, \dots, K$  is given by

$$\mathcal{V}_t = \sum_{i=0}^N \text{tr} \left[ \tilde{\mathbf{P}}_t^{(i)} \tilde{\mathbf{X}}_t^{(i)} \right] + \mu_t^{(i)} \tilde{\omega}_t^{(i)}, \quad (5.53)$$

with  $\tilde{\mathbf{P}}_t^{(i)}$  and  $\tilde{\omega}_t^{(i)}$  computed according to the backward recursions

$$\tilde{\mathbf{P}}_t^{(i)} = \hat{\mathbf{Q}}_t^{(i)} + \left( \mathbf{\Gamma}_t^{(i)} \right)^T \mathcal{E}^{(i)} \left( \tilde{\mathcal{P}}_{t+1} \right) \mathbf{\Gamma}_t^{(i)}, \quad (5.54)$$

$$\tilde{\omega}_t^{(i)} = \mathcal{E}^{(i)}(\tilde{\omega}_{t+1}) + \text{tr} \left[ \mathcal{E}^{(i)} \left( \tilde{\mathcal{P}}_{t+1} \right) \mathbf{N}_t \right], \quad (5.55)$$

that are initialized with  $\tilde{\mathbf{P}}_K^{(i)} = \hat{\mathbf{Q}}_K$  and  $\tilde{\omega}_K^{(i)} = 0$  for  $i = 0, 1, \dots, N$ , and where

$$\begin{aligned} \tilde{\mathcal{P}}_t &= \left( \tilde{\mathbf{P}}_t^{(0)}, \tilde{\mathbf{P}}_t^{(1)}, \dots, \tilde{\mathbf{P}}_t^{(N)} \right) \in \mathbb{H}^{2n\epsilon}, \\ \tilde{\omega}_t &= \left( \tilde{\omega}_t^{(0)}, \tilde{\omega}_t^{(1)}, \dots, \tilde{\omega}_t^{(N)} \right) \in \mathbb{H}^1. \end{aligned}$$

*Proof.* The proof is given in Appendix C.3. □

In particular, we have  $\mathcal{J}(\mathbf{K}_{0:K-1}, \mathbf{L}_{0:K-1}) = \mathcal{V}_0$ . Note that at each stage,  $\mathcal{V}_t$  is a function of the corresponding control law parameters  $(\mathbf{K}_t, \mathbf{L}_t)$  only (via  $\hat{\mathbf{Q}}_t^{(i)}$ ,  $\mathbf{\Gamma}_t^{(i)}$ , and  $\mathbf{N}_t$  in (5.54) and (5.55)). The contribution of the control law parameters  $(\mathbf{K}_{t+1:K-1}, \mathbf{L}_{t+1:K-1})$  to  $\mathcal{V}_t$  is latent via  $\tilde{\mathcal{P}}_{t+1}$  and  $\tilde{\omega}_{t+1}$ , which can be interpreted as the *costate* variables [199]. Consequently, (5.53) can be viewed as the *Hamiltonian* associated with the minimization of the cost (5.52) subject to the second moment dynamics (5.49).

### 5.4.3 Derivation of the Control Laws

Theorem 5.1 suggests a straightforward approach for the iterative computation of the control law parameters that minimize the cost (5.52) for a given second moment  $\tilde{\mathbf{X}}_0^{(0)}, \dots, \tilde{\mathbf{X}}_0^{(K)}$ . Starting with initial control law parameters  $(\mathbf{K}_{0:K-1}^{[0]}, \mathbf{L}_{0:K-1}^{[0]})$  with cost  $\mathcal{J}^{[0]} = \mathcal{J}(\mathbf{K}_{0:K-1}^{[0]}, \mathbf{L}_{0:K-1}^{[0]})$ , a forward pass is carried out to propagate the initial condition  $\tilde{\mathbf{X}}_0^{(i)}$  over the horizon, yielding a reference trajectory  $\tilde{\mathbf{X}}_{0:K}^{(0,[1])}, \tilde{\mathbf{X}}_{0:K}^{(1,[1])}, \dots, \tilde{\mathbf{X}}_{0:K}^{(N,[1])}$ . Then, in a backward pass that starts at the end of the optimization horizon, a new sequence of improved control law parameters  $(\mathbf{K}_{0:K-1}^{[1]}, \mathbf{L}_{0:K-1}^{[1]})$  with cost  $\mathcal{J}^{[1]} \leq \mathcal{J}^{[0]}$  is obtained by minimizing the cost-to-go (5.53) at each stage. In the next iteration,  $(\mathbf{K}_{0:K-1}^{[1]}, \mathbf{L}_{0:K-1}^{[1]})$  is then used in the forward pass to compute a new reference trajectory  $\tilde{\mathbf{X}}_{0:K}^{(0,[2])}, \tilde{\mathbf{X}}_{0:K}^{(1,[2])}, \dots, \tilde{\mathbf{X}}_{0:K}^{(N,[2])}$ ,

which in turn leads to new control law parameters  $(\mathbf{K}_{0:K-1}^{[2]}, \mathbf{L}_{0:K-1}^{[2]})$  with cost  $\mathcal{J}^{[2]} \leq \mathcal{J}^{[1]}$  in a subsequent backward pass. This process is repeated until convergence of the cost, i.e., until we have  $\mathcal{J}^{[c-1]} - \mathcal{J}^{[c]} \leq \epsilon$  for some small  $\epsilon$ , say  $\epsilon = 10^{-8}$ , after the backward pass of iteration  $c$ .

In the literature, such iterative methods are often used to compute control laws for finite or receding horizon control of MJLS. Specifically for state feedback and static output feedback control, where the controller need not estimate the plant state, provably convergent algorithms have been introduced [148, 177, 180, 200]. However, a straightforward implementation of the iterative method sketched above does not result in a convergent algorithm because the cost-to-go (5.53) is not guaranteed to be convex with respect to  $(\mathbf{K}_t, \mathbf{L}_t)$ . This observation is in line with a conclusion drawn in [198] where an algorithm for the computation of linear control laws for finite horizon control of MJLS without mode observation was presented. To illustrate the non-convexity, consider the dynamics (5.37) with  $\underline{\xi}_k, \underline{U}_k \in \mathbb{R}$ ,  $N = 1$ ,  $L = 0$ , and parameters<sup>8</sup>

$$\begin{aligned} \overline{\mathbf{A}}_k^{(0)} &= 1.1, & \overline{\mathbf{A}}_k^{(1)} &= 3.1, & \overline{\mathbf{B}}_k^{(0)} &= 1.1, & \overline{\mathbf{B}}_k^{(1)} &= 7, \\ \overline{\mathbf{C}} &= 2, & \overline{\mathbf{W}} &= 0.1, & \overline{\mathbf{V}} &= 0.1, & \mathbf{T}_k &= \begin{bmatrix} 0.4 & 0.6 \\ 0.8 & 0.2 \end{bmatrix}. \end{aligned}$$

Furthermore, we let  $P[\gamma_{k|k} = 1 | \mathcal{I}_k] = 1$  for all  $k$ , i.e., every measurement is processed without delay, and the parameters of the cost function (5.52) are

$$\hat{\mathbf{Q}}_K = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{\mathbf{Q}}_t^{(0)} = \hat{\mathbf{Q}}_t^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix},$$

with  $K = 3$  and  $t = 0, 1, 2$ . For the initial conditions

$$\underline{\mu}_0 = \begin{bmatrix} 0.2 \\ 0.8 \end{bmatrix}, \quad \tilde{\mathbf{X}}_0^{(0)} = \begin{bmatrix} 0.26 & 0.01 \\ 0.01 & 0.01 \end{bmatrix}, \quad \tilde{\mathbf{X}}_0^{(1)} = \begin{bmatrix} 2.75 & 2.25 \\ 2.25 & 2.25 \end{bmatrix},$$

and  $\mathbf{L}_0 \in [-3, 3]$  and  $\mathbf{K}_0 \in [-3, 3]$ , we obtain the cost-to-go  $\mathcal{V}_0$  plotted in Fig. 5.1. The figure illustrates that the cost-to-go  $\mathcal{V}_0$  is non-convex with respect to the control law parameters  $(\mathbf{K}_0, \mathbf{L}_0)$ .

To obtain a convergent algorithm, we demand that, at each stage  $t$  of the horizon,  $\tilde{\mathbf{X}}_t^{(i)}$  and  $\tilde{\mathbf{P}}_t^{(i)}$  have a certain structure. This is similar to what was done [198] and restricts their possible trajectories. This restriction, then, yields an upper bound  $\bar{\mathcal{V}}_t$  of the cost-to-go (5.53). In contrast to (5.53),  $\bar{\mathcal{V}}_t$  is convex with respect to  $(\mathbf{K}_t, \mathbf{L}_t)$ , which enables us to prove that the iterative method discussed above indeed converges.

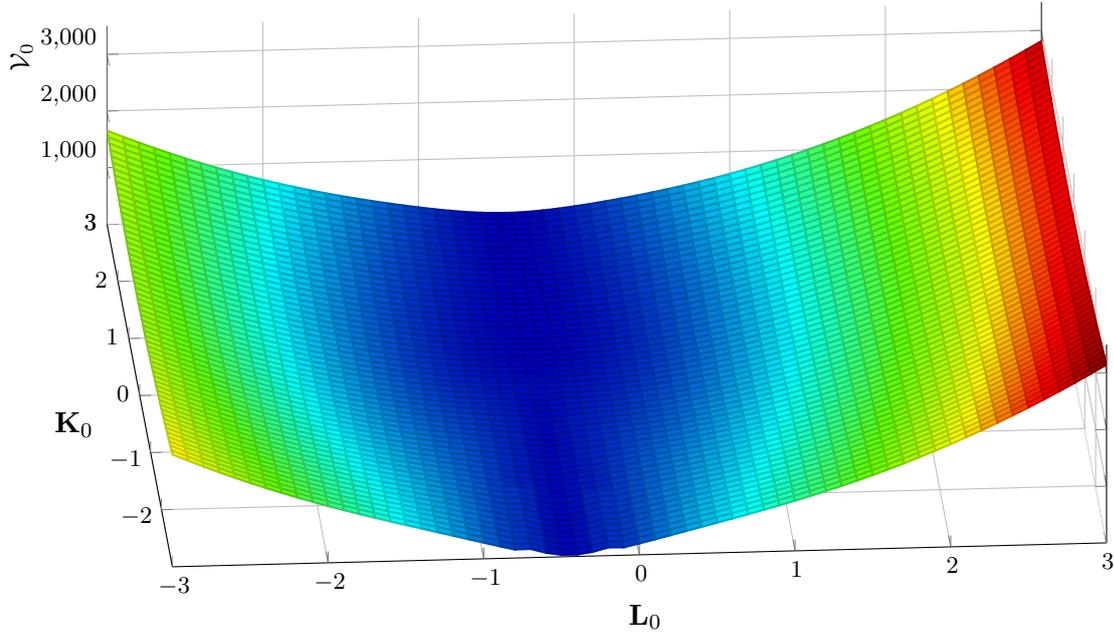
To start, we note that the closed-loop second moment  $\tilde{\mathbf{X}}_t$  can be partitioned as

$$\tilde{\mathbf{X}}_t = \mathbb{E}\{\tilde{\mathbf{x}}_t \tilde{\mathbf{x}}_t^T | \mathcal{I}_0\} = \begin{bmatrix} \mathbb{E}\{\underline{\xi}_t \underline{\xi}_t^T | \mathcal{I}_0\} & \mathbb{E}\{\underline{\xi}_t \hat{\xi}_t^T | \mathcal{I}_0\} \\ \mathbb{E}\{\hat{\xi}_t \underline{\xi}_t^T | \mathcal{I}_0\} & \mathbb{E}\{\hat{\xi}_t \hat{\xi}_t^T | \mathcal{I}_0\} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{X}}_{t,1} & \tilde{\mathbf{X}}_{t,12} \\ \tilde{\mathbf{X}}_{t,12}^T & \tilde{\mathbf{X}}_{t,2} \end{bmatrix}, \quad (5.56)$$

using the definition of the closed-loop state (5.45). By definition,  $\tilde{\mathbf{X}}_t$  is at least positive semidefinite,  $\tilde{\mathbf{X}}_t \succcurlyeq 0$ , so that it must hold (cf. Theorem A.7 in Appendix A.4)

$$\begin{aligned} \tilde{\mathbf{X}}_{t,2} &\succcurlyeq 0, \\ \tilde{\mathbf{X}}_{t,12} &= \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,2}, \\ \tilde{\mathbf{X}}_{t,1} &\succcurlyeq \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,12}^T. \end{aligned} \quad (5.57)$$

<sup>8</sup>The MJLS considered here does not represent an augmented dynamical system as introduced in Section 5.4.1. Instead, we use an ordinary MJLS with two modes, scalar state and control input for illustration purposes.



**Figure 5.1:** Illustration of the non-convex cost-to-go  $\mathcal{V}_0$  at stage  $t = 0$  for different values of  $\mathbf{L}_0$  and  $\mathbf{K}_0$ .

Hence, there exist  $\bar{\mathbf{X}}_t \succcurlyeq 0$  such that we may write [201]

$$\tilde{\mathbf{X}}_t = \begin{bmatrix} \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,12}^\top + \bar{\mathbf{X}}_t & \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,2} \\ \tilde{\mathbf{X}}_{t,2} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,12}^\top & \tilde{\mathbf{X}}_{t,2} \end{bmatrix}. \quad (5.58)$$

The matrix  $\bar{\mathbf{X}}_t$  appearing in (5.58) is the estimation error covariance of an unbiased estimator of  $\underline{\xi}_t$ . To see this, we write  $\underline{\xi}_t$  as

$$\underline{\xi}_t = \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \hat{\underline{\xi}}_t + \underline{\xi}_t - \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \hat{\underline{\xi}}_t, \quad (5.59)$$

and interpret  $\tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \hat{\underline{\xi}}_t$  as an unbiased estimator of  $\underline{\xi}_t$  with zero-mean estimation error

$$\tilde{\underline{e}}_t = \underline{\xi}_t - \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \hat{\underline{\xi}}_t. \quad (5.60)$$

The estimation error and the estimator are uncorrelated since

$$\begin{aligned} \mathbb{E}\left\{\tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \hat{\underline{\xi}}_t \tilde{\underline{e}}_t^\top \mid \mathcal{I}_0\right\} &= \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \mathbb{E}\left\{\hat{\underline{\xi}}_t \underline{\xi}_t^\top \mid \mathcal{I}_0\right\} - \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \mathbb{E}\left\{\hat{\underline{\xi}}_t \hat{\underline{\xi}}_t^\top \mid \mathcal{I}_0\right\} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,12}^\top \\ &= \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,12}^\top - \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,2} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,12}^\top \\ &= \mathbf{0}, \end{aligned}$$

where we used that  $\tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,2} \tilde{\mathbf{X}}_{t,2}^\dagger = \tilde{\mathbf{X}}_{t,2}^\dagger$  (cf. (A.13)). Thus, by the orthogonality principle,  $\tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \hat{\underline{\xi}}_t$  is the linear MMSE estimator. Using (5.60), the estimation error covariance matrix is given by

$$\begin{aligned} \mathbb{E}\{\tilde{\underline{e}}_t \tilde{\underline{e}}_t^\top \mid \mathcal{I}_0\} &= \tilde{\mathbf{X}}_{t,1} + \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,2} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,12}^\top - \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,12}^\top - \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,12}^\top \\ &= \tilde{\mathbf{X}}_{t,1} + \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,12}^\top - 2\tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,12}^\top \\ &= \tilde{\mathbf{X}}_{t,1} - \tilde{\mathbf{X}}_{t,12} \tilde{\mathbf{X}}_{t,2}^\dagger \tilde{\mathbf{X}}_{t,12}^\top \\ &= \bar{\mathbf{X}}_t. \end{aligned} \quad (5.61)$$

Thus, the controller state  $\hat{\underline{\xi}}_t$  is generally *not* an unbiased estimate of the augmented plant state  $\underline{\xi}_t$ . However, the parametrization (5.58) of the closed-loop second moment remains valid for any  $\tilde{\mathbf{X}}_{t,12}$  that fulfills (5.57). In particular, we may demand that

$$\tilde{\mathbf{X}}_{t,12} \stackrel{!}{=} \tilde{\mathbf{X}}_{t,2}, \quad (5.62)$$

so that (5.59) becomes

$$\underline{\xi}_t = \hat{\underline{\xi}}_t + \tilde{\underline{e}}_t,$$

with

$$\tilde{\underline{e}}_t = \underline{\xi}_t - \hat{\underline{\xi}}_t,$$

rendering the controller state the linear MMSE and, hence, unbiased estimate of  $\underline{\xi}_t$ . The error covariance matrix is then given by

$$\begin{aligned} \mathbb{E}\{\tilde{\underline{e}}_t \tilde{\underline{e}}_t^T \mid \mathcal{I}_0\} &= \mathbb{E}\left\{ \left( \underline{\xi}_t - \hat{\underline{\xi}}_t \right) \left( \underline{\xi}_t - \hat{\underline{\xi}}_t \right)^T \mid \mathcal{I}_0 \right\} \\ &= \tilde{\mathbf{X}}_{t,1} + \tilde{\mathbf{X}}_{t,2} - \tilde{\mathbf{X}}_{t,12} - \tilde{\mathbf{X}}_{t,12}^T \\ &= \tilde{\mathbf{X}}_{t,1} - \tilde{\mathbf{X}}_{t,2}. \end{aligned}$$

But, on the other hand, we have  $\mathbb{E}\{\tilde{\underline{e}}_t \tilde{\underline{e}}_t^T \mid \mathcal{I}_0\} = \overline{\mathbf{X}}_t$  by (5.61), so that we can conclude

$$\tilde{\mathbf{X}}_{t,1} = \underline{\mathbf{X}}_t + \overline{\mathbf{X}}_t, \quad (5.63)$$

where we let  $\underline{\mathbf{X}}_t = \tilde{\mathbf{X}}_{t,2}$ .

Thus, the second moment (5.56) becomes

$$\tilde{\mathbf{X}}_t = \begin{bmatrix} \underline{\mathbf{X}}_t + \overline{\mathbf{X}}_t & \underline{\mathbf{X}}_t \\ \underline{\mathbf{X}}_t & \underline{\mathbf{X}}_t \end{bmatrix}. \quad (5.64)$$

Eq. (5.64) allows us to express the second moment of the augmented plant state,  $\tilde{\mathbf{X}}_{t,1}$ , as the sum of the second moment of the controller state,  $\underline{\mathbf{X}}_t$ , and the estimation error covariance  $\overline{\mathbf{X}}_t$ . We again emphasize that (5.64) implies that the controller state  $\hat{\underline{\xi}}_t$  is the linear MMSE estimate of  $\underline{\xi}_t$  at each stage of the optimization horizon. Note that this also demands that the controller state  $\hat{\underline{\xi}}_t$  and the estimation error  $\underline{\xi}_t - \hat{\underline{\xi}}_t$  are uncorrelated. This property, for instance, also holds for the Kalman filter when used to estimate the state of a linear system. Asymptotic uncorrelatedness between estimator and estimation error has also been shown to be a necessary condition for optimality in many infinite horizon control problems with randomly varying parameters [202–205]. Similarly, a closed-loop second moment of the form (5.64) is necessary for a linear control law to be optimal in infinite horizon networked control scenarios without ACKs [124, 206].

With the aid of (5.47), we get the desired parametrization of  $\tilde{\mathbf{X}}_t^{(i)}$  from (5.62), (5.63), and (5.64) according to

$$\tilde{\mathbf{X}}_t^{(i)} = \begin{bmatrix} \underline{\mathbf{X}}_t^{(i)} + \overline{\mathbf{X}}_t^{(i)} & \underline{\mathbf{X}}_t^{(i)} \\ \underline{\mathbf{X}}_t^{(i)} & \underline{\mathbf{X}}_t^{(i)} \end{bmatrix}, \quad (5.65)$$

with the identifications

$$\begin{aligned} \underline{\mathbf{X}}_t^{(i)} &= \tilde{\mathbf{X}}_{t,2}^{(i)} = \tilde{\mathbf{X}}_{t,12}^{(i)}, \\ \overline{\mathbf{X}}_t^{(i)} &= \tilde{\mathbf{X}}_{t,1}^{(i)} + \tilde{\mathbf{X}}_{t,2}^{(i)} - \tilde{\mathbf{X}}_{t,12}^{(i)} - \left( \tilde{\mathbf{X}}_{t,12}^{(i)} \right)^T = \tilde{\mathbf{X}}_{t,1}^{(i)} - \underline{\mathbf{X}}_t^{(i)}, \end{aligned} \quad (5.66)$$

for  $t = 0, 1, \dots, K$  and  $i = 0, 1, \dots, N$ . The dynamics of  $\underline{\mathbf{X}}_t^{(i)}$  and  $\overline{\mathbf{X}}_t^{(i)}$  are provided by the following lemma.

**Lemma 5.3:**

Let  $\tilde{\mathbf{X}}_t^{(i)}$  be parameterized according to (5.65) and let (5.66) hold. Then, the dynamics of  $\underline{\mathbf{X}}_t^{(i)}$  and  $\overline{\mathbf{X}}_t^{(i)}$  are given by

$$\begin{aligned}\underline{\mathbf{X}}_{t+1}^{(j)} &= \sum_{i=0}^N t_{ij} \left[ \mu_t^{(i)} \mathbf{K}_t \hat{\mathbf{S}}_t \bar{\mathbf{V}} \left( \mathbf{K}_t \hat{\mathbf{S}}_t \right)^{\mathbf{T}} \right. \\ &\quad \left. + \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t \right) \underline{\mathbf{X}}_t^{(i)} \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t \right)^{\mathbf{T}} + \mathbf{K}_t \hat{\mathbf{S}}_t \bar{\mathbf{C}} \overline{\mathbf{X}}_t^{(i)} \left( \mathbf{K}_t \hat{\mathbf{S}}_t \bar{\mathbf{C}} \right)^{\mathbf{T}} \right], \\ \overline{\mathbf{X}}_{t+1}^{(j)} &= \sum_{i=0}^N t_{ij} \left[ \mu_t^{(i)} \left( \bar{\mathbf{W}} + \mathbf{K}_t \hat{\mathbf{S}}_t \bar{\mathbf{V}} \left( \mathbf{K}_t \hat{\mathbf{S}}_t \right)^{\mathbf{T}} \right) \right. \\ &\quad \left. + \left( \bar{\mathbf{A}}_t^{(i)} - \hat{\mathbf{A}}_t + \left( \bar{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t \right) \mathbf{L}_t \right) \overline{\mathbf{X}}_t^{(i)} \left( \bar{\mathbf{A}}_t^{(i)} - \hat{\mathbf{A}}_t + \left( \bar{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t \right) \mathbf{L}_t \right)^{\mathbf{T}} \right. \\ &\quad \left. + \left( \bar{\mathbf{A}}_t^{(i)} - \mathbf{K}_t \hat{\mathbf{S}}_t \bar{\mathbf{C}} \right) \overline{\mathbf{X}}_t^{(i)} \left( \bar{\mathbf{A}}_t^{(i)} - \mathbf{K}_t \hat{\mathbf{S}}_t \bar{\mathbf{C}} \right)^{\mathbf{T}} \right],\end{aligned}\tag{5.67}$$

for  $t = 0, 1, \dots, K$  and  $j = 0, 1, \dots, N$ .

*Proof.* The proof of this result is given in Appendix C.4.  $\square$

To obtain an appropriate parametrization of  $\tilde{\mathbf{P}}_t^{(i)}$ , we recall from (5.53) that  $\tilde{\mathbf{P}}_t^{(i)}$  is the costate matrix associated with the second moment  $\tilde{\mathbf{X}}_t^{(i)}$ . Hence, we can regard it as the second moment of the costate  $\tilde{\lambda}_t = \left[ \lambda_t^{\mathbf{T}} \hat{\lambda}_t^{\mathbf{T}} \right]^{\mathbf{T}}$  associated with the closed-loop state  $\tilde{x}_t$ . With this interpretation, we have

$$\tilde{\mathbf{P}}_t = \mathbb{E} \left\{ \tilde{\lambda}_t \tilde{\lambda}_t^{\mathbf{T}} \mid \mathcal{I}_0 \right\} = \begin{bmatrix} \tilde{\mathbf{P}}_{t,1} & \tilde{\mathbf{P}}_{t,12} \\ \tilde{\mathbf{P}}_{t,12}^{\mathbf{T}} & \tilde{\mathbf{P}}_{t,2} \end{bmatrix} = \sum_{i=0}^N \tilde{\mathbf{P}}_t^{(i)},\tag{5.68}$$

with

$$\tilde{\mathbf{P}}_t^{(i)} = \begin{bmatrix} \mathbb{E} \left\{ \lambda_t \lambda_t^{\mathbf{T}} \mathbf{1}_{\{\theta_t=i\}} \mid \mathcal{I}_0 \right\} & \mathbb{E} \left\{ \lambda_t \hat{\lambda}_t^{\mathbf{T}} \mathbf{1}_{\{\theta_t=i\}} \mid \mathcal{I}_0 \right\} \\ \mathbb{E} \left\{ \hat{\lambda}_t \lambda_t^{\mathbf{T}} \mathbf{1}_{\{\theta_t=i\}} \mid \mathcal{I}_0 \right\} & \mathbb{E} \left\{ \hat{\lambda}_t \hat{\lambda}_t^{\mathbf{T}} \mathbf{1}_{\{\theta_t=i\}} \mid \mathcal{I}_0 \right\} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{P}}_{t,1}^{(i)} & \tilde{\mathbf{P}}_{t,12}^{(i)} \\ \left( \tilde{\mathbf{P}}_{t,12}^{(i)} \right)^{\mathbf{T}} & \tilde{\mathbf{P}}_{t,2}^{(i)} \end{bmatrix}.$$

Similar to what we did above, we now demand that  $-\hat{\lambda}_t$  be the linear MMSE estimate of the costate  $\lambda_t$  that is associated with the augmented plant state. The corresponding error covariance matrix is then

$$\bar{\mathbf{P}}_t = \mathbb{E} \left\{ \left( \lambda_t + \hat{\lambda}_t \right) \left( \lambda_t + \hat{\lambda}_t \right)^{\mathbf{T}} \mid \mathcal{I}_0 \right\} = \tilde{\mathbf{P}}_{t,1} + \tilde{\mathbf{P}}_{t,2} + \tilde{\mathbf{P}}_{t,12} + \tilde{\mathbf{P}}_{t,12}^{\mathbf{T}}.$$

Uncorrelatedness of estimator and estimation error requires that

$$\tilde{\mathbf{P}}_{t,12} \stackrel{!}{=} -\tilde{\mathbf{P}}_{t,2} = \mathbf{P}_t,$$

and hence

$$\begin{aligned}\bar{\mathbf{P}}_t &= \tilde{\mathbf{P}}_{t,1} + \tilde{\mathbf{P}}_{t,2} + \tilde{\mathbf{P}}_{t,12} + \tilde{\mathbf{P}}_{t,12}^{\mathbf{T}} \\ &= \tilde{\mathbf{P}}_{t,1} - \mathbf{P}_t.\end{aligned}$$

With this parametrization,  $\tilde{\mathbf{P}}_t$  reads

$$\tilde{\mathbf{P}}_t = \begin{bmatrix} \mathbf{P}_t + \bar{\mathbf{P}}_t & -\mathbf{P}_t \\ -\mathbf{P}_t & \mathbf{P}_t \end{bmatrix}.\tag{5.69}$$

Similar to (5.64), we can express the second moment of the costate associated with the augmented plant state as the sum of the second moment of the costate associated with the controller state and the estimation error covariance. Again, a costate matrix of the form (5.69) is known to be a necessary condition for optimal control laws in infinite horizon control problems with randomly varying parameters and networked control scenarios without ACKs [124, 202–206].

By (5.68), we can use (5.69) to parameterize  $\tilde{\mathbf{P}}_t^{(i)}$  according to

$$\tilde{\mathbf{P}}_t^{(i)} = \begin{bmatrix} \underline{\mathbf{P}}_t^{(i)} + \overline{\mathbf{P}}_t^{(i)} & -\underline{\mathbf{P}}_t^{(i)} \\ -\underline{\mathbf{P}}_t^{(i)} & \underline{\mathbf{P}}_t^{(i)} \end{bmatrix}, \quad (5.70)$$

with the identities

$$\begin{aligned} \underline{\mathbf{P}}_t^{(i)} &= \tilde{\mathbf{P}}_{t,2}^{(i)} = -\tilde{\mathbf{P}}_{t,12}^{(i)}, \\ \overline{\mathbf{P}}_t^{(i)} &= \tilde{\mathbf{P}}_{t,1}^{(i)} + \tilde{\mathbf{P}}_{t,2}^{(i)} + \tilde{\mathbf{P}}_{t,12}^{(i)} + \left(\tilde{\mathbf{P}}_{t,12}^{(i)}\right)^{\text{T}} = \tilde{\mathbf{P}}_{t,1}^{(i)} - \underline{\mathbf{P}}_t^{(i)}, \end{aligned} \quad (5.71)$$

for  $t = 0, 1, \dots, K$  and  $i = 0, 1, \dots, N$ .

### Theorem 5.2:

Fix a sequence of control law parameters  $(\mathbf{K}_0, \mathbf{L}_0), \dots, (\mathbf{K}_{K-1}, \mathbf{L}_{K-1})$  and let  $\tilde{\mathbf{X}}_t^{(i)}$  and  $\tilde{\mathbf{P}}_t^{(i)}$  be parameterized as per (5.65) and (5.70), respectively. Further, let denote

$$\begin{aligned} \underline{\mathcal{P}}_t &= \left(\underline{\mathbf{P}}_t^{(0)}, \underline{\mathbf{P}}_t^{(1)}, \dots, \underline{\mathbf{P}}_t^{(N)}\right) \in \mathbb{H}^{n_\xi}, \\ \overline{\mathcal{P}}_t &= \left(\overline{\mathbf{P}}_t^{(0)}, \overline{\mathbf{P}}_t^{(1)}, \dots, \overline{\mathbf{P}}_t^{(N)}\right) \in \mathbb{H}^{n_\xi}, \\ \overline{\omega}_t &= \left(\overline{\omega}_t^{(0)}, \overline{\omega}_t^{(1)}, \dots, \overline{\omega}_t^{(N)}\right) \in \mathbb{H}^1. \end{aligned}$$

Then, the cost-to-go at each stage  $t = 0, 1, \dots, K$  is given by

$$\overline{V}_t = \sum_{i=0}^N \text{tr} \left[ \underline{\mathbf{P}}_t^{(i)} \overline{\mathbf{X}}_t^{(i)} + \overline{\mathbf{P}}_t^{(i)} \left( \underline{\mathbf{X}}_t^{(i)} + \overline{\mathbf{X}}_t^{(i)} \right) \right] + \mu_t^{(i)} \overline{\omega}_t^{(i)}, \quad (5.72)$$

with  $\underline{\mathbf{X}}_t^{(i)}$  and  $\overline{\mathbf{X}}_t^{(i)}$  given by Lemma 5.3, and where  $\underline{\mathbf{P}}_t^{(i)}$ ,  $\overline{\mathbf{P}}_t^{(i)}$ , and  $\overline{\omega}_t^{(i)}$  are computed by means of the backward recursions

$$\begin{aligned} \underline{\mathbf{P}}_t^{(i)} &= \left(\mathbf{J}^{(i)} \mathbf{L}_t\right)^{\text{T}} \mathbf{R}_t \mathbf{J}^{(i)} \mathbf{L}_t + \left(\overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t\right)^{\text{T}} \mathcal{E}^{(i)}(\overline{\mathcal{P}}_{t+1}) \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \\ &\quad + \left(\hat{\mathbf{A}}_t + \left(\hat{\mathbf{B}}_t - \overline{\mathbf{B}}_t^{(i)}\right) \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}}\right)^{\text{T}} \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}) \left(\hat{\mathbf{A}}_t + \left(\hat{\mathbf{B}}_t - \overline{\mathbf{B}}_t^{(i)}\right) \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}}\right), \\ \overline{\mathbf{P}}_t^{(i)} &= \overline{\mathbf{Q}}_t^{(i)} + \left(\mathbf{J}^{(i)} \mathbf{L}_t\right)^{\text{T}} \mathbf{R}_t \mathbf{J}^{(i)} \mathbf{L}_t + \left(\overline{\mathbf{A}}_t^{(i)} + \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t\right)^{\text{T}} \mathcal{E}^{(i)}(\overline{\mathcal{P}}_{t+1}) \left(\overline{\mathbf{A}}_t^{(i)} + \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t\right) \\ &\quad + \left(\overline{\mathbf{A}}_t^{(i)} - \hat{\mathbf{A}}_t + \left(\overline{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t\right) \mathbf{L}_t\right)^{\text{T}} \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}) \left(\overline{\mathbf{A}}_t^{(i)} - \hat{\mathbf{A}}_t + \left(\overline{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t\right) \mathbf{L}_t\right), \\ \overline{\omega}_t^{(i)} &= \mathcal{E}^{(i)}(\overline{\omega}_{t+1}) + \text{tr} \left[ \mathcal{E}^{(i)}(\overline{\mathcal{P}}_{t+1} + \underline{\mathcal{P}}_{t+1}) \overline{\mathbf{W}} + \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}) \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{V}} \left(\mathbf{K}_t \hat{\mathbf{S}}_t\right)^{\text{T}} \right], \end{aligned} \quad (5.73)$$

that are initialized with  $\underline{\mathbf{P}}_K^{(i)} = \mathbf{0}_{n_\xi}$ ,  $\overline{\mathbf{P}}_K^{(i)} = \overline{\mathbf{Q}}_K$ , and  $\overline{\omega}_K^{(i)} = 0$  for  $i = 0, 1, \dots, N$ .

*Proof.* The proof is given in Appendix C.5. □

Observe that we denote the cost-to-go in Theorem 5.2 by  $\bar{\mathcal{V}}_t$  and not by  $\mathcal{V}_t$  as we did in Theorem 5.1. As already indicated above, the parametrizations (5.65) and (5.70) of  $\tilde{\mathbf{X}}_t^{(i)}$  and  $\tilde{\mathbf{P}}_t^{(i)}$  restrict their possible trajectories. For any sequence of control law parameters  $(\mathbf{K}_{0:K-1}, \mathbf{L}_{0:K-1})$ , the costate matrices  $\tilde{\mathbf{P}}_t^{(i)}$  represent the contribution of the control law parameters  $(\mathbf{K}_{t:K-1}, \mathbf{L}_{t:K-1})$  to  $\bar{\mathcal{V}}_t$ . Thus, restricting their trajectories means that  $\bar{\mathcal{V}}_t \geq \mathcal{V}_t$ , i.e.,  $\bar{\mathcal{V}}_t$  bounds the cost-to-go from above, and in particular,  $\bar{\mathcal{V}}_0 \geq \mathcal{V}_0 = \mathcal{J}(\mathbf{K}_{0:K-1}, \mathbf{L}_{0:K-1})$ . Note also, that a minimizer  $(\mathbf{K}_t^*, \mathbf{L}_t^*)$  of (5.72) is, in general, not a minimizer of (5.53). Hence, any sequence of control law parameters  $(\mathbf{K}_{0:K-1}^*, \mathbf{L}_{0:K-1}^*)$  that is obtained by minimizing  $\bar{\mathcal{V}}_t$  for each stage  $t$ , is, in general, only suboptimal with regards to the cost (5.52).

Using (5.73) reveals that (5.72) is given by

$$\begin{aligned} \bar{\mathcal{V}}_t = & \sum_{i=0}^N \text{tr} \left[ \mathbf{K}_t^T \mathcal{E}^{(i)}(\mathcal{P}_{t+1}) \mathbf{K}_t \mathbf{D}_{t,1}^{(i)} \right] + \text{tr} \left[ \mathbf{L}_t^T \mathbf{D}_{t,2}^{(i)} \mathbf{L}_t \mathbf{X}_t^{(i)} \right] \\ & + 2 \text{tr} \left[ \left( \mathbf{D}_{t,3}^{(i)} \right)^T \mathbf{K}_t \right] + 2 \text{tr} \left[ \left( \mathbf{D}_{t,4}^{(i)} \right)^T \mathbf{L}_t \right] + c_t^{(i)}, \end{aligned} \quad (5.74)$$

with

$$\begin{aligned} \mathbf{D}_{t,1}^{(i)} &= \hat{\mathbf{S}}_t \left( \mu_t^{(i)} \bar{\mathbf{V}} + \bar{\mathbf{C}} \bar{\mathbf{X}}_t^{(i)} \bar{\mathbf{C}}^T \right) \hat{\mathbf{S}}_t^T, \\ \mathbf{D}_{t,2}^{(i)} &= \left( \mathbf{J}^{(i)} \right)^T \mathbf{R}_t \mathbf{J}^{(i)} + \left( \bar{\mathbf{B}}_t^{(i)} \right)^T \mathcal{E}^{(i)}(\bar{\mathcal{P}}_{t+1}) \bar{\mathbf{B}}_t^{(i)} + \left( \bar{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t \right)^T \mathcal{E}^{(i)}(\mathcal{P}_{t+1}) \left( \bar{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t \right), \\ \mathbf{D}_{t,3}^{(i)} &= -\mathcal{E}^{(i)}(\mathcal{P}_{t+1}) \bar{\mathbf{A}}_t^{(i)} \bar{\mathbf{X}}_t^{(i)} \left( \hat{\mathbf{S}}_t \bar{\mathbf{C}} \right)^T, \\ \mathbf{D}_{t,4}^{(i)} &= \left( \left( \bar{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t \right)^T \mathcal{E}^{(i)}(\mathcal{P}_{t+1}) \left( \bar{\mathbf{A}}_t^{(i)} - \hat{\mathbf{A}}_t \right) + \left( \bar{\mathbf{B}}_t^{(i)} \right)^T \mathcal{E}^{(i)}(\bar{\mathcal{P}}_{t+1}) \bar{\mathbf{A}}_t^{(i)} \right) \mathbf{X}_t^{(i)}, \end{aligned}$$

and where  $c_t^{(i)} \geq 0$  contains only terms independent of  $\mathbf{K}_t$  and  $\mathbf{L}_t$ . Eq. (5.74) implies that  $\bar{\mathcal{V}}_t$  is convex with regards to  $\mathbf{K}_t$  and  $\mathbf{L}_t$ , as the following lemma shows.

**Lemma 5.4:**

$\bar{\mathcal{V}}_t$  is convex with regards to  $(\mathbf{K}_t, \mathbf{L}_t)$ .

*Proof.* The proof is given in Appendix C.6. □

For the example from the beginning of this section, the resulting convexified cost-to-go  $\bar{\mathcal{V}}_0$  is depicted in Fig. 5.2. Comparing the figure with Fig. 5.1 also illustrates what we mentioned above:  $\bar{\mathcal{V}}_0$  constitutes an upper bound of the cost.

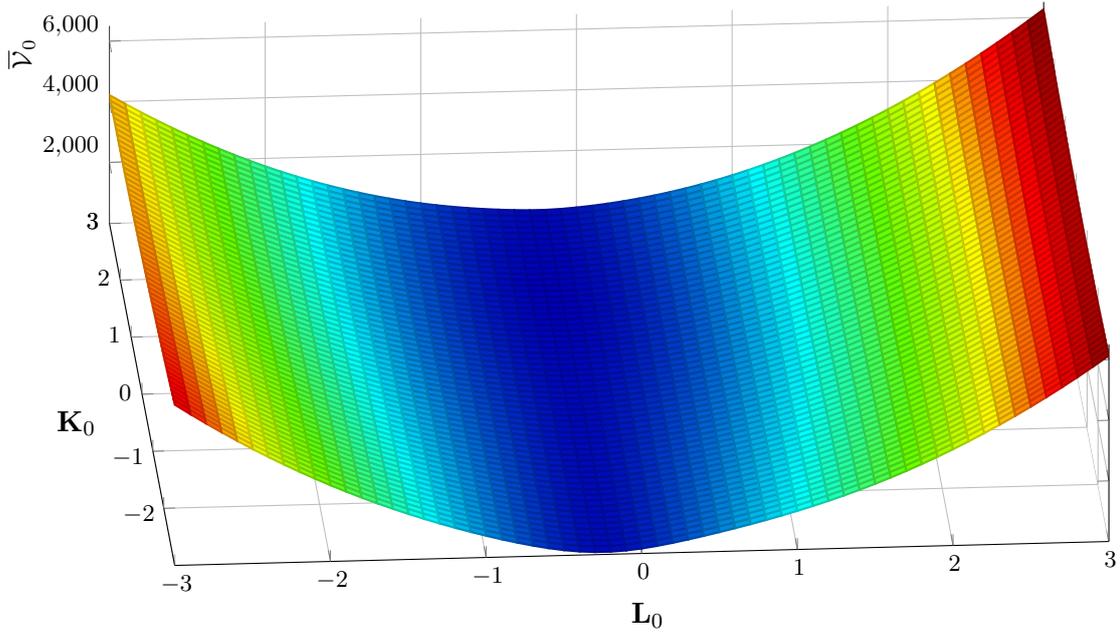
Lemma 5.4 implies that any minimizer of  $\bar{\mathcal{V}}_t$  is globally optimal. Setting the derivatives of (5.74) with respect to  $\mathbf{K}_t$  and  $\mathbf{L}_t$  to zero yields the necessary and sufficient minimality conditions

$$\sum_{i=0}^N \mathcal{E}^{(i)}(\mathcal{P}_{t+1}) \mathbf{K}_t \mathbf{D}_{t,1}^{(i)} + \mathbf{D}_{t,3}^{(i)} \stackrel{!}{=} \mathbf{0}_{n_\xi \times (L+1)n_y}, \quad (5.75)$$

$$\sum_{i=0}^N \mathbf{D}_{t,2}^{(i)} \mathbf{L}_t \mathbf{X}_t^{(i)} + \mathbf{D}_{t,4}^{(i)} \stackrel{!}{=} \mathbf{0}_{Nn_u \times n_\xi}, \quad (5.76)$$

for  $t = 0, 1, \dots, K-1$ , where we used the identities [207]

$$\frac{\text{dtr}[\mathbf{A}\mathbf{X}]}{\text{d}\mathbf{X}} = \mathbf{A}^T, \quad \frac{\text{dtr}[\mathbf{X}^T \mathbf{A} \mathbf{X} \mathbf{B}]}{\text{d}\mathbf{X}} = \mathbf{A} \mathbf{X} \mathbf{B} + \mathbf{A}^T \mathbf{X} \mathbf{B}^T.$$



**Figure 5.2:** Illustration of the convexified cost-to-go  $\bar{V}_0$  at stage  $t = 0$  for different values of  $\mathbf{L}_0$  and  $\mathbf{K}_0$ .

Eqs. (5.75) and (5.76) are *generalized Sylvester equations* that arise frequently in control theory [208–211]. Although being linear with respect to  $\mathbf{K}_t$  and  $\mathbf{L}_t$ , respectively, (5.75) and (5.76) cannot be solved directly. They are, however, equivalent to a system of linear equations due to the relationship between the Kronecker product and vectorization of a matrix (cf. Appendix A.3).

**Corollary 5.1:**

A necessary and sufficient condition for  $(\mathbf{K}_t^*, \mathbf{L}_t^*)$  to minimize  $\bar{V}_t$  is

$$\begin{aligned} \left( \sum_{i=0}^N \mathbf{D}_{t,1}^{(i)} \otimes \mathcal{E}^{(i)}(\mathcal{P}_{t+1}) \right) \text{vec}(\mathbf{K}_t^*) + \sum_{i=0}^N \text{vec}(\mathbf{D}_{t,3}^{(i)}) &= \mathbf{0}_{n_\xi(L+1)n_y}, \\ \left( \sum_{i=0}^N \mathbf{X}_t^{(i)} \otimes \mathbf{D}_{t,2}^{(i)} \right) \text{vec}(\mathbf{L}_t^*) + \sum_{i=0}^N \text{vec}(\mathbf{D}_{t,4}^{(i)}) &= \mathbf{0}_{Nn_u n_\xi}, \end{aligned} \quad (5.77)$$

for  $t = 0, 1, \dots, K - 1$ , where  $\otimes$  denotes the Kronecker product (cf. (A.4)) and  $\text{vec}(\cdot)$  vectorization of a matrix (cf. (A.10)).

*Proof.* The result follows directly from the application of (A.11) to (5.75) and (5.76).  $\square$

As in the systems of linear equations that appeared in Section 5.3, the matrices on the left side of (5.77) are in general rank deficient, implying the existence of infinitely many solutions. Again, one particularly attractive solution is the minimum norm solution

$$\text{vec}(\mathbf{K}_t^+) = - \left( \sum_{i=0}^N \mathbf{D}_{t,1}^{(i)} \otimes \mathcal{E}^{(i)}(\mathcal{P}_{t+1}) \right)^\dagger \left( \sum_{i=0}^N \text{vec}(\mathbf{D}_{t,3}^{(i)}) \right), \quad (5.78)$$

$$\text{vec}(\mathbf{L}_t^+) = - \left( \sum_{i=0}^N \mathbf{X}_t^{(i)} \otimes \mathbf{D}_{t,2}^{(i)} \right)^\dagger \left( \sum_{i=0}^N \text{vec}(\mathbf{D}_{t,4}^{(i)}) \right). \quad (5.79)$$

---

**Algorithm 5.4** Iterative Procedure for the Computation of the Control Laws
 

---

**Input:**  $\underline{\mathbf{X}}_0^{(i)}, \overline{\mathbf{X}}_0^{(i)}, \mu_0^{(i)}, \gamma_0$   
**Output:**  $(\mathbf{K}_0, \mathbf{L}_0), \dots, (\mathbf{K}_{K-1}, \mathbf{L}_{K-1})$

- 1: **for**  $t = 0$  **to**  $K - 1$  **do**
- 2:     Compute  $\hat{\mathbf{A}}_t$  and  $\hat{\mathbf{B}}_t$  using  $\mu_t^{(i)}$  in (5.41) and (5.42)
- 3:     Compute  $\hat{\mathbf{S}}_t$  using  $\gamma_0$ , (5.38), and (5.43)
- 4:     **for**  $i = 0$  **to**  $N$  **do**
- 5:          $\mu_{t+1}^{(i)} = \sum_j t_{ji} \mu_t^{(j)}$  // Mode probability prediction
- 6:     **end for**
- 7: **end for**
- 8: Initialize the iteration counter  $c = 0$  and set  $\overline{\mathcal{V}}_0^{[0]} = \infty$
- 9: Choose initial sequence of control law parameters  $(\mathbf{K}_0^{[0]}, \mathbf{L}_0^{[0]}), \dots, (\mathbf{K}_{K-1}^{[0]}, \mathbf{L}_{K-1}^{[0]})$
- 10: **repeat**
- 11:      $c = c + 1$   
       // Forward pass: Update reference trajectory
- 12:     Set  $\underline{\mathbf{X}}_0^{(i,[c])} = \underline{\mathbf{X}}_0^{(i)}$  and  $\overline{\mathbf{X}}_0^{(i,[c])} = \overline{\mathbf{X}}_0^{(i)}$
- 13:     **for**  $t = 0$  **to**  $K - 1$  **do**
- 14:         **for**  $j = 0$  **to**  $N$  **do**
- 15:             Compute  $\underline{\mathbf{X}}_{t+1}^{(j,[c])}$  and  $\overline{\mathbf{X}}_{t+1}^{(j,[c])}$  using  $(\mathbf{K}_t^{[c-1]}, \mathbf{L}_t^{[c-1]})$  in (5.67) // Lemma 5.3
- 16:         **end for**
- 17:     **end for**  
       // Backward pass: Update control laws
- 18:     Set  $\underline{\mathbf{P}}_t^{(i,[c])} = \mathbf{0}_{n_\xi}$ ,  $\overline{\mathbf{P}}_K^{(i,[c])} = \overline{\mathbf{Q}}_K$ , and  $\overline{\omega}_K^{(i,[c])} = 0$
- 19:     **for**  $t = K - 1$  **to**  $0$  **do**
- 20:         Evaluate (5.78) and (5.79) to obtain minimizing control law parameters  $(\mathbf{K}_t^{[c]}, \mathbf{L}_t^{[c]})$
- 21:         **for**  $i = 0$  **to**  $N$  **do**
- 22:             Compute  $\underline{\mathbf{P}}_t^{(i,[c])}$ ,  $\overline{\mathbf{P}}_t^{(i,[c])}$ , and  $\overline{\omega}_t^{(i,[c])}$  using  $(\mathbf{K}_t^{[c]}, \mathbf{L}_t^{[c]})$  in (5.73) // Theorem 5.2
- 23:         **end for**
- 24:     **end for**  
       // Evaluate cost improvement
- 25:     Evaluate either (5.72) or (5.74) to obtain  $\overline{\mathcal{V}}_0^{[c]}$
- 26: **until**  $\overline{\mathcal{V}}_0^{[c-1]} - \overline{\mathcal{V}}_0^{[c]} \leq \epsilon$
- 27: **return** Sequence of control law parameters  $(\mathbf{K}_0^{[c]}, \mathbf{L}_0^{[c]}), \dots, (\mathbf{K}_{K-1}^{[c]}, \mathbf{L}_{K-1}^{[c]})$

---

Based on the convexified cost-to-go  $\overline{\mathcal{V}}_t$ , we present an iterative procedure for the computation of the control law parameters  $(\mathbf{K}_0, \mathbf{L}_0), \dots, (\mathbf{K}_{K-1}, \mathbf{L}_{K-1})$  in Algorithm 5.4. The algorithm is straightforward to implement since Corollary 5.1 enables us to determine minimizers of  $\overline{\mathcal{V}}_t$  directly without the need for numerical optimization. In each iteration  $c$ , a sequence of control law parameters  $(\mathbf{K}_0^{[c]}, \mathbf{L}_0^{[c]}), \dots, (\mathbf{K}_{K-1}^{[c]}, \mathbf{L}_{K-1}^{[c]})$  is computed that leads to an improvement of the cost, i.e., it holds  $\overline{\mathcal{V}}_0^{[c]} \leq \overline{\mathcal{V}}_0^{[c-1]}$ . Convergence of the proposed iterative procedure is guaranteed by the following theorem.

**Theorem 5.3:**

*Let  $(\mathbf{K}_0^{[c]}, \mathbf{L}_0^{[c]}), \dots, (\mathbf{K}_{K-1}^{[c]}, \mathbf{L}_{K-1}^{[c]})$  be the control law parameters at the end of iteration  $c$  in Algorithm 5.4, i.e., after the completion of the backward pass of the algorithm. Then, it holds  $\overline{\mathcal{V}}_0^{[c]} \leq \overline{\mathcal{V}}_0^{[c-1]}$  and, moreover,  $(\mathbf{K}_0, \mathbf{L}_0), \dots, (\mathbf{K}_{K-1}, \mathbf{L}_{K-1}) = \lim_{c \rightarrow \infty} (\mathbf{K}_0^{[c]}, \mathbf{L}_0^{[c]}), \dots, (\mathbf{K}_{K-1}^{[c]}, \mathbf{L}_{K-1}^{[c]})$  exists.*

*Proof.* The proof is given in Appendix C.7. □

**Algorithm 5.5** Conceptual Algorithm for the Proposed SMPC

---

```

1: Set initial conditions  $\hat{\underline{\xi}}_0$ ,  $\underline{\mathbf{X}}_0^{(i)}$ ,  $\overline{\mathbf{X}}_0^{(i)}$ , and  $\underline{\mu}_0$  according to (5.80) and (5.81)
2: for  $k = 0, 1 \dots$  do
3:   Get newest mode realization  $\theta_{k-r'} = j$  from  $\mathcal{A}_k$  // As described in Section 4.1
4:    $\underline{\mu}_{k-r'} = \underline{e}_{N+1}^{(j+1)}$  // Update  $\underline{\mu}_{k-r'}$  according to (5.31)
5:   for  $r = r' - 1$  to 0 do
6:     for  $i = 0$  to  $N$  do
7:        $\mu_{k-r}^{(i)} = \sum_j t_{ji} \mu_{k-(r+1)}^{(j)}$  // Update of previous mode probability
8:     end for
9:   end for
10:  // Process measurements
11:  Construct  $\underline{y}_k$  from  $\mathcal{Y}_k$  and determine the corresponding mode  $\underline{\gamma}_k$  according to (5.32) and (5.35)
12:  // Compute control law parameters
13:  Call Algorithm 5.4 with  $\underline{\mathbf{X}}_k^{(i)}$ ,  $\overline{\mathbf{X}}_k^{(i)}$ ,  $\mu_k^{(i)}$ , and  $\underline{\gamma}_k$  to obtain  $(\mathbf{K}_0, \mathbf{L}_0)$ 
14:  Compute  $\underline{U}_k = \mathbf{L}_0 \hat{\underline{\xi}}_k$  according to (5.39)
15:  Transmit  $\underline{U}_k$  to the actuator
16:  // Propagate state estimate
17:  Predict  $\hat{\underline{\xi}}_{k+1}$  using  $\hat{\underline{\xi}}_k$ ,  $\underline{y}_k$ , and  $\mathbf{K}_0$  in (5.39)
18:  for  $j = 0$  to  $N$  do
19:    Compute  $\underline{\mathbf{X}}_{k+1}^{(j)}$  and  $\overline{\mathbf{X}}_{k+1}^{(j)}$  using  $(\mathbf{K}_0, \mathbf{L}_0)$  in (5.67)
20:  end for
21: end for

```

---

**5.4.4 Summary of the Proposed SMPC Algorithm**

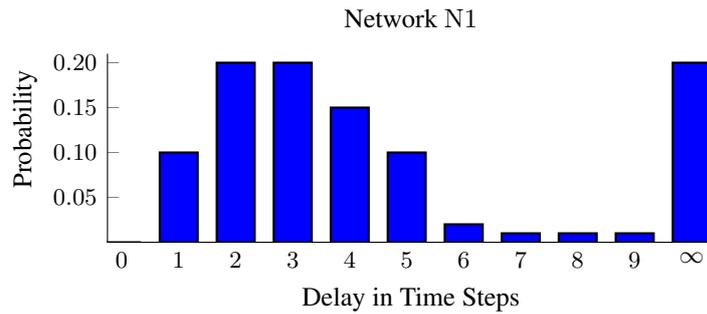
The results of the previous sections are the basis for the proposed SMPC approach based on a local approximation of the cost-to-go. We present the final conceptual algorithm in Algorithm 5.5. The controller state is initialized with

$$\hat{\underline{\xi}}_0 = \begin{bmatrix} \mathbf{1}_{L+1} \otimes \hat{\underline{x}}_0 \\ \underline{\mathbf{0}} \end{bmatrix}, \quad (5.80)$$

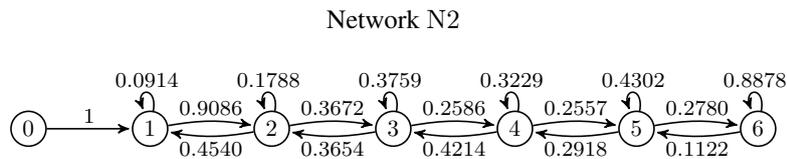
i.e., we stack the mean  $\hat{\underline{x}}_0$  of the initial plant state  $L + 1$  times and set  $\underline{\eta}_0 = \underline{\mathbf{0}}$  to reflect that the buffer at the actuator side is initially empty. Accordingly, we set  $\underline{\mu}_0 = \underline{e}_{N+1}^{(N+1)}$ . Estimation error covariance and second moment of the controller state are set to

$$\begin{aligned} \overline{\mathbf{X}}_0^{(i)} &= \mathbf{0}, & \underline{\mathbf{X}}_0^{(i)} &= \mathbf{0}, \\ \overline{\mathbf{X}}_0^{(N)} &= \begin{bmatrix} \mathbf{1}_{L+1} \otimes \mathbf{X}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, & \underline{\mathbf{X}}_0^{(N)} &= \hat{\underline{\xi}}_0 \hat{\underline{\xi}}_0^T + \overline{\mathbf{X}}_0^{(N)}, \end{aligned} \quad (5.81)$$

for  $i = 0, 1, \dots, N - 1$ , with  $\mathbf{1}_{L+1} \in \mathbb{R}^{L+1 \times L+1}$  the matrix of ones, and where we recall  $\mathbf{X}_0$  as the covariance matrix of the initial plant state  $\underline{x}_0$ . To construct the augmented measurement  $\underline{y}_k$  from  $\mathcal{Y}_k$  in line 10, recall from (5.33) that we can safely set the element  $\tilde{y}_{k|k-l}$  to  $\underline{\mathbf{0}}$ , if the measurement  $y_{k-l}$  is not available. At  $k = 0$ , when Algorithm 5.4 is called for the first time, the parameters  $(\mathbf{K}, \mathbf{L})$  of the infinite horizon control law presented in [124] can be used as the initial sequence of control law parameters  $(\mathbf{K}_0^{[0]}, \mathbf{L}_0^{[0]}), \dots, (\mathbf{K}_{K-1}^{[0]}, \mathbf{L}_{K-1}^{[0]})$ . This control law, however, need not always exist. Alternatively, the initial control law parameters can be chosen at random. Then, at the subsequent time steps  $k = 1, 2, \dots$ , the control law parameters from the previous time step can serve as the initial guess. We note that we can replace  $t_{ji}$  in line 7 by  $t_{k-r,j,i}$  if  $\theta_k$  is a time-inhomogeneous Markov chain. We also note that the computational complexity of the algorithm is higher than that of Algorithm 5.2 proposed in Section 5.3. In particular, its computational demand per time step depends on the number of iterations needed until convergence of the cost and is thus not fixed. One option to circumvent this is



(a) Probability distribution governing the packet delays  $\tau_k^{ca}$ ,  $\tau_k^{sc}$ , and  $\tau_k^{ac}$  in network N1. The notation  $\infty$  indicates a packet loss, treated as infinite delay.



(b) Markov Chain governing the packet delays  $\tau_k^{ca}$ ,  $\tau_k^{sc}$ , and  $\tau_k^{ac}$  in network N2. State 6 indicates a packet loss.

**Figure 5.3:** Characteristics of the networks N1 and N2 considered in the evaluation.

to trade-off computation time against solution quality by carrying out only a small number of iterations instead of iterating until convergence.

To conclude this section, we highlight that Algorithm 5.5 is a closed-loop feedback control approach because the availability of future measurements is anticipated during the computation of the control laws. In doing so, the impact of the dual effect on the cost is taken into account. Specifically in control tasks, where the dual effect substantially contributes to the cost, it is reasonable to expect that Algorithm 5.5 achieves a better control performance than the open-loop feedback approach (Algorithm 5.3) from Section 5.3. The simulation results we provide in the next section show that this is indeed the case.

## 5.5 Evaluation

In this section, we assess the performance of the proposed approaches by means of a simulation. To that end, we consider two setups with different networks N1 and N2. In the first network, N1, the packet delay and loss probabilities are independent over time as introduced in Section 3.2.1. For simplicity, we assume that they occur with fixed probabilities according to the probability distribution  $p$  given by (F.1) in Appendix F.1 that is depicted in Fig. 5.3a. In the second network, N2, the packet delays and losses are Markovian as introduced in Section 3.2.2. The Markov chain is shown in Fig. 5.3b and the corresponding transition matrix is given by (F.2) in Appendix F.1.

Our goal is to evaluate the performance of our approaches in two different control tasks. The first task is the control of a double integrator plant and the second one is the control of an overhead crane maneuvering a load. In each task, we compare the two proposed SMPC algorithms with the open-loop feedback control approaches presented in [152] and [169].<sup>9</sup> Rather than using the augmented dynamics (4.1), these approaches use the model (4.26) and interpret the actual plant input  $\underline{u}_{k+t}$  at each stage of the optimization horizon as a random variable with Dirac mixture PDF  $f_{k+t}^u$ . The key idea of

<sup>9</sup>The earlier approach presented in [212] uses similar ideas, but its applicability is, however, restricted to plants with discrete-valued inputs.

the approach from [152] is to propagate a given state estimate  $\hat{\underline{x}}_k = \mathbb{E}\{\underline{x}_k \mid \mathcal{I}_k\}$  over the optimization horizon according to

$$\hat{\underline{x}}_{k+t+1} = \mathbf{A}\hat{\underline{x}}_{k+t} + \mathbf{B}\hat{\underline{u}}_{k+t}, \quad (5.82)$$

with  $\hat{\underline{u}}_{k+t} = \mathbb{E}\{\underline{u}_{k+t} \mid \mathcal{I}_k\}$  the expected input at stage  $t$ , which is calculated based on the weighting factors  $\omega_{k+t}^{(i)}$  of the Dirac mixture  $f_{k+t}^u$ . The expected inputs depend on  $\underline{u}_{k|k}, \underline{u}_{k+1|k}, \dots, \underline{u}_{k+N-1|k}$ , which form the control sequence  $\underline{U}_k$ , and in [152] it was proposed to compute them based on a pre-defined controller gain  $\mathbf{L}$  by virtue of the propagated state estimate

$$\underline{u}_{k+t|k} = \mathbf{L}\hat{\underline{x}}_{k+t}.$$

However, here, we choose  $\underline{u}_{k|k}, \underline{u}_{k+1|k}, \dots, \underline{u}_{k+N-1|k}$  as the minimizers of an approximation of the cost function (5.2) given by

$$\tilde{\mathcal{J}} = \hat{\underline{x}}_{k+K}^T \mathbf{Q}_K \hat{\underline{x}}_{k+K} + \sum_{t=0}^{K-1} \hat{\underline{x}}_{k+t}^T \mathbf{Q}_t \hat{\underline{x}}_{k+t} + \hat{\underline{u}}_{k+t}^T \mathbf{R}_t \hat{\underline{u}}_{k+t}, \quad (5.83)$$

which can be obtained by numerical optimization. To provide the required state estimate, we use the IMM filter presented in Section 5.3.2.

Instead of solely considering expected inputs, the approach presented in [169] enumerates all possible realizations of the random variables  $\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+K-1}$  so as to determine all possible evolutions of a given the state estimate  $\hat{\underline{x}}_k$  over the optimization horizon. This leads to a tree-like structure, where each path from the root node to a leaf, referred to as scenario, corresponds to one possible sequence of applied inputs. Based on this tree, an approximation of the cost function (5.2) is then computed as the weighted sum of the cost of all scenarios

$$\tilde{\mathcal{J}} = \sum_j s_k^{(j)} \mathcal{J}^{(j)}, \quad (5.84)$$

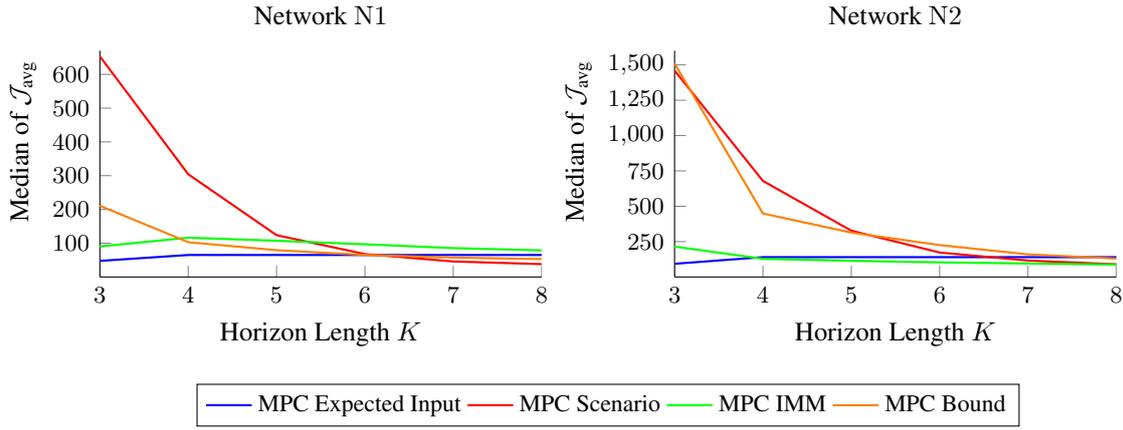
with  $\mathcal{J}^{(j)}$  the cost of the  $j$ -the scenario, which is given by (5.83) evaluated with the predicted states and the inputs belonging to the scenario, and where the weighting factor  $s_k^{(j)}$  denotes the probability that the inputs are actually applied. This probability depends on the packet delay and loss probabilities and is calculated based on the weighting factors  $\omega_{k+t}^{(i)}$  of the Dirac mixtures  $f_{k+t}^u$  given by (4.27). Finally, the input sequence  $\underline{U}_k$  is composed of the inputs  $\underline{u}_{k|k}, \underline{u}_{k+1|k}, \dots, \underline{u}_{k+N-1|k}$  that minimize the approximated cost (5.84), which need to be determined numerically. To provide the required state estimate, we again use the IMM filter presented in Section 5.3.2.

We present the evaluation results in the following two sections. To facilitate the presentation, we will refer to the approach from [152] as *MPC Expected Input* and to the approach from [169] as *MPC Scenario*. Similarly, we will use the terms *MPC IMM* and *MPC Bound* for the SMPC approaches we derived in Sections 5.3 and 5.4.

### 5.5.1 Control of a Double Integrator

The first control task we consider is the canonical example of a double integrator plant, which is used to model the horizontal movement of a mass due to a force applied to it. Denoting by  $q$  the horizontal position (in meters) of a body with mass  $m = 2$  kg, the continuous-time plant dynamics is of the form (3.2) and given by

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} w(t),$$



**Figure 5.4:** Control of the double integrator: Medians of the average incurred cost  $\mathcal{J}_{\text{avg}}$  in the simulations with network N1 (left) and network N2 (right).

with state  $\underline{x}(t) = [q(t) \dot{q}(t)]^T$ , control input  $u(t)$  the applied force (in newtons), and where  $w(t)$  is a Gaussian, zero-mean white disturbance (in newtons) with intensity  $\mathbf{W}_c = 0.001 \text{ N}^2$ . Discretization with the sampling interval  $t_a = 0.4 \text{ s}$  during which the control input and the disturbance are assumed constant yields the corresponding discrete-time dynamics (3.1) with  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{W}$  computed according to (3.3). Additionally, we set

$$\mathbf{C} = [1 \ 0], \quad \mathbf{V} = 0.04,$$

i.e., a sensor device provides noisy measurements of  $q$ . Since we focus on the impact of the packet delays and losses, both  $\mathbf{W}$  and  $\mathbf{V}$  are deliberately chosen small so as to keep the impact of the noise on the control performance low.

The weighting matrices in the cost function (5.2) are

$$\mathbf{Q}_t = \mathbf{Q} = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}, \quad \mathbf{R}_t = \mathbf{R} = 1, \quad (5.85)$$

for  $t = 0, 1, \dots, K-1$ , and  $\mathbf{Q}_K$  is chosen as the unique stabilizing solution  $\mathbf{X}$  of the associated discrete-time algebraic Riccati equation

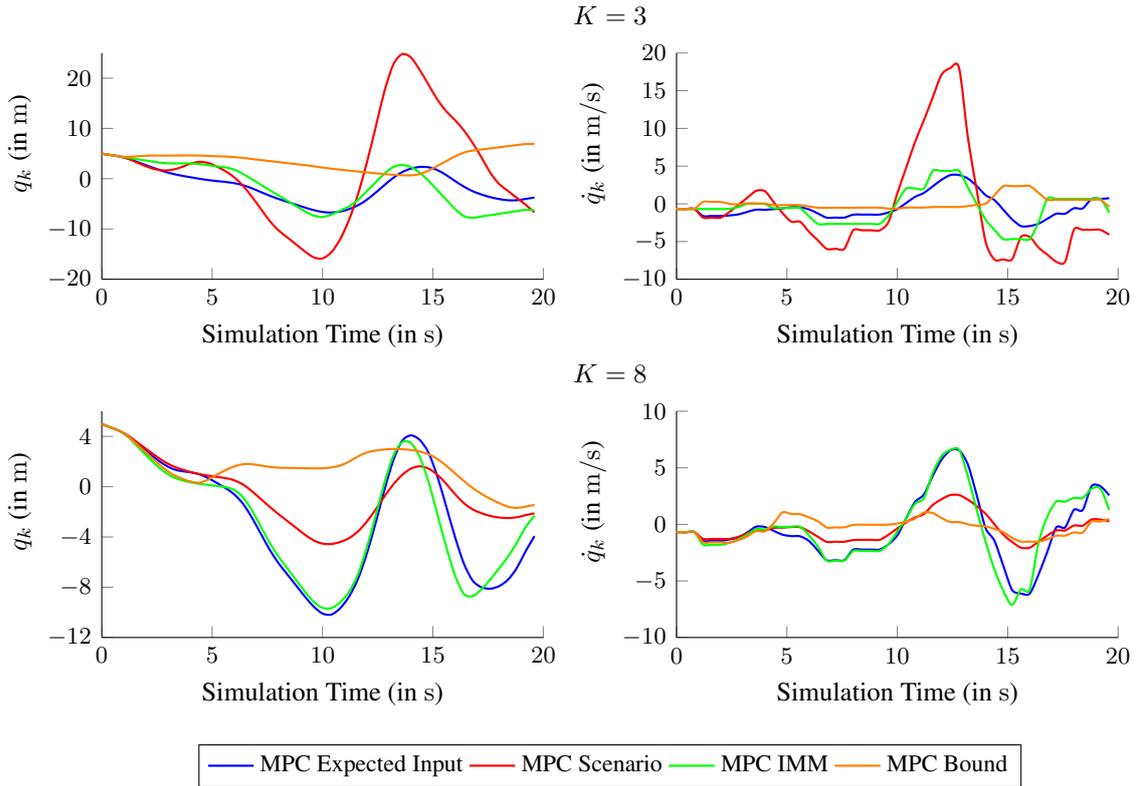
$$\mathbf{X} = \mathbf{A}^T \mathbf{X} \mathbf{A} - \mathbf{A}^T \mathbf{X} \mathbf{B} (\mathbf{B}^T \mathbf{X} \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^T \mathbf{X} \mathbf{A} + \mathbf{Q}, \quad (5.86)$$

resulting in the matrix given by (F.3) in Appendix F.1.

For the comparison, we conduct several Monte Carlo simulations in each setup, where we increase the horizon length from  $K = 3$  to  $K = 8$ . In each simulation, we carry out  $N_s = 1000$  simulation runs. In each run, the plant is simulated over  $t_s = 20 \text{ s}$ , i.e., over  $K_s = 50$  time steps, and the initial plant state  $\underline{x}_0$  is randomly drawn from a Gaussian distribution with mean and covariance

$$\hat{\underline{x}}_0 = \begin{bmatrix} 5 \\ 0 \end{bmatrix}, \quad \mathbf{X}_0 = 0.25 \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}.$$

In the first setup with network N1, the realizations  $\tau_k^{\text{ca}}, \tau_k^{\text{sc}}, \tau_k^{\text{ac}}$  are independently drawn according to the probability distribution shown in Fig. 5.3a. We set  $N = 4$  and choose to discard measurements that experience a delay of more than five time steps so that Assumption 5.1 is met with  $L = 5$ . In the second setup with network N2,  $\tau_k^{\text{ca}}, \tau_k^{\text{sc}}, \tau_k^{\text{ac}}$  are determined according to the Markov chain depicted in Fig. 5.3b, where state 6 indicates a packet loss. The initial conditions  $\tau_0^{\text{ca}}, \tau_0^{\text{sc}}, \tau_0^{\text{ac}}$  are uniformly drawn at random prior to each run. For the computation of the lumped dynamics for  $\theta_k$  according



**Figure 5.5:** Control of the double integrator: Evolution of the plant states  $q_k$  (left) and  $\dot{q}_k$  (right) in exemplary runs with  $K = 3$  (top row) and  $K = 8$  (bottom row) and network N1.

to (4.24), we replace all zero entries in the transition matrix (F.2) by  $10^{-15}$  and then re-normalize the rows to meet Assumption 3.5. Moreover, we again choose  $N = 4$  and  $L = 5$ .

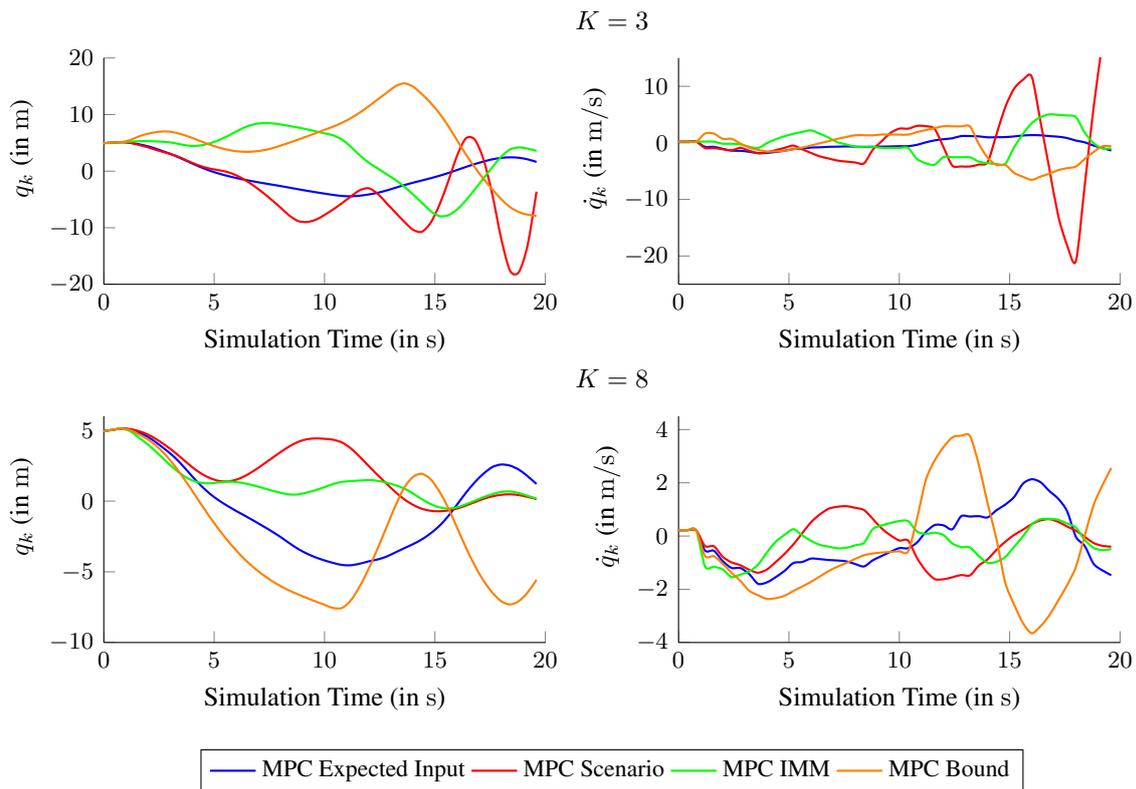
To measure the performance of the controllers, we compute the average incurred cost  $\mathcal{J}_{\text{avg}}$  in each run according to

$$\mathcal{J}_{\text{avg}} = \frac{1}{K_s} \left[ \underline{x}_{K_s}^T \mathbf{Q} \underline{x}_{K_s} + \sum_{k=0}^{K_s-1} \underline{x}_k^T \mathbf{Q} \underline{x}_k + u_k^2 \mathbf{R} \right]. \quad (5.87)$$

For both setups, the resulting medians of  $\mathcal{J}_{\text{avg}}$  are plotted in Fig. 5.4, and exemplary state trajectories for  $K = 3$  and  $K = 8$  are depicted in Figs. 5.5 and 5.6, respectively.

For the *MPC Scenario*, the *MPC IMM*, and the *MPC Bound* the results indicate an improvement of the control performance as the optimization horizon increases. This is an expected outcome because longer optimization horizons permit better predictions of the impact of the occurring packet delays and losses on the plant state, which in turn improve the controllers' decision-making. On the other hand, no performance gain can be observed for the *MPC Expected Input* for horizon lengths  $K \geq N = 4$ . This results from the fact that the expected inputs  $\hat{u}_{k+t}$  at stages  $t = N + 1, \dots, K - 1$  coincide with  $\underline{u}_{k+t|k}$ , i.e., the control inputs to be computed for these stages, so that the impact of occurring packet delays and losses is no longer included in the propagation of the state estimates (5.82). The *MPC Scenario* and the *MPC Bound* perform significantly worse for  $K = 3$ , but exhibit sharp performance increases, so that, ultimately, their inferiority vanishes. In the simulations with network N1 they even outperform the other approaches for horizon lengths  $K \geq 7$ .

The simulation results also reveal that in this control task neither of our proposed approaches can be deemed superior to the other. While the *MPC Bound* achieves better performance than the *MPC*



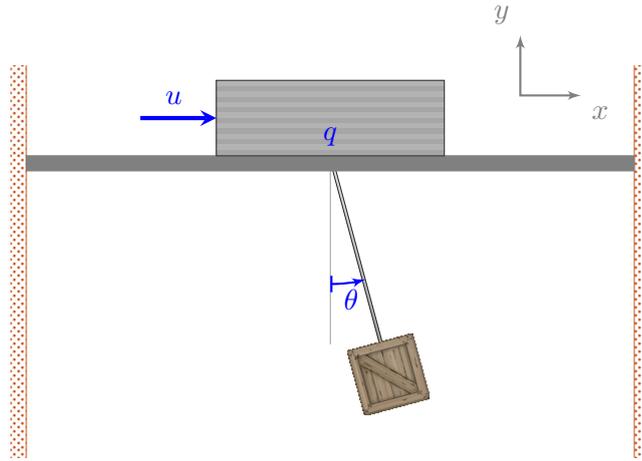
**Figure 5.6:** Control of the double integrator: Evolution of the plant states  $q_k$  (left) and  $\dot{q}_k$  (right) in exemplary runs with  $K = 3$  (top row) and  $K = 8$  (bottom row) and network N2.

*IMM* in the simulations with network N1 for all considered horizon lengths but  $K = 3$ , we observe the opposite in the simulations with network N2. Here, the *MPC IMM* outperforms the *MPC Bound* for all horizon lengths. At first glance, this observation seems to be contrary to the closed-loop feedback property of the *MPC Bound* that renders it advantageous from a theoretical perspective. However, in this control task, the impact of the dual effect is negligibly small, so that the assumptions and approximations we had to make in the course of Section 5.4 to consider its contribution to the cost do not pay off. Instead, they render the derived bound  $\bar{V}_t$  of the cost-to-go, and consequently the resulting control laws, too conservative, in particular for short optimization horizons.

We conclude from the results that the control performance achieved by our proposed SMPC approaches is comparable to that of state-of-the-art approaches from the literature when the impact of the dual effect on the cost is small enough to be neglected. The results presented in the next section show that both the *MPC IMM* and the *MPC Bound* achieve a markedly better performance than the *MPC Expected Input* and the *MPC Scenario* in more challenging control tasks, where the contribution of the dual effect to the cost is more significant.

### 5.5.2 Control of an Overhead Crane

The second task we consider is to control an overhead crane that maneuvers a load as sketched in Fig. 5.7. Let  $q$  and  $\theta$  denote the horizontal position of the trolley (in meters) and the deviation of the load (in radians) from the vertical, chosen such that positive values correspond to the counter-clockwise direction. Then, with the parameters listed in Table 5.1 the continuous-time nonlinear dynamics can be



**Figure 5.7:** Sketch of an overhead crane that maneuvers a load.

**Table 5.1:** Parameters of the overhead crane used in the simulation.

Parameter	Symbol	Value
Mass of trolley	$m_t$	1 kg
Mass of load	$m_l$	2 kg
Length of wire rope	$l$	0.5 m
Gravitational acceleration	$g$	9.81 m/s <sup>2</sup>
Damping coefficient of trolley	$b_t$	0.5 N s/m
Damping coefficient of load	$b_l$	0.1 N s/m

written as [213]

$$\dot{\underline{x}}(t) = \begin{bmatrix} \dot{q}(t) \\ \dot{\theta}(t) \\ \mathbf{M}(\theta(t))^{-1} \underline{f}(\underline{x}(t), u(t), w(t)) \end{bmatrix},$$

where  $\underline{x}(t) = [q(t) \theta(t) \dot{q}(t) \dot{\theta}(t)]^T$  is the state,  $u(t)$  is the force (in newtons) applied to move the trolley along the bridge,  $w(t)$  is an external disturbance (in newtons) acting on both the trolley and the load, and where  $\mathbf{M}$  and  $\underline{f}$  are given by<sup>10</sup>

$$\mathbf{M}(\theta) = \begin{bmatrix} \cos(\theta) & l \\ m_t + m_l & m_l l \cos(\theta) \end{bmatrix},$$

$$\underline{f}(\underline{x}, u, w) = \begin{bmatrix} -g \sin(\theta) - \frac{l}{m_l} b_l \dot{\theta} + \frac{w}{m_l} \cos(\theta) \\ u - b_t \dot{q} - b_l (\dot{q} + l \dot{\theta} \cos(\theta)) + m_l l \dot{\theta}^2 \sin(\theta) + w \end{bmatrix}.$$

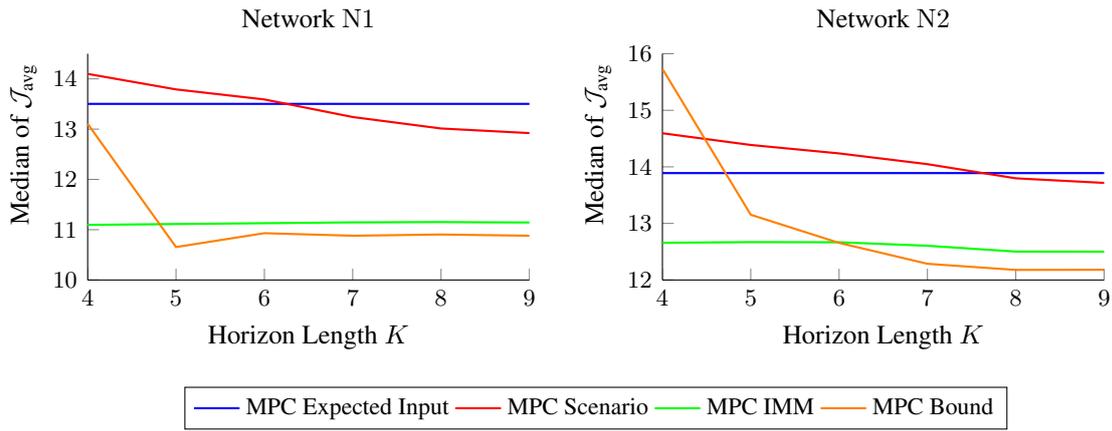
Linearization around the equilibrium  $\underline{x}_e(t) = \underline{0}$  yields a linear dynamics of the form (3.2)

$$\dot{\underline{x}}(t) = \mathbf{A}_c \underline{x}(t) + \mathbf{B}_c u(t) + \mathbf{G}_c w(t),$$

with  $\mathbf{A}_c$ ,  $\mathbf{B}_c$ , and  $\mathbf{G}_c$  as per (F.4) in Appendix F.1.

Assuming that  $w(t)$  is a Gaussian, zero-mean white disturbance with intensity  $\mathbf{W}_c = 0.001 \text{ N}^2$  and setting the sampling interval to  $t_a = 0.1 \text{ s}$  yields the discrete-time dynamics (3.1) with  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{W}$

<sup>10</sup>For better readability, we omit the dependence of  $\underline{x}$ ,  $u$ , and  $w$  on the time  $t$ .



**Figure 5.8:** Control of an overhead crane: Medians of the average incurred cost  $\mathcal{J}_{\text{avg}}$  in the simulations with network N1 (left) and network N2 (right).

computed according to (3.3). Furthermore, we set

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{V} = 0.04\mathbf{I}_2,$$

that is, a sensor takes noisy measurements of the trolley position and the deviation of the load from the vertical. Again,  $\mathbf{W}$  and  $\mathbf{V}$  are chosen small so that the impact of the noise on the control performance is kept low.

We choose the weighting matrices in the cost function (5.2) as

$$\mathbf{Q}_t = \mathbf{Q} = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix}, \quad \mathbf{R}_t = \mathbf{R} = 1, \quad (5.88)$$

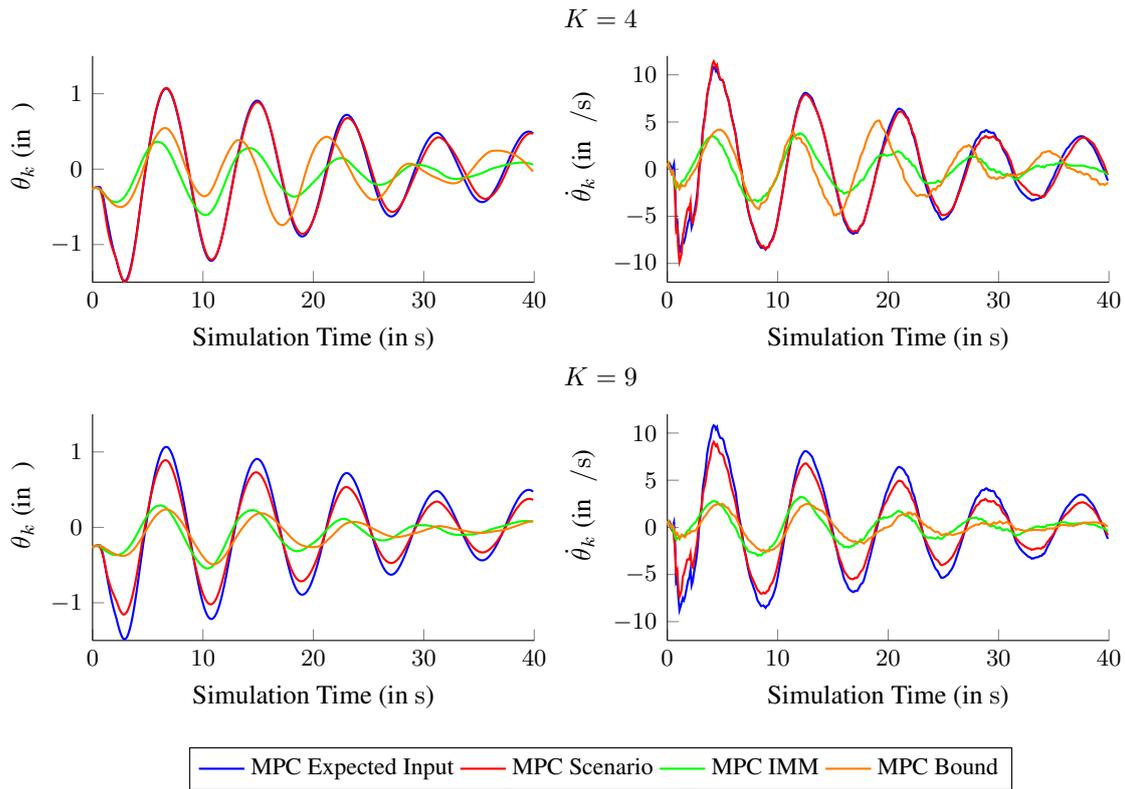
for  $t = 0, 1, \dots, K-1$ , and set  $\mathbf{Q}_K$  to the matrix given by (F.5), which is the unique stabilizing solution  $\mathbf{X}$  of the discrete-time algebraic Riccati equation (5.86).

For the comparison, we conduct several Monte Carlo simulations with both networks, where we increase the horizon length from  $K = 4$  to  $K = 9$ . In each simulation, we carry out  $N_s = 500$  simulation runs. In each run, the plant is simulated over  $t_s = 40$  s, i.e., over  $K_s = 400$  time steps, and the initial plant state  $\underline{x}_0$  is randomly drawn from a Gaussian distribution with mean and covariance

$$\hat{\underline{x}}_0 = \begin{bmatrix} -0.5 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{X}_0 = 1 \cdot 10^{-4}\mathbf{I}_4.$$

In the simulation runs, we again let  $N = 4$ ,  $L = 5$ , and the realizations  $\tau_k^{\text{ca}}$ ,  $\tau_k^{\text{sc}}$ ,  $\tau_k^{\text{ac}}$  are determined as described in Section 5.5.1. To measure the performance of the controllers, we compute the average incurred cost  $\mathcal{J}_{\text{avg}}$  in each run according to (5.87), of which the resulting medians are plotted in Fig. 5.8.

The results show that both proposed approaches perform significantly better than the *MPC Expected Input* and the *MPC Scenario* for all considered optimization horizons in the simulations with network



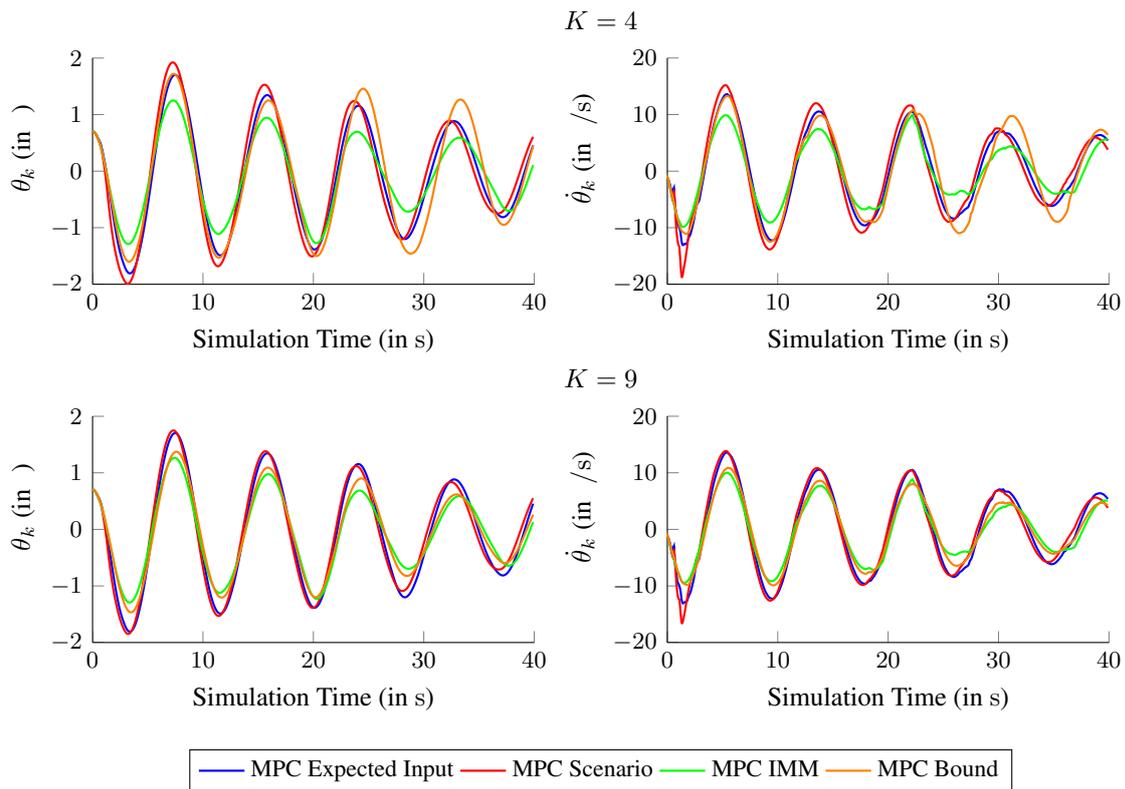
**Figure 5.9:** Control of an overhead crane: Evolution of the plant states  $\theta_k$  (left) and  $\dot{\theta}_k$  (right) in exemplary runs with  $K = 4$  (top row) and  $K = 9$  (bottom row) and network N1.

N1 and for all but the shortest one  $K = 4$  in the simulations with network N2. In particular for  $K = 9$ , the cost incurred by the proposed approaches in the simulation runs with network N1 are up to 20% lower compared to the cost incurred by the *MPC Expected Input*, while in the runs with network N2 up to 13% of the cost are saved. The increased control performance achieved by the *MPC IMM* and the *MPC Bound* leads to reduced oscillations of the load maneuvered by the crane, which is desirable in practice. We illustrate this observation with exemplary trajectories of the load angle  $\theta_k$  and its angular velocity  $\dot{\theta}_k$  from simulation runs with both networks N1 and N2 that are depicted in Figs. 5.9 and 5.10 for  $K = 4$  and  $K = 9$ .

Similar to the control task considered in the previous section, the *MPC Bound* exhibits a substantial performance gain as the length of the optimization horizon increases, rendering it clearly superior to the *MPC IMM* for horizon lengths  $K \geq 6$ . This results demonstrates the distinctive feature of the *MPC Bound* – its ability to take the impact of the dual effect into account. As the optimization horizon increases, the contribution of the dual effect to the cost becomes more and more significant, and neglecting it, as is done by the open-loop feedback approaches *MPC Expected Input*, *MPC Scenario*, and *MPC IMM*, entails a considerable loss of control performance.

## 5.6 Conclusions

Stochastic model predictive control (SMPC) is well-suited for networked control tasks thanks to its consistent and rigorous framework for the handling of uncertainties, allowing us to use the probabilistic network models provided by the CoCPN translator to predict the impact of packet delays and losses on the future evolution of the plant state. Hence, the goal of this chapter was to develop algorithms for



**Figure 5.10:** Control of an overhead crane: Evolution of the plant states  $\theta_k$  (left) and  $\dot{\theta}_k$  (right) in exemplary runs with  $K = 4$  (top row) and  $K = 9$  (bottom row) and network N2.

sequence-based SMPC based on the augmented dynamics (4.1) derived in Chapter 4. Conceptually, such an algorithm obtains control laws by formulating a stochastic optimal control problem, whose solutions then, in principle, can be found by means of the dynamic programming (DP) recursion (5.6) exploiting Bellman’s principle of optimality. However, the absence of reliable actuator feedback in the considered NCS introduces a dual effect, which prevented us from carrying out the DP recursion analytically and, even worse, rendered the numerical evaluation of the cost-to-go intractable. Thus, we focused on tractable approaches for the computation of suboptimal control laws and proposed two novel algorithms for sequence-based SMPC.

For the first algorithm, we exploited the connection between Markov jump linear systems and multiple model estimators, which maintain the estimate of the plant state in terms of a Gaussian mixture. Based on this connection and a tailored interacting multiple model filter, the proposed algorithm consists of a nonlinear control law that combines the output of multiple linear control laws, one for each mode of the augmented dynamics (4.1). Our key step in the course of the derivation was to write the cost-to-go at each stage of the DP recursion as a weighted sum of quadratic, mode-conditioned cost-to-go. To that end, we employed a global approximation of the cost-to-go, i.e., one that is valid for all possible state estimates provided by the filter, and neglected the portion of the cost related to the dual effect.

Our second presented algorithm possesses contrasting features – the underlying control law is linear, the cost-to-go is approximated only locally in the neighborhood of a reference trajectory obtained by propagating a given state estimate over the optimization horizon, and the impact of the dual effect on the cost is taken into account. For the last aspect, we had to find a way to anticipate the information provided by measurements to be received in the future and their impact on the quality of the propagated state estimates. The key idea to achieve this was to introduce an additional state augmentation and, then,

to reformulate the cost-to-go in terms of the second moment of this augmented state. The dynamics of the second moment is linear and depends only on the availability of measurements and not on the measured values. This property allowed us to generate closed-loop reference trajectories that integrate the information gained from measurements received in the future. For a given reference trajectory, we then derived an upper bound  $\bar{V}_t$  of the cost-to-go. Finally, we obtained a convergent iterative procedure for the computation of the parameters of a linear control law by minimizing the bound  $\bar{V}_t$  via repeated improvements of a given reference trajectory.

We evaluated the performance of the proposed algorithms in simulations with two different control tasks. Compared to state-of-the-art sequence-based SMPC approaches from the literature, the results showed that our approaches can achieve markedly better control performance. The simulation results also revealed that neither of the proposed approaches can be deemed superior to the other. The first one belongs to the class of open-loop feedback approaches and, as such, neglects the contribution of the dual effect to the cost-to-go, which necessarily yields suboptimal control laws. Especially for short optimization horizons  $K$ , however, the resulting loss of control performance can become insignificant. The second approach, on the other hand, is of the closed-loop feedback type, rendering it advantageous from a theoretical perspective as this implies that the impact of the dual effect on the cost is considered. However, this advantage comes at the cost of higher computational complexity and only pays off in terms of substantially improved control performance if the contribution of the dual effect is significant enough to outweigh the approximations we had to make during the derivation of the bound  $\bar{V}_t$  of the cost-to-go.

In Chapter 7, we will integrate the approaches for sequence-based SMPC developed in this chapter into the CoCPN architecture. For both approaches, we shall implement the CoCPN translator interface as described in Chapter 2 and demonstrate their applicability to quickly respond to changes of the network QoS and to achieve and maintain the given  $QoS_{\text{target}}$  in simulation scenarios with a challenging control task – the stabilization of an inverted double pendulum on a cart.

Before we do so, we will again turn our attention to the augmented dynamics (4.1) in Chapter 6. The mode transition probabilities are determined by the properties of the network model  $\tau_k^{\text{ca}}$ , which itself is derived based on status information collected inside the communication system, such as link and queue utilizations or packet error and loss rates (cf. Section 2.1 and Algorithm 2.2). Thus,  $\tau_k^{\text{ca}}$  is always an estimate of the “state” of the communication system and, consequently, the mode transition probabilities cannot be assumed perfectly known. Hence, questions about the influence of modeling errors on the control performance and about the existence of controllers that are, to some extent, robust to modeling errors naturally arise. We shall answer these questions in the next chapter.

## Sequence-Based Stochastic Robust Control

We have seen in the course of the discussion in Section 4.1 that the actual plant input  $\underline{u}_k$  directly bears upon the mode  $\theta_k$  of the augmented dynamical system (4.1). The subsequent results presented in Sections 4.2 and 4.3 then showed that the mode transition probabilities are determined by the properties of the network model  $\tau_k^{\text{ca}}$ , which is in turn based on monitoring feedback from the communication system. The collected monitoring data is based on noisy measurements itself and can only provide *snapshots* of the past. The resulting network model  $\tau_k^{\text{ca}}$  is thus always an estimate of the “state” of the underlying communication system. Consequently, the dynamics of  $\theta_k$ , and specifically the mode transition probabilities  $t_{k,ij}$ , are subject to modeling errors and cannot be assumed perfectly known. The SMPC approaches developed in Chapter 5, however, require the mode transition probabilities for the prediction of the plant behavior and the computation of the control sequences. Hence, it is natural to investigate the impact of modeling errors on the control performance and to devise control algorithms that are robust to modeling errors.

Research in this direction can be roughly divided into two categories. The first category comprises research that utilizes data-driven methods to estimate the unknown model parameters. For instance, an approach to estimate the parameters of a Gilbert-Elliott model for a wireless channel, which models packet losses by a Markov chain with two states [214, 215], is proposed in [216]. Similarly, for MJLS, estimators for the unknown mode transition probabilities have been proposed by the target tracking community [217–220]. Estimates of the plant state are provided as a by-product, making their usage very appealing in many control applications. However, the unknown transition probabilities are typically required to be time-invariant. Additionally, assessing the impact of estimation errors on the control performance is not straightforward.

Data-driven approaches that attempt to synthesize and analyze controllers without the need to estimate unknown model parameters also belong to the first category. In [221] and [222], control over wireless channels with Bernoulli losses is considered, where the packet loss rate is unknown. Having only a fixed number of channel samples at hand, i.e., for a set of packets it was recorded whether they were transmitted successfully, the authors present a data-driven approach to assess the performance of a given controller. The recent works [223–227] go one step further, as they combine the analysis of the impact of an unknown packet loss rate with the development of methods for controller synthesis. The key idea here is to apply methods from *reinforcement learning* which generally aims at finding control

laws for plants with unknown parameters or only partially known dynamics [228, 229]. However, the unknown packet loss rate is typically assumed to be fixed over time. Control approaches based on reinforcement learning are also presented in [230, 231], where the authors consider unknown delays in the communication of control inputs and measurements. Although the delays can be time-varying, they are assumed to be bounded by some known value. Model-free reinforcement learning methods based on policy iteration have also been successfully applied to compute control laws for MJLS with unknown mode transition probabilities in [232–235]. They demand, however, that the transition probabilities be time-invariant.

The second category of research that aims to take the impact of modeling errors into account consists of methods from *robust control* [236]. The general idea of these methods is to assume a particular representation of the unknown or uncertain model parameters. Then, one seeks to design controllers such that desired performance properties are satisfied for all uncertainties that are admissible with respect to the chosen representation. Specifically for MJLS a common approach in this regard is to assume that the mode transition matrix  $\mathbf{T}_k$  lies in the neighborhood of some nominal transition matrix [237] or varies in a finite set [238–240]. The latter is in fact a special case of a *polytopic* uncertainty, where the mode transition matrix is allowed to vary within a convex polytope [241–246]. That is, there exist  $R$  transition matrices  $\mathbf{\Lambda}^{(1)}, \mathbf{\Lambda}^{(2)}, \dots, \mathbf{\Lambda}^{(R)}$ , referred to as *vertices*, such that at each time step,  $\mathbf{T}_k$  can be expressed in terms of a convex combination

$$\mathbf{T}_k = \sum_{r=1}^R \alpha_k^{(r)} \mathbf{\Lambda}^{(r)}, \quad \sum_{r=1}^R \alpha_k^{(r)} = 1, \quad \alpha_k^{(1)}, \alpha_k^{(2)}, \dots, \alpha_k^{(R)} \geq 0. \quad (6.1)$$

If we denote the set of vertices by  $\mathcal{L}$ , then (6.1) is compactly written by means of its convex hull  $\text{conv}(\mathcal{L})$  as

$$\mathbf{T}_k \in \text{conv}(\mathcal{L}). \quad (6.2)$$

Markov jump linear systems with mode transition transition matrices that fulfill (6.1), or, equivalently, (6.2) are referred to as *polytopic MJLS*. They have gained significant research attention because they also cover cases where some transition probabilities are bounded or even completely unknown [247–249]. In that regard, (6.1) can be seen as the natural representation of uncertain transition matrices.

In this chapter, we exploit the observation made in Chapter 4 that the mode transition matrix  $\mathbf{T}_k$  associated with the augmented dynamics (4.1) is always a lower Hessenberg matrix. We use this observation to show that  $\mathbf{T}_k$  varies within a convex polytope as defined by (6.1), whose number of vertices  $R$  solely depends on the control sequence length  $N$ . A polytopic MJLS is thus a convenient way to tackle uncertainties in the network model. Consequently, the synthesis of sequence-based controllers that are robust to uncertainties in the network model translates into the synthesis of controllers for polytopic MJLS. The goal of this chapter is to synthesize a mode-independent state feedback controller for polytopic MJLS that guarantees the stability of the closed-loop dynamics. More specifically, we seek a control law that stabilizes (4.1) in the mean square sense according to the following definitions.

**Definition 6.1:** (Costa et al. [127, Definition 3.8])

The MJLS (4.1) with  $\underline{U}_k \equiv \underline{0}$  is mean square stable (MSS) if for any initial condition  $\underline{\psi}_0$  and  $\theta_0$  it holds

$$\lim_{k \rightarrow \infty} \left\| \mathbb{E} \left\{ \underline{\psi}_k \right\} \right\| = 0, \quad (6.3)$$

$$\lim_{k \rightarrow \infty} \left\| \mathbb{E} \left\{ \underline{\psi}_k \underline{\psi}_k^T \right\} - \mathbf{\Psi} \right\| = 0, \quad (6.4)$$

for some  $\mathbf{\Psi} \in \mathbb{R}^{n_\psi \times n_\psi}$  that is independent of  $\underline{\psi}_0$  and  $\theta_0$ .

**Definition 6.2:** (Costa et al. [127, Definition 3.40])

The MJLS (4.1) is mean square stabilizable if there is a matrix  $\mathbf{L} \in \mathbb{R}^{Nn_u \times n_\psi}$  such that (4.1) is MSS when  $\underline{U}_k = \mathbf{L}\underline{\psi}_k$ . In this case,  $\mathbf{L}$  is said to stabilize the MJLS.

Note that the conditions (6.3) and (6.4) are independent of the chosen norm because on finite-dimensional spaces all norms are equivalent (cf. Appendix A.2). Based on Definition 6.1, we first derive a necessary and sufficient condition for the mean square stability of polytopic MJLS, which requires the calculation of the *joint spectral radius (JSR)* of a set of matrices. The condition itself is not new and has already appeared in [244, 250, 251]. There, however, only noise-free polytopic MJLS and polytopic MJLS with bounded disturbances were considered, whereas, here, the polytopic MJLS is driven by wide-sense stationary noise. To determine whether a given polytopic MJLS is MSS based on the derived condition is  $\mathcal{NP}$ -hard, so that its practical usefulness is limited. However, it enables us to derive a sufficient stability condition in terms of a *linear matrix inequality (LMI)* feasibility problem. The LMI condition is less restrictive than the ones proposed in [245, 248] and can be evaluated by state-of-the-art solvers (usually) in polynomial time. As a by-product, it immediately enables us to determine the existence of a state feedback law that stabilizes the polytopic MJLS according to Definition 6.2.

In Section 6.1, we first introduce the joint spectral radius and summarize some properties that are needed for the derivation of the chief results of this chapter. Then, in Section 6.2, we construct the polytope for the mode transition matrix  $\mathbf{T}_k$  based on the observations from Sections 4.2 and 4.3. Sections 6.3 and 6.4 constitute the main contribution of this chapter and present the derivation of the conditions for mean square stability and the existence of a stabilizing control law. Finally, we provide a numerical example in Section 6.5 to illustrate the results.

*This chapter is based on results presented in our publication [286].*

## 6.1 The Joint Spectral Radius

Recall that the *spectral radius* of a square matrix  $\mathbf{M} \in \mathbb{R}^{n \times n}$  is defined as the maximum modulus of its eigenvalues, i.e.,

$$\rho(\mathbf{M}) = \max\{|\lambda| : \lambda \text{ is eigenvalue of } \mathbf{M}\}.$$

The spectral radius quantifies the asymptotic growth rate of any norm of the powers of a matrix because of Gelfand's formula [252, Corollary 5.6.14], which states that

$$\rho(\mathbf{M}) = \lim_{k \rightarrow \infty} \sqrt[k]{\|\mathbf{M}^k\|}.$$

The desire to have a related characterization for the asymptotic growth rate of the norm of “long” products of a set of matrices led to the introduction of the *joint spectral radius (JSR)* [253, 254]. For a formal definition, let  $\mathcal{M}$  be a set of square matrices and denote by  $\Pi_k(\mathcal{M})$  the set of all products of length  $k$  whose factors are elements of  $\mathcal{M}$ , i.e.,

$$\Pi_k(\mathcal{M}) = \left\{ \prod_{i=1}^k \mathbf{M}_i \mid \mathbf{M}_i \in \mathcal{M} \right\}. \quad (6.5)$$

For any  $\mathbf{P}_k \in \Pi_k(\mathcal{M})$  and any norm, we call  $\sqrt[k]{\|\mathbf{P}_k\|}$  its *averaged norm* and define

$$\hat{\rho}_k(\mathcal{M}) = \sup_{\mathbf{P}_k \in \Pi_k(\mathcal{M})} \sqrt[k]{\|\mathbf{P}_k\|}. \quad (6.6)$$

For bounded sets, the limit  $\lim_{k \rightarrow \infty} \hat{\rho}_k(\mathcal{M})$  is well-defined and thus always exists. Moreover, it is independent of the chosen norm [254, 255]. Consequently, the following definition is justified.

**Definition 6.3:**

For a bounded set of square matrices  $\mathcal{M}$ , its joint spectral radius, denoted by  $\hat{\rho}(\mathcal{M})$ , is defined as the limit of the sequence  $\{\hat{\rho}_k(\mathcal{M})\}$  defined by (6.6), that is,

$$\hat{\rho}(\mathcal{M}) = \lim_{k \rightarrow \infty} \hat{\rho}_k(\mathcal{M}).$$

In the remainder, we will make use of the following two facts. The first one, given in Theorem 6.1, is the analog of the well-known result that  $\rho(\mathbf{M}) < 1 \Leftrightarrow \lim_{k \rightarrow \infty} \mathbf{M}^k = \mathbf{0}$ . The second one, given in Theorem 6.2, states that the taking the convex hull of a set of matrices does not change the JSR. Proofs are, for instance, given in [255] and [254], respectively.

**Theorem 6.1:**

For any bounded set of matrices  $\mathcal{M}$ ,  $\hat{\rho}(\mathcal{M}) < 1$  if and only if any  $\mathbf{P}_k \in \Pi_k(\mathcal{M})$  converges to  $\mathbf{0}$  as  $k \rightarrow \infty$ .

**Theorem 6.2:**

For any bounded set of matrices  $\mathcal{M}$ , it holds  $\hat{\rho}(\mathcal{M}) = \hat{\rho}(\text{conv}(\mathcal{M}))$ , where  $\text{conv}(\mathcal{M})$  is the convex hull of  $\mathcal{M}$ .

Additionally, the results presented in the next lemma will prove useful in the subsequent derivations.

**Lemma 6.1:**

Let  $\mathcal{M}$  be a finite set of  $n$ -by- $n$  matrices and suppose  $\hat{\rho}(\mathcal{M}) < 1$ . Then the following propositions are true:

- (i) There exist  $\xi \geq 1$  and  $\beta \in (0, 1)$  such that for all  $\mathbf{P}_k \in \Pi_k(\text{conv}(\mathcal{M}))$  it holds  $\|\mathbf{P}_k\| \leq \xi\beta^k$  for all  $k \in \mathbb{N}$ .
- (ii) For any  $\mathbf{P}_k \in \Pi_k(\text{conv}(\mathcal{M}))$ ,  $\|\mathbf{P}_k\| \rightarrow 0$  as  $k \rightarrow \infty$ .
- (iii) The series  $\sum_{k=0}^{\infty} \|\mathbf{P}_k\|$  is convergent for all possible  $\mathbf{P}_k \in \Pi_k(\text{conv}(\mathcal{M}))$ .

*Proof.* The proof is provided in Appendix D.1. □

In the next section, we will show that the mode transition matrix, for any realization of the network model  $\tau_k^{\text{ca}}$ , always lies in a polytope spanned by  $R$  vertices, where  $R$  is a function of the employed sequence length  $N$ . Subsequently, in Section 6.3, we will construct a set  $\mathcal{A}_R$  that consists of  $R$  matrices, each of which is associated with one vertex of the polytope. Stability of the polytopic MJLS then proves to hinge on the question if  $\hat{\rho}(\mathcal{A}_R) < 1$ .

## 6.2 Derivation of the Polytopic MJLS

In Chapter 3, we posited Assumption 3.3 to ensure that the control sequences are “long enough” so that the probability of two successive applications of the default input  $\underline{u}_k^{\text{df}}$  is less than one. From the discussion in Chapter 4 it is clear that this assumption implies that

$$\mathbb{P}[\theta_{k+1} = N | \theta_k = N] \leq \delta, \quad (6.7)$$

for some  $\delta < 1$ . We will exploit this assumption in the following sections. If it were not to hold, i.e., if we allowed  $P[\theta_{k+1} = N | \theta_k = N] = 1$ , we would allow the pathological case that the plant eventually remains in open-loop operation, rendering any control effort futile.<sup>1</sup> In this case, however, stability of the closed-loop dynamics necessitates that the plant has stable dynamics in the first place.

Before we proceed, let us denote by  $\Delta^n$  the  $n$ -dimensional *standard simplex* or *probability simplex* [256, p. 33]

$$\Delta^n = \{ \underline{x} \in \mathbb{R}^{n+1} \mid \underline{x} \geq 0, \mathbf{1}_{n+1}^T \underline{x} = 1 \},$$

that contains all  $(n+1)$ -dimensional vectors with nonnegative entries that sum to one. The vertices of  $\Delta^n$  are the  $n+1$  standard basis vectors  $\underline{e}_{n+1}^{(i)}$  in  $\mathbb{R}^{n+1}$  [256, p. 33].

More generally, any set of  $n+1$  affinely independent<sup>2</sup> vectors  $\underline{v}_1, \dots, \underline{v}_{n+1} \in \mathbb{R}^m$ ,  $m \geq n$ , determines an  $n$ -dimensional *simplex* in  $\mathbb{R}^m$  given by [256, p. 32]

$$\mathcal{S}^n = \{ \alpha_1 \underline{v}_1 + \dots + \alpha_{n+1} \underline{v}_{n+1} \mid \alpha_i \geq 0, \alpha_1 + \dots + \alpha_{n+1} = 1 \}.$$

### 6.2.1 Polytopic MJLS: Independent Packet Delays and Losses

We first recall our observations from Section 4.2 (cf. Fig. 4.3). There, we concluded from Theorem 4.1 that the mode transition matrix  $\mathbf{T}_k$  is a lower Hessenberg matrix, whose subdiagonal entries in each column are equal to the column's diagonal entry, and whose last two rows are equal. These observations imply that the  $N+1$  elements  $t_{k,N0}, t_{k,N1}, \dots, t_{k,NN}$  of the last row completely determine  $\mathbf{T}_k$ . Eq. (6.7) states that for the last element it always holds  $t_{k,NN} \in [0, \delta]$  for some  $\delta < 1$ , whereas the other elements can attain any value in  $[0, 1]$  subject to the constraint that  $\sum_{i=0}^N t_{k, Ni} = 1$ . The following result is a direct consequence.

#### Theorem 6.3:

*Consider the augmented dynamics (4.1) with mode transition matrix  $\mathbf{T}_k$  computed according to Theorem 4.1. Then, there exist  $R = 2N$  vertices  $\mathbf{\Lambda}^{(1)}, \mathbf{\Lambda}^{(2)}, \dots, \mathbf{\Lambda}^{(R)} \in \mathbb{R}^{N+1 \times N+1}$  such that  $\mathbf{T}_k$  can be written according to (6.1), i.e.,*

$$\mathbf{T}_k = \sum_{r=1}^R \alpha_k^{(r)} \mathbf{\Lambda}^{(r)}, \quad \sum_{r=1}^R \alpha_k^{(r)} = 1, \quad \alpha_k^{(1)}, \alpha_k^{(2)}, \dots, \alpha_k^{(R)} \geq 0.$$

for some coefficients  $\alpha_k^{(r)}$ .

*Proof.* The proof is constructive and exploits that the last row  $\underline{t}_k^T = [t_{k,N0} \ t_{k,N1} \ \dots \ t_{k,NN}]$  completely determines  $\mathbf{T}_k$ . All elements  $t_{k, Ni}$  must be nonnegative and sum to one and additionally we have  $t_{k, NN} \in [0, \delta]$  for some  $\delta < 1$ , so that we can write  $\underline{t}_k \in \mathcal{F}$  with  $\mathcal{F}$  given by

$$\mathcal{F} = \left\{ \underline{x} \in \mathbb{R}^{N+1} \mid \underline{x} \geq 0, \mathbf{1}_{N+1}^T \underline{x} = 1, \underline{x}^T \underline{e}_{N+1}^{(N+1)} \leq \delta \right\}. \quad (6.8)$$

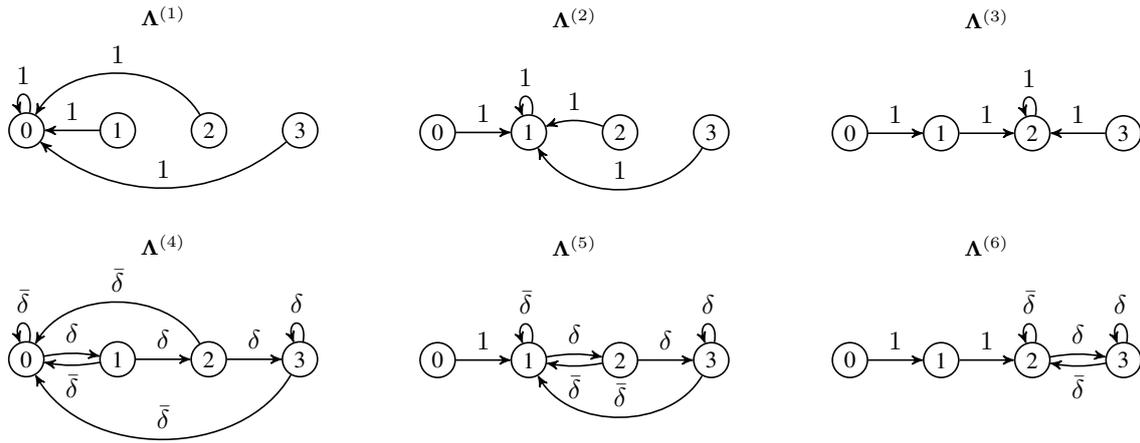
With the definition of the half-space  $\mathcal{G}$  according to

$$\mathcal{G} = \left\{ \underline{x} \in \mathbb{R}^{N+1} \mid \underline{x}^T \underline{e}_{N+1}^{(N+1)} \leq \delta \right\},$$

we may compactly express the set as  $\mathcal{F} = \Delta^N \cap \mathcal{G}$ . The intersection of a simplex with a half-space is called a *frustum* of a simplex [257]. For  $N = 2$ , the frustum  $\mathcal{F}$  is an isosceles trapezoid as illustrated in Fig. 6.2.

<sup>1</sup>Note that this case could also eventuate if the actuator used a hold-input strategy instead of the zero-input strategy.

<sup>2</sup>The vectors  $\underline{v}_1, \dots, \underline{v}_{n+1}$  are *affinely independent* if  $\underline{v}_2 - \underline{v}_1, \dots, \underline{v}_{n+1} - \underline{v}_1$  are linearly independent.



**Figure 6.1:** State transition diagrams of the vertices  $\Lambda^{(1)}, \Lambda^{(2)}, \dots, \Lambda^{(6)}$  resulting from the application of Theorem 6.3 with  $N = 3$ . For better readability, we set  $\bar{\delta} = 1 - \delta$ .

The vertices of  $\mathcal{F}$  are given by the  $N$  vertices of  $\Delta^N$  that reside in  $\mathcal{G}$

$$\underline{v}^{(r)} = \underline{e}_{N+1}^{(r)}, \quad (6.9)$$

in addition to the  $N$  intersection points of  $\Delta^N$  with the boundary of  $\mathcal{G}$

$$\underline{v}^{(N+r)} = \begin{bmatrix} (1 - \delta)\underline{e}_N^{(r)} \\ \delta \end{bmatrix}, \quad (6.10)$$

with  $r = 1, 2, \dots, N$ .

Consequently, the last row is expressed as

$$\underline{t}_k^T = \sum_{r=1}^N \alpha_k^{(r)} \left( \underline{v}^{(r)} \right)^T + \alpha_k^{(N+r)} \left( \underline{v}^{(N+r)} \right)^T,$$

which is a convex combination of the  $R = 2N$  vectors given by (6.9) and (6.10). The vertices  $\Lambda^{(1)}, \Lambda^{(2)}, \dots, \Lambda^{(R)}$  then directly follow from the dependencies between the elements of  $\mathbf{T}_k$ .  $\square$

For  $N = 3$ , the state transition diagrams of the resulting six vertices are depicted in Fig. 6.1. The figure allows an interpretation of the vertices in terms of the delay and loss probabilities. The vertices depicted in the top row correspond to networks with “deterministic” behavior where it holds  $\text{P}[\tau_k^{\text{ca}} = 0] = 1$ ,  $\text{P}[\tau_k^{\text{ca}} = 1] = 1$ , and  $\text{P}[\tau_k^{\text{ca}} = 2] = 1$ , respectively. The vertices in the bottom row describe networks where control sequences either experience a fixed delay of  $i$  time steps or reach the actuator too late to be applied (or get lost), i.e., it holds

$$\text{P}[\tau_k^{\text{ca}} = i] = 1 - \delta, \quad \text{P}[\tau_k^{\text{ca}} > N - 1] = \delta,$$

for  $i = 0, 1, 2$ . We conclude this section with the remark that the number of vertices reduces to  $R = N$  in case  $\text{P}[\theta_{k+1} = N | \theta_k = N] = 0$ , i.e., if we have  $\delta = 0$  in (6.7). The frustum  $\mathcal{F}$  then becomes an  $(N - 1)$ -dimensional simplex  $\mathcal{S}^{N-1}$  in  $\mathbb{R}^{N+1}$ , which, for  $N = 2$ , is a line segment (cf. Fig. 6.2).

## 6.2.2 Polytopic MJLS: Correlated Packet Delays and Losses

As mentioned in Section 4.3, the mode transition matrix is no longer determined by its last row in case of correlated packet delays and losses when it is computed according to (4.24). Moreover, there are generally no dependencies between the entries. However, it is always a lower Hessenberg matrix of the form

$$\mathbf{T} = \begin{bmatrix} [0, 1] & [0, 1] & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & [0, 1] & [0, 1] \\ [0, 1] & \dots & \dots & [0, 1] & [0, \delta] \end{bmatrix}, \quad (6.11)$$

where the notation  $[0, a]$  indicates that the respective element can take any value between zero and  $a$ . Together with (6.7), this observation suffices to establish the following result.

### Theorem 6.4:

Consider the augmented dynamics (4.1) with mode transition matrix  $\mathbf{T}$  computed according to (4.24). Then, there exist  $R = 2N(N+1)!$  vertices  $\mathbf{\Lambda}^{(1)}, \mathbf{\Lambda}^{(2)}, \dots, \mathbf{\Lambda}^{(R)} \in \mathbb{R}^{N+1 \times N+1}$  such that  $\mathbf{T}$  can be written according to (6.1), i.e.,

$$\mathbf{T} = \sum_{r=1}^R \alpha^{(r)} \mathbf{\Lambda}^{(r)}, \quad \sum_{r=1}^R \alpha^{(r)} = 1, \quad \alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(R)} \geq 0.$$

for some coefficients  $\alpha^{(r)}$ .

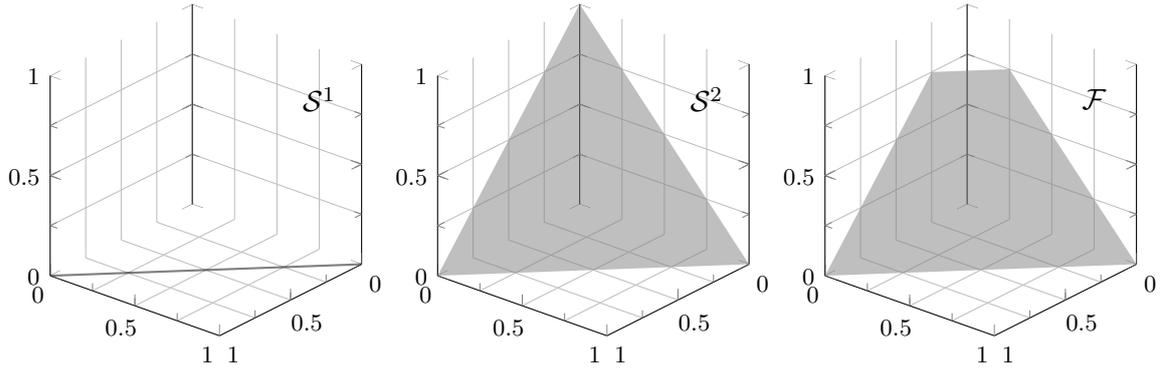
*Proof.* We prove the proposition constructively by showing that  $\mathbf{T}$  always lies in a *proprism*, which is defined as the Cartesian product of multiple polytopes. More specifically, we show that  $\mathbf{T}$  lies in a proprism  $\mathcal{P}$  resulting from the Cartesian product of  $N$  simplexes  $\mathcal{S}^n$  in  $\mathbb{R}^{N+1}$  and the frustum  $\mathcal{F}$  given by (6.8) that we already encountered in the proof of the preceding theorem. To that end, we use that  $\mathbf{T}$  is of the form (6.11) subject to the constraint that all rows must sum to one.

The first row has two unknown entries that must sum to one so that it can be written as  $\underline{t}_1^T = [\underline{z}_1^T \ 0]$  for some  $\underline{z}_1 \in \Delta^1$ . Hence, for some  $\alpha_1^{(1)}, \alpha_1^{(2)} \geq 0$ ,  $\alpha_1^{(1)} + \alpha_1^{(2)} = 1$

$$\begin{aligned} \underline{t}_1 &= \alpha_1^{(1)} \begin{bmatrix} \underline{e}_2^{(1)} \\ \underline{0} \end{bmatrix} + \alpha_1^{(2)} \begin{bmatrix} \underline{e}_2^{(2)} \\ \underline{0} \end{bmatrix} \\ &= \alpha_1^{(1)} \underline{v}_1^{(1)} + \alpha_1^{(2)} \underline{v}_1^{(2)}, \end{aligned}$$

i.e.,  $\underline{t}_1$  lies in a 1-simplex  $\mathcal{S}^1$  (a line segment, cf. Fig. 6.2) spanned by the vertices  $\underline{v}_1^{(1)}$  and  $\underline{v}_1^{(2)}$ . Similarly, the second row has three unknown entries that must sum to one so that it can be written as  $\underline{t}_2^T = [\underline{z}_2^T \ 0]$  for some  $\underline{z}_2 \in \Delta^2$ . Consequently, it lies in a 2-simplex  $\mathcal{S}^2$  (a triangle, cf. Fig. 6.2) determined by the vertices  $\underline{v}_2^{(1)}$ ,  $\underline{v}_2^{(2)}$ , and  $\underline{v}_2^{(3)}$  as per

$$\begin{aligned} \underline{t}_2 &= \alpha_2^{(1)} \underline{v}_2^{(1)} + \alpha_2^{(2)} \underline{v}_2^{(2)} + \alpha_2^{(3)} \underline{v}_2^{(3)}, \\ &= \alpha_2^{(1)} \begin{bmatrix} \underline{e}_3^{(1)} \\ \underline{0} \end{bmatrix} + \alpha_2^{(2)} \begin{bmatrix} \underline{e}_3^{(2)} \\ \underline{0} \end{bmatrix} + \alpha_2^{(3)} \begin{bmatrix} \underline{e}_3^{(3)} \\ \underline{0} \end{bmatrix}. \end{aligned}$$



**Figure 6.2:** Graphical illustration of the simplexes  $\mathcal{S}^1$ ,  $\mathcal{S}^2$ , and the frustum  $\mathcal{F}$  for  $N = 2$  and  $\delta = 0.75$ .

In the same vein, we obtain that the  $i$ -th row lies in an  $i$ -simplex  $\mathcal{S}^i$  spanned by the  $i + 1$  vertices  $\underline{v}_i^{(r)}$

$$\underline{t}_i = \sum_{r=1}^{i+1} \alpha_i^{(r)} \underline{v}_i^{(r)} = \sum_{r=1}^{i+1} \alpha_i^{(r)} \begin{bmatrix} \underline{e}_{i+1}^{(r)} \\ \underline{0} \end{bmatrix},$$

for  $i = 3, \dots, N$ , and where  $\underline{0} \in \mathbb{R}^{(N-i)}$ . The last row  $\underline{t}_{N+1}^T$  is again of the form

$$[[0, 1] \quad \dots \quad \dots \quad [0, 1] \quad [0, \delta]],$$

so that it lies in the frustum  $\mathcal{F}$  given by (6.8). According to (6.9) and (6.10),  $\mathcal{F}$  is spanned by the  $2N$  vertices

$$\underline{v}_{N+1}^{(r)} = \underline{e}_{N+1}^{(r)}, \quad \underline{v}_{N+1}^{(N+r)} = \begin{bmatrix} (1 - \delta) \underline{e}_N^{(r)} \\ \delta \end{bmatrix},$$

with  $r = 1, 2, \dots, N$ . For  $N = 2$ ,  $\mathcal{F}$  is an isosceles trapezoid as illustrated in Fig. 6.2.

The resulting proprism  $\mathcal{P}$  is the Cartesian product of the  $N$  simplexes  $\mathcal{S}^1, \dots, \mathcal{S}^N$ , and the frustum  $\mathcal{F}$  according to

$$\mathcal{P} = \mathcal{S}^1 \times \mathcal{S}^2 \times \dots \times \mathcal{S}^N \times \mathcal{F} = \{(\underline{t}_1, \underline{t}_2, \dots, \underline{t}_N, \underline{t}_{N+1}) \mid \underline{t}_{N+1} \in \mathcal{F}, \underline{t}_i \in \mathcal{S}^i, i = 1, 2, \dots, N\}.$$

Every tuple  $(\underline{t}_1, \underline{t}_2, \dots, \underline{t}_N, \underline{t}_{N+1})$  in  $\mathcal{P}$  then yields a transition matrix  $\mathbf{T}$  if we arrange its elements row-wise, i.e.,

$$\mathbf{T} = \begin{bmatrix} \underline{t}_1^T \\ \underline{t}_2^T \\ \vdots \\ \underline{t}_N^T \\ \underline{t}_{N+1}^T \end{bmatrix}.$$

The vertices of  $\mathcal{P}$  are the Cartesian product of the vertices of the individual polytopes, i.e., every vertex is a tuple of the form

$$\left( \underline{v}_1^{(r_1)}, \underline{v}_2^{(r_2)}, \dots, \underline{v}_N^{(r_N)}, \underline{v}_{N+1}^{(r_{N+1})} \right),$$

with  $r_1 \in \{1, 2\}$ ,  $r_2 \in \{1, 2, 3\}$ ,  $\dots$ ,  $r_N \in \{1, \dots, N + 1\}$ , and  $r_{N+1} \in \{1, \dots, 2N\}$ , implying that  $\mathcal{P}$  has

$$R = 2N \prod_{j=2}^{N+1} j = 2N(N + 1)!,$$

vertices. Row-wise arrangement of the elements  $\underline{v}_i^{(r_j)}$  then yields the vertices  $\Lambda^{(r)}$ .  $\square$

Note that again the number of vertices is reduced by half if  $\delta = 0$  in (6.7), i.e., we have  $R = N(N+1)!$  in case  $P[\theta_{k+1} = N | \theta_k = N] = 0$ .

### 6.3 A Necessary and Sufficient Stability Condition

The previous section showed that an uncertain or unknown network model  $\tau_k^{\text{ca}}$  renders the augmented dynamical system (4.1) a polytopic MJLS, thereby translating modeling errors into a polytopic parameter uncertainty. Mean square stability as defined in Definition 6.1 is an asymptotic property and demands the investigation of the convergence properties of the sequences  $\{\|\hat{\underline{\psi}}_k\|\}$  and  $\{\|\Psi_k\|\}$ , where

$$\begin{aligned}\hat{\underline{\psi}}_k &= \mathbb{E}\{\underline{\psi}_k\}, \\ \Psi_k &= \mathbb{E}\{\underline{\psi}_k \underline{\psi}_k^T\},\end{aligned}$$

are the first and second moment of the augmented state  $\underline{\psi}_k$ . These sequences are convergent if and only if

$$\lim_{k \rightarrow \infty} \hat{\underline{\psi}}_k = \underline{\psi}, \quad (6.12)$$

$$\lim_{k \rightarrow \infty} \Psi_k = \Psi, \quad (6.13)$$

hold, that is, if and only if the first and second moment converge. In this section, we establish a condition that is both necessary and sufficient for the mean square stability of the uncontrolled version of the MJLS (4.1) whose mode transition matrix varies within a polytope with vertices  $\mathcal{L} = \{\Lambda^{(1)}, \Lambda^{(2)}, \dots, \Lambda^{(R)}\}$ , i.e.,

$$\begin{aligned}\underline{\psi}_{k+1} &= \tilde{\mathbf{A}}^{(\theta_k)} \underline{\psi}_k + \tilde{\mathbf{w}}_k, \\ \mathbf{T}_k &\in \text{conv}(\mathcal{L}).\end{aligned} \quad (6.14)$$

Denoting the entries of  $\Lambda^{(r)}$  by  $\lambda_{ij}^{(r)}$ , (6.14) implies that we can write the mode transition probabilities as

$$t_{k,ij} = \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)}, \quad \sum_{r=1}^R \alpha_k^{(r)} = 1, \quad \alpha_k^{(1)}, \alpha_k^{(2)}, \dots, \alpha_k^{(R)} \geq 0. \quad (6.15)$$

We shall establish the stability condition in two steps. First, we derive a necessary and sufficient condition for the convergence of the second moment as per (6.13). Then, we show that (6.12) holds with  $\underline{\psi} = \underline{0}$  whenever the second moment converges.

The main machinery for the derivation of a necessary and sufficient condition for the convergence of the second moment  $\Psi_k$  is a vector-valued sequence  $\{\underline{z}_k\}$  that is convergent if and only if the matrix-valued sequence  $\{\Psi_k\}$  is convergent. To construct  $\{\underline{z}_k\}$ , we proceed similar to Section 5.4 and decompose  $\Psi_k$  into  $N+1$  matrices according to

$$\Psi_k^{(i)} = \mathbb{E}_{\underline{\psi}_k, \theta_k} \left\{ \underline{\psi}_k \underline{\psi}_k^T \mathbf{1}_{\{\theta_k=i\}} \right\}, \quad (6.16)$$

for  $i = 0, 1, \dots, N$ , so that it holds [127, p. 31]

$$\Psi_k = \sum_{i=0}^N \Psi_k^{(i)}. \quad (6.17)$$

Letting

$$\mathcal{Z}_k = \left( \Psi_k^{(0)}, \Psi_k^{(1)}, \dots, \Psi_k^{(N)} \right), \quad (6.18)$$

we may write  $\mathcal{Z}_k \in \mathbb{H}^{n_\psi}$ , where we recall from Section 5.4.2 the space  $\mathbb{H}^{n_\psi}$  as the set of all  $(N+1)$ -tuples of  $n_\psi$ -by- $n_\psi$  matrices, i.e.,

$$\mathbb{H}^{n_\psi} \triangleq \left\{ \left( \mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)} \right) \mid \mathbf{X}^{(i)} \in \mathbb{R}^{n_\psi \times n_\psi} \right\}.$$

$\mathbb{H}^{n_\psi}$  is a finite-dimensional *Banach space* (cf. Appendix A.7) with addition and scalar multiplication defined as

$$\begin{aligned} \mathcal{X} + \mathcal{Y} &\triangleq \left( \mathbf{X}^{(0)} + \mathbf{Y}^{(0)}, \mathbf{X}^{(1)} + \mathbf{Y}^{(1)}, \dots, \mathbf{X}^{(N)} + \mathbf{Y}^{(N)} \right), \\ \alpha \mathcal{X} &\triangleq \left( \alpha \mathbf{X}^{(0)}, \alpha \mathbf{X}^{(1)}, \dots, \alpha \mathbf{X}^{(N)} \right), \end{aligned}$$

for  $\mathcal{X} = (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)})$ ,  $\mathcal{Y} = (\mathbf{Y}^{(0)}, \mathbf{Y}^{(1)}, \dots, \mathbf{Y}^{(N)}) \in \mathbb{H}^{n_\psi}$ ,  $\alpha \in \mathbb{R}$ , and with norm  $\|\cdot\|_{\mathbb{H}}$  given by

$$\|\mathcal{X}\|_{\mathbb{H}} \triangleq \sum_{i=0}^N \|\mathbf{X}^{(i)}\|, \quad (6.19)$$

for any matrix norm  $\|\cdot\|$ . Moreover,  $\mathbb{H}^{n_\psi}$  is *uniformly homeomorphic* to  $\mathbb{R}^{(N+1)n_\psi^2}$  via the mapping  $\hat{\varphi}: \mathbb{H}^{n_\psi} \mapsto \mathbb{R}^{(N+1)n_\psi^2}$  defined as [127, 258]

$$\hat{\varphi}(\mathcal{X}) \triangleq \begin{bmatrix} \text{vec}(\mathbf{X}^{(0)}) \\ \text{vec}(\mathbf{X}^{(1)}) \\ \vdots \\ \text{vec}(\mathbf{X}^{(N)}) \end{bmatrix}. \quad (6.20)$$

Hence, the sequence of tuples  $\{\mathcal{Z}_k\}$  as introduced above in (6.18) converges if and only if the corresponding sequence of vectors  $\{\underline{z}_k\}$  with  $\underline{z}_k = \hat{\varphi}(\mathcal{Z}_k)$  converges (cf. Theorem A.1).

The first result of this section states that the dynamics of  $\underline{z}_k$  is linear. To facilitate its presentation, let us first introduce

$$\begin{aligned} \tilde{\mathbf{W}}_k^{(i)} &= \mathbb{E}_{\tilde{\mathbf{w}}_k, \theta_k} \left\{ \tilde{\mathbf{w}}_k \tilde{\mathbf{w}}_k^T \mathbb{1}_{\{\theta_k=i\}} \right\} \in \mathbb{R}^{n_\psi \times n_\psi}, \\ \tilde{\mathcal{W}}_k &= \left( \tilde{\mathbf{W}}_k^{(0)}, \tilde{\mathbf{W}}_k^{(1)}, \dots, \tilde{\mathbf{W}}_k^{(N)} \right) \in \mathbb{H}^{n_\psi}, \\ \underline{w}'_k &= \hat{\varphi}(\tilde{\mathcal{W}}_k) \in \mathbb{R}^{(N+1)n_\psi^2}, \end{aligned} \quad (6.21)$$

and

$$\underline{\mathbf{A}} = \begin{bmatrix} \tilde{\mathbf{A}}^{(0)} \otimes \tilde{\mathbf{A}}^{(0)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{A}}^{(1)} \otimes \tilde{\mathbf{A}}^{(1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \tilde{\mathbf{A}}^{(N)} \otimes \tilde{\mathbf{A}}^{(N)} \end{bmatrix} \in \mathbb{R}^{(N+1)n_\psi^2 \times (N+1)n_\psi^2}. \quad (6.22)$$

**Lemma 6.2:**

Let  $\underline{z}_k = \hat{\varphi}(\mathcal{Z}_k)$  with  $\mathcal{Z}_k$  and  $\hat{\varphi}(\cdot)$  according to (6.18) and (6.20). Then, the dynamics of  $\underline{z}_k$  is linear and given by

$$\underline{z}_{k+1} = \mathbf{A}'_k \underline{z}_k + \mathbf{G}'_k \underline{w}'_k, \quad \underline{z}_0 = \hat{\varphi}(\mathcal{Z}_0), \quad (6.23)$$

with  $\underline{w}'_k$  as per (6.21),  $\mathbf{A}'_k, \mathbf{G}'_k \in \mathbb{R}^{(N+1)n_\psi^2 \times (N+1)n_\psi^2}$  according to

$$\mathbf{A}'_k = \sum_{r=1}^R \alpha_k^{(r)} \left( \left( \mathbf{\Lambda}^{(r)} \right)^T \otimes \mathbf{I}_{n_\psi^2} \right) \underline{\mathbf{A}}, \quad (6.24)$$

$$\mathbf{G}'_k = \sum_{r=1}^R \alpha_k^{(r)} \left( \left( \mathbf{\Lambda}^{(r)} \right)^T \otimes \mathbf{I}_{n_\psi^2} \right), \quad (6.25)$$

and where

$$\sum_{r=1}^R \alpha_k^{(r)} = 1, \quad \alpha_k^{(1)}, \alpha_k^{(2)}, \dots, \alpha_k^{(R)} \geq 0.$$

*Proof.* The proof is provided in Appendix D.2.  $\square$

In view of (6.24), denote by  $\mathcal{A}_R$  the set

$$\mathcal{A}_R = \left\{ \left( \left( \mathbf{\Lambda}^{(1)} \right)^T \otimes \mathbf{I}_{n_\psi^2} \right) \underline{\mathbf{A}}, \left( \left( \mathbf{\Lambda}^{(2)} \right)^T \otimes \mathbf{I}_{n_\psi^2} \right) \underline{\mathbf{A}}, \dots, \left( \left( \mathbf{\Lambda}^{(R)} \right)^T \otimes \mathbf{I}_{n_\psi^2} \right) \underline{\mathbf{A}} \right\}, \quad (6.26)$$

with  $\underline{\mathbf{A}}$  given by (6.22), so that we can write  $\mathbf{A}'_k \in \text{conv}(\mathcal{A}_R)$  for all  $k$ . The following result reveals a connection between the global asymptotic stability of the dynamics of  $\underline{z}_k$  and the JSR of  $\mathcal{A}_R$ .

**Theorem 6.5:**

Consider the dynamics (6.23) with  $\underline{z}_k = \hat{\underline{\psi}}(\underline{\mathbf{Z}}_k)$  and  $\underline{\mathbf{Z}}_k$  and  $\hat{\underline{\psi}}(\cdot)$  according to (6.18) and (6.20), and let  $\mathcal{A}_R$  be given by (6.26). Then  $\lim_{k \rightarrow \infty} \underline{z}_k = \underline{z}$  for some  $\underline{z} \in \mathbb{R}^{(N+1)n_\psi^2}$  that is independent of  $\underline{z}_0$  if and only if  $\hat{\rho}(\mathcal{A}_R) < 1$ .

*Proof.* The proof of this result can be found in Appendix D.3.  $\square$

With these prerequisites, we are now in a position to establish a necessary and sufficient condition for the convergence of the second moment  $\underline{\Psi}_k$  according to (6.13) based on the JSR of  $\mathcal{A}_R$ .

**Corollary 6.1:**

Consider the polytopic MJLS (6.14) with second moment  $\underline{\Psi}_k = \mathbb{E} \left\{ \underline{\psi}_k \underline{\psi}_k^T \right\}$  and let  $\mathcal{A}_R$  be given by (6.26). Then  $\lim_{k \rightarrow \infty} \underline{\Psi}_k = \underline{\Psi}$  for some  $\underline{\Psi} \in \mathbb{R}^{n_\psi \times n_\psi}$  that is independent of  $\underline{\psi}_0$  and  $\theta_0$  if and only if  $\hat{\rho}(\mathcal{A}_R) < 1$ .

*Proof.* The mapping  $\hat{\underline{\psi}}(\cdot)$  is uniformly homeomorphic. Thus, by Theorem A.1, the sequence of tuples  $\{\underline{\mathbf{Z}}_k\}$  as introduced above in (6.18) converges to some  $\underline{\mathbf{Z}} = (\underline{\Psi}^{(0)}, \underline{\Psi}^{(1)}, \dots, \underline{\Psi}^{(N)}) \in \mathbb{H}^{n_\psi}$  whenever the sequence of vectors  $\{\underline{z}_k\}$  converges. Consequently,  $\lim_{k \rightarrow \infty} \underline{\Psi}_k^{(i)} = \underline{\Psi}^{(i)}$  for  $i = 0, 1, \dots, N$  and, by (6.17),  $\lim_{k \rightarrow \infty} \underline{\Psi}_k = \sum_{i=0}^N \underline{\Psi}^{(i)} = \underline{\Psi}$ .  $\square$

Corollary 6.1 implies that one of the two conditions for mean square stability, namely condition (6.4) in Definition 6.1, is equivalent to the condition  $\hat{\rho}(\mathcal{A}_R) < 1$ . The next result also associates the remaining condition (6.3) with the JSR of  $\mathcal{A}_R$ . An immediate consequence of the result is that the first moment converges to  $\underline{0}$  whenever the second moment converges.

**Theorem 6.6:**

Consider the polytopic MJLS (6.14) with first moment  $\hat{\underline{\psi}}_k = \mathbb{E}\{\underline{\psi}_k\}$  and let  $\mathcal{A}_R$  be given by (6.26). Then (6.12) holds with  $\underline{\psi} = \underline{0}$  whenever  $\hat{\rho}(\mathcal{A}_R) < 1$  holds, i.e., the first moment  $\hat{\underline{\psi}}_k$  converges to  $\underline{0}$  if the JSR of  $\mathcal{A}_R$  is less than one.

*Proof.* The proof is given in Appendix D.4. □

The combination of Theorems 6.5 and 6.6 yields the main result of this section and establishes the connection between the mean square stability of (6.14) and the JSR of the set  $\mathcal{A}_R$ .

**Corollary 6.2:**

The polytopic MJLS (6.14) is MSS according to Definition 6.1 if and only if  $\hat{\rho}(\mathcal{A}_R) < 1$ , where  $\mathcal{A}_R$  is given by (6.26).

**Remark 6.1:**

For an ordinary MJLS (4.1) with time-invariant mode transition matrix  $\mathbf{T}$ , the set  $\mathcal{A}_R$  contains only the element  $\mathbf{M} = \left(\mathbf{T}^T \otimes \mathbf{I}_{n_\psi}\right) \underline{\mathbf{A}}$ . Then, Corollary 6.2 reduces to the well-known result that mean square stability holds if and only if the spectral radius  $\rho(\mathbf{M})$  is smaller than one [127, 258].

As already mentioned in the introduction of this chapter, the condition  $\hat{\rho}(\mathcal{A}_R) < 1$  appearing in Corollary 6.2 is not new [244, 250, 251]. However, thus far, it has only been reported to be necessary and sufficient for the stability of polytopic MJLS (6.14) that are noise-free, i.e., for which it holds  $\underline{\tilde{w}}_k \equiv \underline{0}$ , or are subject to bounded and square-summable disturbances, i.e., for which it holds  $\sum_{k=0}^{\infty} \|\underline{\tilde{w}}_k\|^2 < \infty$ . Thus, our result constitutes a substantial theoretical contribution since it extends the applicability of the condition  $\hat{\rho}(\mathcal{A}_R) < 1$  to the broader class of polytopic MJLS that are subject to wide-sense stationary noise.

Unfortunately, the practical usefulness of Corollary 6.2 is limited because it is  $\mathcal{NP}$ -hard to decide whether  $\hat{\rho}(\mathcal{A}_R) < 1$  for any  $\mathcal{A}_R$  with at least two elements [250]. In fact, to determine if  $\hat{\rho}(\mathcal{M}) \leq 1$  is even undecidable for any nontrivial set of matrices  $\mathcal{M}$  [254]. In consequence, there cannot exist any polynomial time algorithm to decide whether (6.14) is MSS or not unless  $\mathcal{NP} = \mathcal{P}$ . However, Corollary 6.2 enables us to propose sufficient conditions for mean square stability and stabilizability in terms of LMI feasibility problems. LMI feasibility problems are convex and, as such, usually solved efficiently by state-of-the-art solvers in polynomial time [259]. The proposed conditions are presented in the next section.

To close this section, let us add that several other notions of stability exist in the literature, the most popular of which are referred to as stochastic stability and exponential mean square stability. Specifically for noise-free polytopic MJLS, these two are known to be equivalent to the notion of mean square stability considered in this thesis [250]. The numerical example provided in Section 6.5 strongly suggests that this also carries over to polytopic MJLS driven by wide-sense stationary noise. A formal verification, however, is yet to be brought forth.

## 6.4 Sufficient Conditions for Stability and Stabilizability

The first result of this section links mean square stability to the existence of positive definite solutions of a set of coupled Lyapunov-like equations. The key observation leading to this result is that such solutions, if existent, allow the construction of radially unbounded Lyapunov functions for

the dynamics (6.23) of  $\underline{z}_k$ . The global asymptotic stability of (6.23) then implies  $\hat{\rho}(\mathcal{A}_R) < 1$  (Theorem 6.5) and, hence, the mean square stability of the polytopic MJLS (6.14) (Corollary 6.2).

**Theorem 6.7:**

If there exist positive definite matrices  $\tilde{\mathbf{D}}^{(0)}, \tilde{\mathbf{D}}^{(1)}, \dots, \tilde{\mathbf{D}}^{(N)} \in \mathbb{R}^{n_\psi \times n_\psi}$  such that it holds

$$\tilde{\mathbf{D}}^{(i)} - \sum_{j=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)} \left( \tilde{\mathbf{A}}^{(i)} \right)^T \tilde{\mathbf{D}}^{(j)} \tilde{\mathbf{A}}^{(i)} \succ 0, \quad (6.27)$$

for  $i = 0, 1, \dots, N$  and any set of coefficients  $\alpha_k^{(1)}, \alpha_k^{(2)}, \dots, \alpha_k^{(R)} \geq 0$  satisfying  $\sum_{r=1}^R \alpha_k^{(r)} = 1$ , then the polytopic MJLS (6.14) is MSS.

*Proof.* The proof is provided in Appendix D.5. □

**Remark 6.2:**

For an ordinary MJLS with time-invariant mode transition probabilities  $t_{ij}$ , Theorem 6.7 reduces to the well-known result that mean square stability holds if

$$\tilde{\mathbf{D}}^{(i)} - \sum_{j=0}^N t_{ij} \left( \tilde{\mathbf{A}}^{(i)} \right)^T \tilde{\mathbf{D}}^{(j)} \tilde{\mathbf{A}}^{(i)} \succ 0, \quad (6.28)$$

for  $i = 0, 1, \dots, N$ . In stark contrast to (6.27), however, (6.28) is not only sufficient but also necessary [127, Theorem 3.9].

Theorem 6.7 cannot be evaluated in practice because it requires to determine the feasibility of infinitely many LMIs. However, it constitutes a paramount intermediate result on the way to a testable condition because (6.27) is equivalent to a finite set of LMIs. Roughly speaking, this equivalence stems from the fact that the set of positive definite matrices is convex so that it suffices that (6.27) is satisfied for MJLS with mode transition matrices restricted to the vertices  $\mathbf{A}^{(r)}$ , i.e., for which  $\mathbf{T}_k \in \mathcal{L}$  holds. More formally, this intuition comes to the fore in the following theorem.

**Theorem 6.8:**

The following two assertions are equivalent:

- (i) There exist positive definite matrices  $\tilde{\mathbf{D}}^{(0)}, \tilde{\mathbf{D}}^{(1)}, \dots, \tilde{\mathbf{D}}^{(N)} \in \mathbb{R}^{n_\psi \times n_\psi}$  such that (6.27) is satisfied for  $i = 0, 1, \dots, N$ .
- (ii) There exist positive definite matrices  $\mathbf{D}^{(0)}, \mathbf{D}^{(1)}, \dots, \mathbf{D}^{(N)} \in \mathbb{R}^{n_\psi \times n_\psi}$  and positive definite matrices  $\mathbf{E}^{(0)}, \mathbf{E}^{(1)}, \dots, \mathbf{E}^{(N)} \in \mathbb{R}^{n_\psi \times n_\psi}$  satisfying

$$\begin{bmatrix} 2\mathbf{E}^{(i)} - \mathbf{D}^{(i)} & \mathbf{E}^{(i)} \left( \tilde{\mathbf{A}}^{(i)} \right)^T \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right) \\ \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right)^T \tilde{\mathbf{A}}^{(i)} \mathbf{E}^{(i)} & \bar{\mathbf{D}} \end{bmatrix} \succ 0, \quad (6.29)$$

for  $i = 0, 1, \dots, N$ , and  $r = 1, 2, \dots, R$ , where

$$\underline{\lambda}^{(i,r)} = \left[ \sqrt{\lambda_{i0}^{(r)}} \quad \sqrt{\lambda_{i1}^{(r)}} \quad \dots \quad \sqrt{\lambda_{iN}^{(r)}} \right] \in \mathbb{R}^{1 \times N+1}, \quad (6.30)$$

$$\bar{\mathbf{D}} = \begin{bmatrix} \mathbf{D}^{(0)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^{(1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{D}^{(N)} \end{bmatrix} \in \mathbb{R}^{(N+1)n_\psi \times (N+1)n_\psi}. \quad (6.31)$$

*Proof.* The proof is given in Appendix D.6.  $\square$

Eq. (6.29) describes a set of  $R(N + 1)$  LMIs that are (usually) evaluated efficiently by interior-point methods [256, 260]. Similar sets of LMI conditions were derived in [245, 248], where the authors dealt with polytopic MJLS and MJLS with partly unknown transition probabilities, respectively. Our results are, however, less restrictive since the number of inequalities to be satisfied is smaller.

Theorem 6.8 enables us to determine the existence of a mode-independent state feedback law

$$\underline{U}_k = \underline{\pi}_k(\underline{\psi}_k) = \mathbf{L}\underline{\psi}_k, \quad (6.32)$$

that stabilizes the polytopic MJLS

$$\begin{aligned} \underline{\psi}_{k+1} &= \tilde{\mathbf{A}}^{(\theta_k)} \underline{\psi}_k + \tilde{\mathbf{B}}^{(\theta_k)} \underline{U}_k + \tilde{\underline{w}}_k, \\ \mathbf{T}_k &\in \text{conv}(\mathcal{L}), \end{aligned} \quad (6.33)$$

according to Definition 6.2. The chief result of this section is summarized by the corollary below.

**Corollary 6.3:**

*The polytopic MJLS (6.33) is mean square stabilizable by mode-independent state feedback (6.32) according to Definition 6.2 if there exist positive definite matrices  $\mathbf{D}^{(0)}, \mathbf{D}^{(1)}, \dots, \mathbf{D}^{(N)}, \mathbf{E} \in \mathbb{R}^{n_\psi \times n_\psi}$  and a matrix  $\mathbf{M} \in \mathbb{R}^{N n_u \times n_\psi}$  such that the LMIs*

$$\left[ \begin{array}{cc} 2\mathbf{E} - \mathbf{D}^{(i)} & \left( \tilde{\mathbf{A}}^{(i)} \mathbf{E} + \tilde{\mathbf{B}}^{(i)} \mathbf{M} \right)^\top \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right) \\ \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right)^\top \left( \tilde{\mathbf{A}}^{(i)} \mathbf{E} + \tilde{\mathbf{B}}^{(i)} \mathbf{M} \right) & \bar{\mathbf{D}} \end{array} \right] \succ 0, \quad (6.34)$$

are satisfied for  $i = 0, 1, \dots, N$ , and  $r = 1, 2, \dots, R$ , and where  $\underline{\lambda}^{(i,r)}$  and  $\bar{\mathbf{D}}$  are as per (6.30) and (6.31). In the affirmative case, the stabilizing controller gain is then given by  $\mathbf{L} = \mathbf{M}\mathbf{E}^{-1}$ .

*Proof.* Plugging (6.32) into (6.33) yields the closed-loop dynamics

$$\begin{aligned} \underline{\psi}_{k+1} &= \left( \tilde{\mathbf{A}}^{(\theta_k)} + \tilde{\mathbf{B}}^{(\theta_k)} \mathbf{L} \right) \underline{\psi}_k + \tilde{\underline{w}}_k, \\ \mathbf{T}_k &\in \text{conv}(\mathcal{L}), \end{aligned}$$

and condition (6.29) in Theorem 6.8 becomes

$$\left[ \begin{array}{cc} 2\mathbf{E}^{(i)} - \mathbf{D}^{(i)} & \left( \tilde{\mathbf{A}}^{(i)} \mathbf{E}^{(i)} + \tilde{\mathbf{B}}^{(i)} \mathbf{L} \mathbf{E}^{(i)} \right)^\top \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right) \\ \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right)^\top \left( \tilde{\mathbf{A}}^{(i)} \mathbf{E}^{(i)} + \tilde{\mathbf{B}}^{(i)} \mathbf{L} \mathbf{E}^{(i)} \right) & \bar{\mathbf{D}} \end{array} \right] \succ 0.$$

Eq. (6.34) then results from the change of variables  $\mathbf{M} = \mathbf{L}\mathbf{E}$  when we demand that

$$\mathbf{E} \stackrel{!}{=} \mathbf{E}^{(0)} \stackrel{!}{=} \mathbf{E}^{(1)} \stackrel{!}{=} \dots \stackrel{!}{=} \mathbf{E}^{(N)}.$$

$\square$

Before we illustrate the derived results in the next section, let us briefly discuss the role of the parameter  $\delta$ , which appears in the computation of the vertices  $\Lambda^{(r)}$  in Theorems 6.3 and 6.4. According to (6.7), the value of  $\delta$  is such that

$$t_{k,NN} = \mathbb{P}[\theta_{k+1} = N | \theta_k = N] \leq \delta < 1,$$

holds for all  $k$ , i.e., it is an upper bound for the probability of two successive applications of the default input  $\underline{u}_k^{\text{df}}$ . However, given that the mode transition probabilities are prone to modeling errors and updated once the CoCPN translator has updated its underlying probabilistic model of the packet delays and losses, a tight upper bound is hard to determine in advance. Hence,  $\delta$  can be interpreted as a hyperparameter to be set by the designer. A small value of  $\delta$  is appropriate if high packet delays and packet losses are assumed unlikely, but the resulting controller gain  $\mathbf{L}$  might no longer stabilize the plant if this assumption is violated. On the other hand, a large value of  $\delta$ , i.e., a value close to one, increases the robustness to successive packet losses and huge delays but might result in a controller that acts overly conservatively for most of the time during operation. One option to circumvent a tradeoff is to adapt  $\delta$  during operation once new information is available from the CoCPN translator. This is easily realized with a *gain scheduling* approach. Stabilizing controller gains  $\mathbf{L}_{\delta^{(i)}}$  are precomputed offline by means of Corollary 6.3 for a set of different values  $\delta^{(i)} \in [0, 1)$ ,  $i = 1, \dots, I$  for some  $I \in \mathbb{N}$ . During operation, once new information is available from the CoCPN translator,  $t_{k,NN}$  is first calculated as described in Chapter 4. Then, we select the controller gain as  $\mathbf{L} = \mathbf{L}_{\delta^{(j)}}$ , where

$$\delta^{(j)} = \min\{\delta^{(i)} \mid \delta^{(i)} \geq t_{k,NN}\},$$

which is then in use until  $t_{k,NN}$  is recalculated.

## 6.5 Numerical Example

In this section, we provide a numerical example to illustrate the results of the chapter by means of a simulation. For simplicity, we consider the canonical example of a double integrator plant. Denoting by  $q$  the horizontal position (in meters) of a body with mass  $m = 2$  kg, the continuous-time dynamics is of the form (3.2) and given by

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} w(t),$$

with state  $\underline{x}(t) = [q(t) \dot{q}(t)]^T$ , control input  $u(t)$  the applied force (in newtons), and where  $w(t)$  is a Gaussian, zero-mean white disturbance (in newtons) with intensity  $\mathbf{W}_c = 0.001 \text{ N}^2$ . Discretization with the sampling interval  $t_a = 0.01$  s during which the control input is assumed constant yields the corresponding discrete-time dynamics (3.1) with

$$\mathbf{A} = \begin{bmatrix} 1 & t_a \\ 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} t_a^2/m^2 \\ t_a/m \end{bmatrix}, \quad \mathbf{W} = 0.1 \begin{bmatrix} t_a^2/m^2 \\ t_a/m \end{bmatrix} \begin{bmatrix} t_a^2/m^2 \\ t_a/m \end{bmatrix}^T.$$

Our goal is to synthesize a stabilizing controller that transmits the control sequences to the plant over a network with unknown characteristics, i.e., the network model  $\tau_k^{\text{ca}}$  is not available. More specifically, we consider two scenarios, S1 and S2. In the first scenario, S1, the packet delay and loss probabilities are time-varying and independent over time as introduced in Section 3.2.1 and not known to the controller. In the second scenario, S2, the packet delays and losses are Markovian as introduced in Section 3.2.2 with time-varying transition probabilities that are unknown to the controller.

In both scenarios, we assume that control sequences of length  $N = 4$  are long enough to meet Assumption 3.3 so that the augmented dynamics (4.1) is a polytopic MJLS with five modes. According to Theorems 6.3 and 6.4, the resulting transition matrix polytopes have  $R_{S1} = 8$  and  $R_{S2} = 960$  vertices. With the aid of the JSR toolbox [261], we verify that  $1 \leq \hat{\rho}(\mathcal{A}_{R_{S1}}) \leq 1.186$  and  $1 \leq \hat{\rho}(\mathcal{A}_{R_{S2}}) \leq 1.03125$ , implying that the augmented dynamics is not MSS. This is expected because the plant dynamics taken by itself is not asymptotically stable.

However, letting  $\delta = 0.1$ , that is, assuming that  $P[\theta_{k+1} = 4 | \theta_k = 4] \leq 0.1$  holds for all  $k$ , we can invoke Corollary 6.3 to obtain the stabilizing gains

$$\mathbf{L}_{S1} = \begin{bmatrix} -31.371 & -180.336 & -1.065 & -11.840 & -15.885 & -0.798 & -11.034 & -0.766 \\ -0.186 & -0.411 & 0.021 & 0.026 & 0.029 & 0.016 & 0.023 & 0.016 \\ -0.174 & -0.385 & 0.020 & 0.022 & 0.027 & 0.014 & 0.019 & 0.014 \\ -0.192 & -0.403 & 0.020 & 0.022 & 0.026 & 0.015 & 0.018 & 0.014 \end{bmatrix},$$

$$\mathbf{L}_{S2} = \begin{bmatrix} -26.476 & -178.825 & -0.620 & -7.643 & -16.526 & -0.322 & -11.713 & -0.229 \\ -0.137 & -0.424 & 0.015 & 0.018 & 0.026 & 0.010 & 0.019 & 0.016 \\ -0.124 & -0.373 & 0.013 & 0.013 & 0.021 & 0.008 & 0.015 & 0.011 \\ -0.226 & -0.667 & 0.019 & 0.018 & 0.023 & 0.011 & 0.012 & 0.013 \end{bmatrix}.$$

To illustrate that  $\mathbf{L}_{S1}$  and  $\mathbf{L}_{S2}$  do indeed stabilize the plant, we carry out  $N_s = 5000$  simulation runs in each scenario. In each run, the plant is simulated over  $t_s = 100$  s, i.e., over  $K_s = 10\,000$  time steps, and the initial plant state is a random draw from a Gaussian distribution with mean and covariance

$$\hat{\underline{x}}_0 = \begin{bmatrix} 5 \\ 0 \end{bmatrix}, \quad \mathbf{X}_0 = 0.25 \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}.$$

To simulate a network with time-varying delay and loss probabilities (scenario S1), at each time step a probability distribution  $p_k$  is chosen by randomly selecting a row from the stochastic matrix  $\mathbf{S}$  given by (F.6) in Appendix F.2, according to which the actual realization  $\tau_k^{\text{ca}} \in \{0, 1, \dots, 4\}$  is then drawn. Note that for each row of  $\mathbf{S}$ , Assumption 3.3 is satisfied with  $N = 4$ . Moreover, computing the mode transition probabilities  $t_{k,ij}$  by means of Theorem 4.1 shows that  $P[\theta_{k+1} = 4 | \theta_k = 4] \leq \delta$  for each row. In scenario S2, we first randomly draw an initial condition  $\tau_0^{\text{ca}} \in \{0, 1, \dots, 5\}$  prior to each run. Then, at each time step, the transition matrix  $\mathbf{P}^{\text{ca}}$  is randomly picked from the ten candidates given in (F.7), according to which  $\tau_k^{\text{ca}}$  is then determined. The candidates are chosen such that the corresponding lumped dynamics for  $\theta_k$ , computed by virtue of Theorem 4.2, fulfills  $P[\theta_{k+1} = 4 | \theta_k = 4] \leq \delta$ .

After the simulation runs, we calculate an estimate of  $E\{\underline{x}_k\}$  and  $E\{\underline{x}_k \underline{x}_k^T\}$  in terms of the sample mean  $\bar{\underline{x}}_k$  and the sample second moment  $\bar{\mathbf{X}}_k$ , which are given by

$$\bar{\underline{x}}_k = \frac{1}{N_s} \sum_{n=1}^{N_s} \underline{x}_k^{[n]},$$

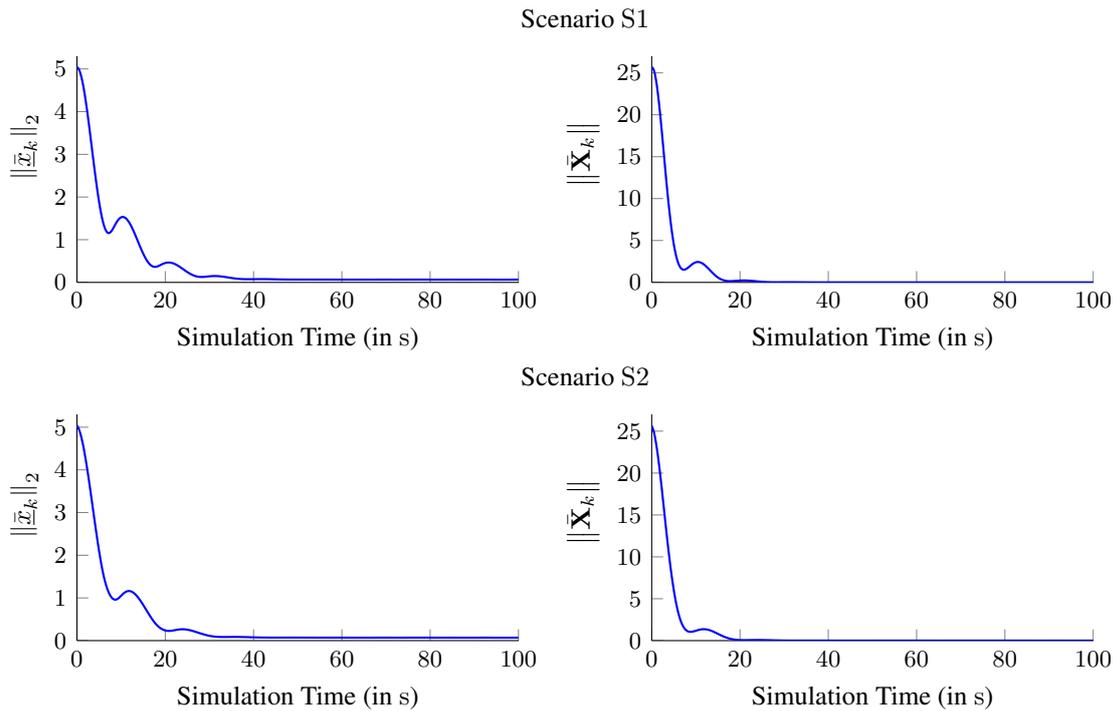
$$\bar{\mathbf{X}}_k = \frac{1}{N_s} \sum_{n=1}^{N_s} \underline{x}_k^{[n]} \left( \underline{x}_k^{[n]} \right)^T,$$

where  $\underline{x}_k^{[n]}$  denotes the plant state at time step  $k$  in run  $n$ .

The evolutions of  $\|\bar{\underline{x}}_k\|_2$  and  $\|\bar{\mathbf{X}}_k\|_F$  over time are depicted in Fig. 6.3 for both scenarios. Exemplary state trajectories from five simulation runs are shown in Fig. 6.4. The simulation results indicate that the closed-loop system is mean square stable. They even suggest exponential mean square stability.

## 6.6 Conclusions

This chapter was motivated by the observation that the network model  $\tau_k^{\text{ca}}$  is always an estimate of the “state” of the communication system because it bases on the monitoring data collected inside the communication system. Since the properties of  $\tau_k^{\text{ca}}$  determine the dynamics of the mode  $\theta_k$  of the MJLS (4.1), the mode transition probabilities are subject to modeling errors. Hence, the goal of this



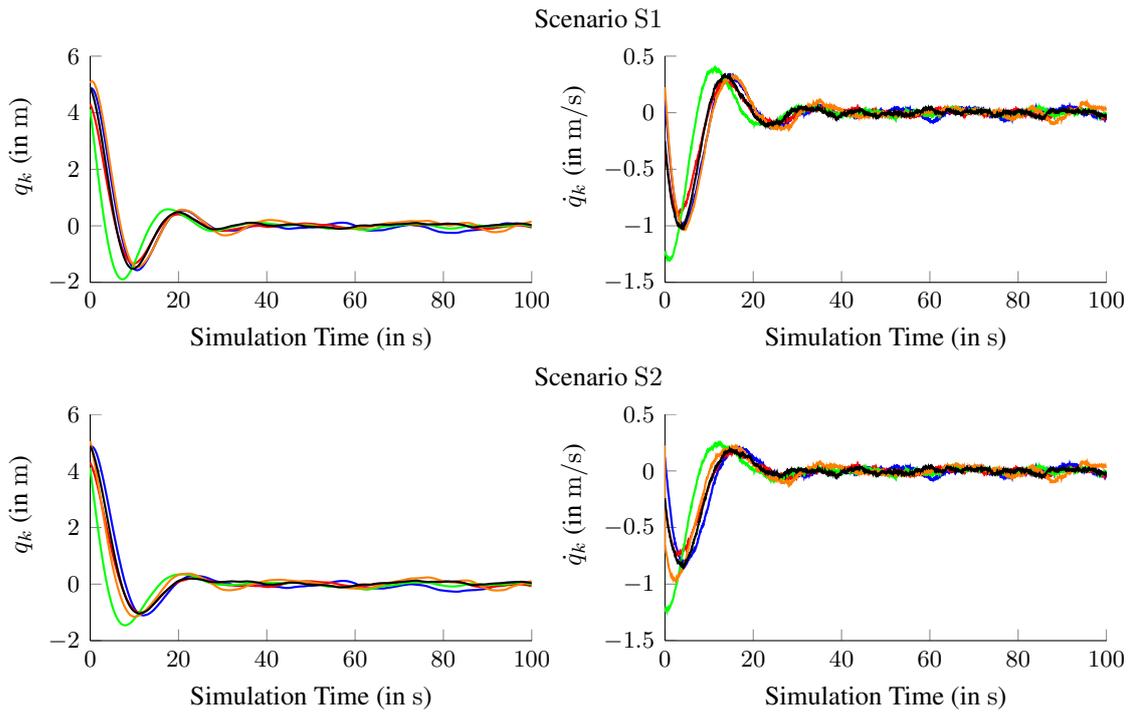
**Figure 6.3:** Norms of the sample mean  $\bar{x}_k$  (left) and the sample second moment  $\bar{X}_k$  (right) of the plant state over time in scenario S1 (top row) and scenario S2 (bottom row).

chapter was to investigate the impact of erroneous mode transition probabilities and to devise control algorithms that are robust to these modeling errors.

Our first result was the insight that an uncertain or even completely unknown network model  $\tau_k^{ca}$  renders the augmented dynamics (4.1) a polytopic MJLS, that is, an MJLS whose mode transition matrix is allowed to vary within a convex polytope spanned by a set of  $R$  vertices. Using the result from Chapter 4 that the mode transition matrix is always a lower Hessenberg matrix, we showed that the number of vertices solely depends on the control sequence length  $N$  for both independent and Markovian packet delays and losses. Then, we derived a necessary and sufficient condition for the mean square stability of arbitrary polytopic MJLS that are subject to wide-sense stationary noise. More specifically, we proved that mean square stability hinges on the question if the joint spectral radius of a particular set  $\mathcal{A}_R$  of  $R$  matrices, each of which associated with one vertex of the polytope, is less than one, i.e., one must determine if  $\hat{\rho}(\mathcal{A}_R) < 1$ . The key idea for the derivation of this condition was the construction of a vector-valued sequence that is convergent if and only if the second moment of the state is convergent. Although the condition  $\hat{\rho}(\mathcal{A}_R) < 1$  itself is not new, our result constitutes a substantial theoretical contribution because it extends its applicability to a broader class of systems.

On the other hand, the practical usefulness of the condition  $\hat{\rho}(\mathcal{A}_R) < 1$  is limited since it is  $\mathcal{NP}$ -hard to decide whether it is satisfied when  $R \geq 2$ . Aspiring after a more tractable solution, we decided to sacrifice necessity and then proposed a sufficient stability condition that consists in determining the feasibility of  $R(N+1)$  linear matrix inequalities, which is usually efficiently done by state-of-the-art solvers. Based on this condition, finally, we presented a similar set of linear matrix inequalities, whose feasibility implies the existence of a stabilizing mode-independent state feedback control law.

Next, in Chapter 7, we will integrate the results of this and the previous chapters into the CoCPN architecture. In simulation studies with CoCPN-Sim, we shall illustrate the applicability of CoCPN and demonstrate how the contributions of this thesis enable the reliable operation of control loops



**Figure 6.4:** Evolution of the plant states  $q_k$  (left) and  $\dot{q}_k$  (right) in five exemplary runs from scenario S1 (top row) and scenario S2 (bottom row).

under changing communication conditions, in the presence of cross traffic with different properties, and when links are highly utilized.

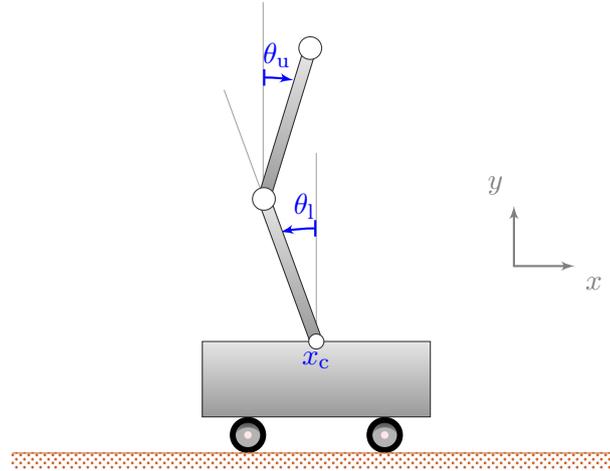
To conclude this chapter, we note that the number of vertices  $R$  grows faster than exponentially with the sequence length  $N$  in case the polytopic MJLS is constructed for Markovian packet delays and losses. For instance, we have  $R = 144$  for  $N = 3$ , but already  $R = 564\,480$  for  $N = 7$ . Depending on the available computing capacity, the resulting number of linear matrix inequalities to be evaluated might be prohibitively big even for small values of  $N$ . In such situations, we recommend constructing the polytope under the assumption of independent packet delays and losses, where the number of vertices is only  $R = 2N$ .

## Integration into the CoCPN Architecture

In the previous chapters, we concerned ourselves with the main research goal of this thesis – rendering control loops elastic, i.e., to make them flexible enough to manage changing communication conditions. For this purpose, we developed novel networked control algorithms which, supported by the data exchange realized by the CoCPN translator, allow sequence-based controllers to respond to changes of the network QoS so as to efficiently use the available communication resources. In the present chapter, we shall use CoCPN-Sim to illustrate how CoCPN and its cooperative concept work in two scenarios that simulate typical cases of application. In the first one, multiple control loops use the same network network but start and stop at different points in time, which leads to an overstretched link once a certain number of them is active concurrently. In the second one, a cloud-based controller must share the available communication capacity with unrelated traffic from other applications, which leads to rapid and unforeseeable changes of the communication conditions. Using the example of these two scenarios, we show how the cooperation between our algorithms and the congestion control CoCC enables CoCPN to realize a fair distribution of the available communication capacity without severe degradations of the network QoS. To that end, we first need to integrate our approaches into the CoCPN architecture. As described in Chapter 2, this demands implementing the CoCPN translator interface. In particular, we must develop an appropriate notion of QoC that meets the requirements of Definition 2.1 and, additionally, provide the mappings  $\text{rate} = r(\text{QoC})$  (cf. (2.1)) needed by the congestion control (cf. Algorithm 2.1).

We begin this chapter in Section 7.1 with a description of the double inverted pendulum on a cart, whose stabilization is the control task considered in all simulation runs. Then, in Section 7.2, we integrate all our proposed control algorithms into the CoCPN architecture by deriving the mappings  $\text{rate} = r(\text{QoC})$ . The required QoC calculation will be grounded on the average deviation from the upright position in steady-state operation. Finally, we present the two mentioned simulation scenarios in Section 7.3.

In all simulation runs carried out in this chapter, the CoCPN translator supplies the controllers with discrete probability distributions that model the occurring packet delays and losses (cf. Section 3.2.1). They are exchanged between OMNeT++ and Matlab via the hook function `ncs_doLoopStep` according to Algorithm 2.3. For simplicity, we do not compute the probability distributions based on collected monitoring data. Instead, we obtain them directly from histograms that we create based on



**Figure 7.1:** Sketch of the double inverted pendulum.

the recorded packet delays and losses. In all simulation runs, we use an Ethernet-based network. The most relevant parameters configured in OMNeT++/INET are listed in Table F.2 in Appendix F.3.2. Furthermore, unless otherwise noted, all network links are perfect, that is, no bit errors occur and all packets are transmitted with zero propagation delay.

In the following, we shall use the terms *MPC IMM* and *MPC Bound* to refer to the SMPC algorithms we developed in Sections 5.3 and 5.4. The controller resulting from the application of Corollary 6.3 will be referred to as *Robust Controller*.

## 7.1 Description of the Considered Control Task

Throughout this chapter, the control task we consider is to stabilize the double inverted pendulum on a cart that is illustrated in Fig. 7.1. Let  $x_c$ ,  $\theta_l$ , and  $\theta_u$  denote, respectively, the horizontal position of the cart (in meters) and the deviations (in radians) of the lower and the upper pendulum rod from the upright position, chosen such that positive values correspond to the clock-wise direction, and define the state  $\underline{x}(t) \in \mathbb{R}^6$  as

$$\underline{x}(t) = [x_c(t) \ \theta_l(t) \ \theta_u(t) \ \dot{x}_c(t) \ \dot{\theta}_l(t) \ \dot{\theta}_u(t)]^T .$$

Then, the derivation provided in Appendix E shows that the continuous-time nonlinear dynamics can be written as

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), u(t), \underline{w}(t)) , \quad (7.1)$$

with control input  $u(t)$  the force (in newtons) applied to move the cart,  $\underline{w}(t) = [w_c(t) \ w_l(t) \ w_u(t)]^T$  external disturbances acting on the cart ( $w_c(t)$ , in newtons) and on both pendulum tips ( $w_l(t)$  and  $w_u(t)$ , in newton-meters), and where the system function  $\underline{f}$  is given by (E.5).

With the parameters listed in Table F.1 in Appendix F.3 we can linearize (7.1) around the unstable upward equilibrium  $\underline{x}_e(t) = \underline{0}$  to obtain a linear dynamics of the form (3.2)

$$\dot{\underline{x}}(t) = \mathbf{A}_c \underline{x}(t) + \mathbf{B}_c u(t) + \mathbf{G}_c \underline{w}(t) ,$$

with  $\mathbf{A}_c$ ,  $\mathbf{B}_c$ , and  $\mathbf{G}_c$  as per (E.6). Assuming that  $\underline{w}(t)$  is a Gaussian, zero-mean white disturbance with intensity

$$\mathbf{W}_c = 1 \cdot 10^{-11} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} ,$$



**Figure 7.2:** Basic network topology used to obtain the mapping rate  $= r(\text{QoC})$  provided by the CoCPN translator.

we obtain the discrete-time dynamics (3.1) with  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{W}$  according to (3.3). Furthermore, we set

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{V} = 1 \cdot 10^{-8} \mathbf{I}_3,$$

that is, a sensor takes noisy measurements of the cart position, the deviation of the lower pendulum rod, and the difference between the deviations of the two pendulum rods. Since we are interested in the impact of the packet delays and losses, both  $\mathbf{W}_c$  and  $\mathbf{V}$  are chosen small so that the influence of the noise on the control performance is kept low.

In all simulation runs carried out in this chapter, the initial plant state  $\underline{x}_0$  is randomly drawn from a Gaussian distribution with mean and covariance

$$\hat{\underline{x}}_0 = \underline{0}, \quad \mathbf{X}_0 = 1 \cdot 10^{-6} \mathbf{I}_6. \quad (7.2)$$

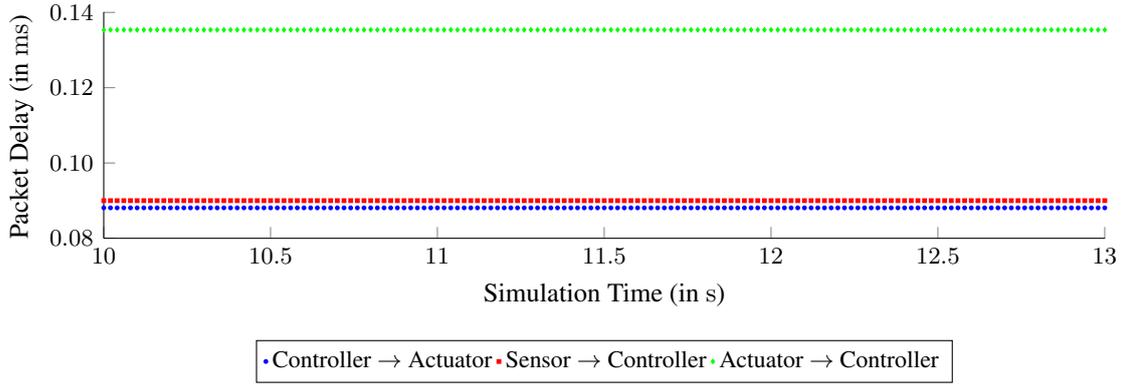
To simulate the plant, we numerically integrate the nonlinear differential equation (7.1). For the integration, we use fixed intervals of length 0.001 s, i.e., we use a sampling rate of 1 kHz to simulate the plant. Note that the controllers operate at a lower rate  $f_a$ , which, in all simulation runs will lie between  $f_{a_{\min}} = 50$  Hz and  $f_{a_{\max}} = 200$  Hz. Thus, control sequences will be computed every  $t_a = 1/f_a$  seconds.

## 7.2 Implementation of the CoCPN Translator Interface

To use the approaches developed in Chapters 5 and 6 in CoCPN, we must implement the CoCPN translator interface as described in Chapter 2. In particular, we need to define the notion of QoC for the control task at hand so that we can derive the mappings rate  $= r(\text{QoC})$ . For this purpose, we compare the control performance that is achieved with different sampling rates  $f_a$  between  $f_{a_{\min}} = 50$  Hz and  $f_{a_{\max}} = 200$  Hz. To exclude performance fluctuations due to changing communication conditions, we consider a simulation setup with a single NCS that exclusively uses the communication resources.

The network topology used in this section is depicted in Fig. 7.2. The actuator and the sensor are physically co-located and thus connected to the same router. The controller is connected to this network via a second router. For their communication, actuator, sensor, and controller use UDP and IPv6 with the parameters listed in Table F.2. As Fig. 7.3 illustrates, this setup results in end-to-end delays in the sub-millisecond range. Furthermore, no packet losses occur. Hence, the impact of the communication system on the control performance is negligible. Differences in the achieved performance solely originate from better disturbance rejection that is possible with higher sampling rates.

To quantify the relationship between control performance and sampling rate, we conduct several Monte Carlo simulations with the *MPC IMM*, the *MPC Bound*, and the *Robust Controller*, respectively. In the simulations, we gradually lower the sampling rate from  $f_a = f_{a_{\max}}$  to  $f_a = f_{a_{\min}}$  in steps of five hertz, i.e., we consider the 31 different sampling rates 200 Hz, 195 Hz, 190 Hz,  $\dots$ , 50 Hz. We carry out  $N_s = 50$  simulation runs with each sampling rate. In each run, the NCS is simulated over  $t_s = 30$  s, and the initial plant state  $\underline{x}_0$  is randomly drawn from a Gaussian distribution with mean and covariance given by (7.2). The *MPC IMM*, the *MPC Bound*, and the *Robust Controller* are configured as described



**Figure 7.3:** Excerpt of the recorded end-to-end delays with the basic network topology shown in Fig. 7.2.

in Appendix F.3.3. Furthermore, in view of the negligibly small end-to-end delays, we set  $N = 2$  and  $L = 1$  in all runs.<sup>1</sup>

To measure the achieved control performance, we denote by  $\underline{x}_k^{[n]}(f_a)$  the plant state at time step  $k$  in the  $n$ -th simulation run with sampling rate  $f_a$ , where  $k$  is with respect to the plant sampling rate. As described in Section 7.1, we numerically integrate the continuous-time plant dynamics (7.1) using fixed integration intervals of length 0.001 s. Hence, with  $t_s = 30$  s, we have  $k \in \{0, 1, \dots, 30\,000\}$ . Then, the sample mean of the norm of the deviation of the plant state from the upward equilibrium  $\underline{x}_e = \underline{0}$  at time  $k$  is given by

$$\begin{aligned} \bar{e}_k(f_a) &= \frac{1}{N_s} \sum_{n=1}^{N_s} \left\| \underline{x}_k^{[n]}(f_a) - \underline{x}_e \right\|_2 \\ &= \frac{1}{N_s} \sum_{n=1}^{N_s} \left\| \underline{x}_k^{[n]}(f_a) \right\|_2. \end{aligned} \quad (7.3)$$

The evolutions of  $\bar{e}_k$  over time for some of the considered sampling rates are depicted in the top rows of Figs. 7.4 to 7.6 for the *MPC IMM*, the *MPC Bound*, and the *Robust Controller*, respectively. For all control algorithms,  $\bar{e}_k$  decreases as  $f_a$  increases, which implies that the achievable control performance increases with  $f_a$ . This result nicely illustrates the idea behind CoCPN and its cooperative concept – the sharing of the available communication resources can be improved by adjusting the desired control performance. The curves also indicate that the performance gain decreases as the sampling rate approaches  $f_{a_{\max}} = 200$  Hz, so that increasing it further would only marginally raise the achievable performance.

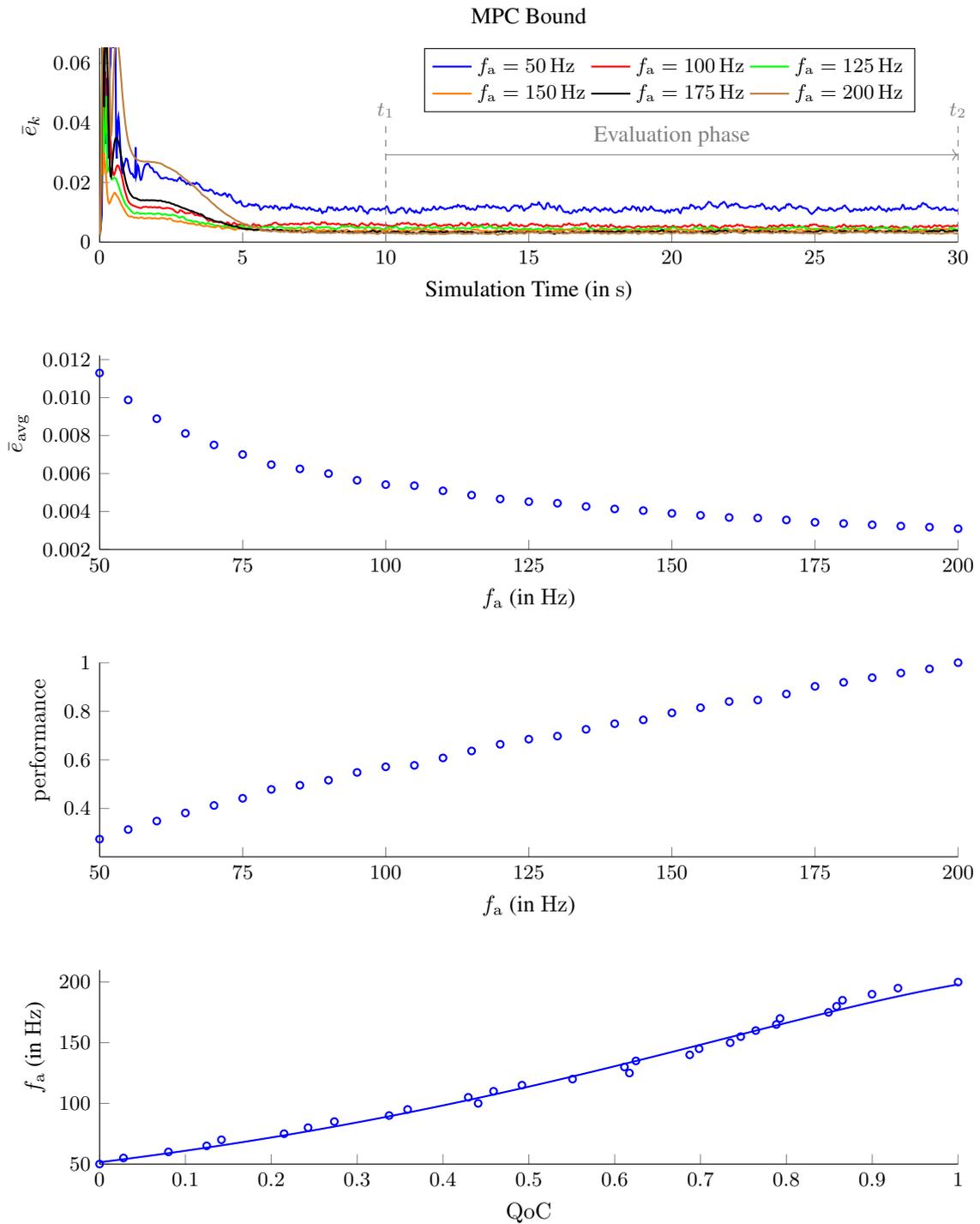
To express the relationships between control performance and sampling rate in the form of mappings performance =  $p(f_a)$  as required by Algorithm 2.1, we first compute the time average of  $\bar{e}_k$  for each  $f_a$  according to

$$\bar{e}_{\text{avg}}(f_a) = \frac{1}{k_2 - k_1 + 1} \sum_{k=k_1}^{k_2} \bar{e}_k(f_a), \quad (7.4)$$

with  $k_1 = 10\,001$ , which corresponds to  $t_1 = 10.001$  s and  $k_2 = 30\,000$ , which corresponds to  $t_2 = t_s = 30$  s. That is, we interpret the first 10 s of each simulation run as transient response to the initial excitation of the plant, where the controller attempts to drive the state back to the upward

<sup>1</sup>Even with  $f_a = f_{a_{\max}} = 200$  Hz, the controller's sampling period is  $t_a = 5$  ms, whereas the packet delays are in the sub-millisecond range. Hence, any transmitted packet is available at the next sampling instant.



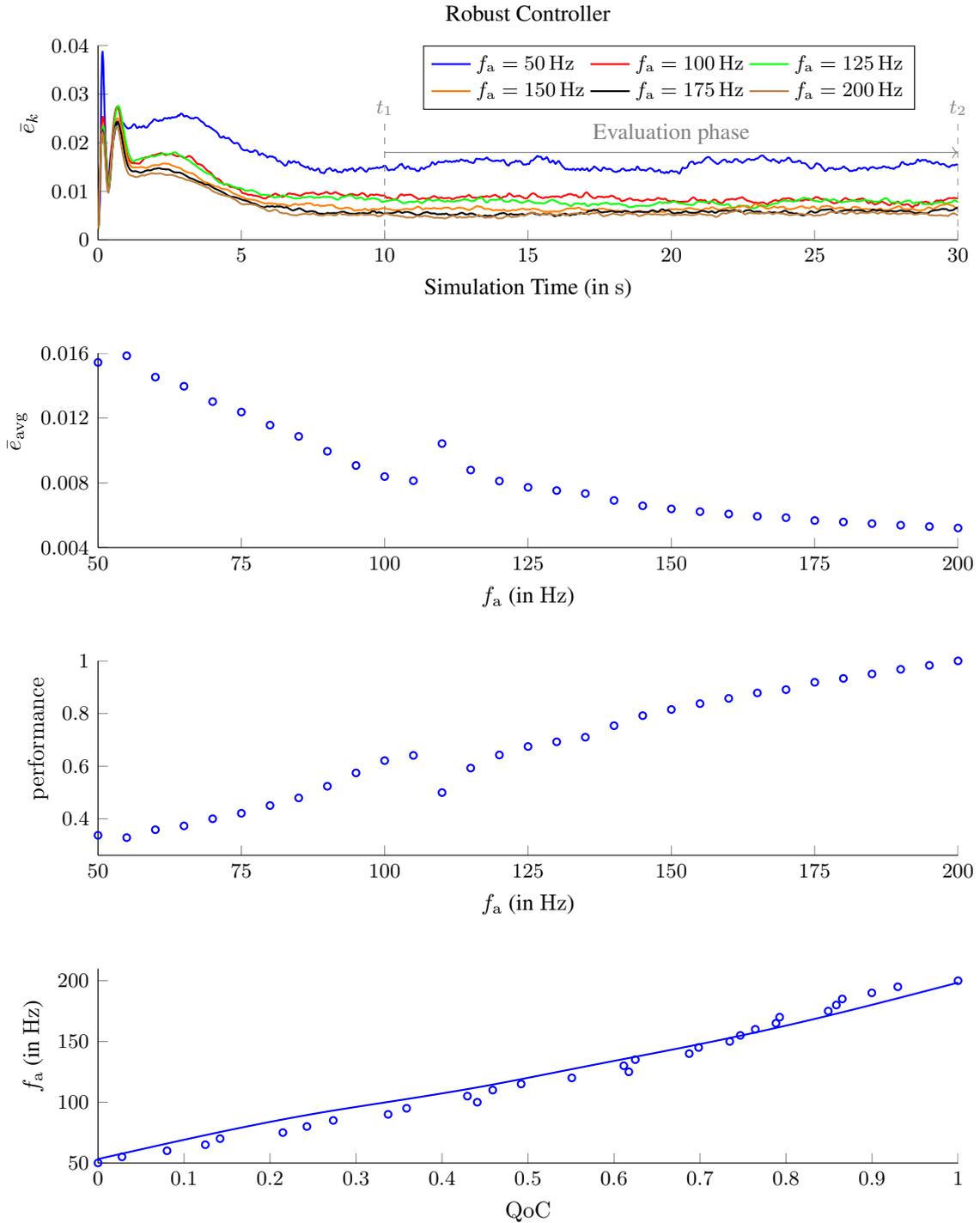


**Figure 7.5:** Derivation of the mapping  $f_a = f(\text{QoC})$  (bottom row) based on  $\bar{e}_k$  (7.3) (top row),  $\bar{e}_{\text{avg}}$  (7.4) (second row), and performance (7.5) (third row) for the *MPC Bound*.

Based on  $\bar{e}_{\text{avg}}$ , we now define the relative performance measure

$$\text{performance} = p(f_a) \triangleq \frac{\bar{e}_{\text{avg}}(f_{a_{\text{max}}})}{\bar{e}_{\text{avg}}(f_a)} \in (0, 1]. \quad (7.5)$$

For  $f_{a_{\text{max}}}$  we have performance = 1, whereas for  $f_a < f_{a_{\text{max}}}$ , we always have performance < 1. Thus,  $p(f_a)$  quantifies the “amount” of control performance that remains when the controller lowers its



**Figure 7.6:** Derivation of the mapping  $f_a = f(\text{QoC})$  (bottom row) based on  $\bar{e}_k$  (7.3) (top row),  $\bar{e}_{\text{avg}}$  (7.4) (second row), and performance (7.5) (third row) for the *Robust Controller*.

sampling rate from  $f_{a_{\text{max}}}$  to  $f_a$ . For all considered sampling rates, performance computed according to (7.5) is plotted in the third rows of Figs. 7.4 to 7.6.

Using (7.5), an appropriate notion of QoC is obtained by virtue of the transformation

$$\text{QoC} \triangleq \frac{p(f_a) - p(f_{a_{\text{min}}})}{1 - p(f_{a_{\text{min}}})}, \tag{7.6}$$

**Algorithm 7.1** Implementation of the  $\oplus$  Interface: Controller  $\rightarrow$  Communication System**Input:** 31 data pairs  $(f_a, \bar{e}_{avg})$  which  $\bar{e}_{avg}$  as per (7.4)**Output:**  $rate = r(QoC)$  // Strictly increasing mapping

- 1: Translate  $\bar{e}_{avg}$  into QoC using (7.5) and (7.6)
- 2: Compute mapping  $f_a = f(QoC)$  by fitting a suitable function to the pairs  $(QoC, f_a)$
- 3: Determine packet size  $s$  (in bits) // Depends on control sequence length
- 4: Obtain mapping  $rate = r(QoC)$  using (7.7)
- 5: **return**  $r$

**Algorithm 7.2** Implementation of the  $\oplus$  Interface: Communication System  $\rightarrow$  Controller**Input:** Monitoring Data,  $QoC_{target}$ **Output:** Network Model, Target Sampling Rate  $f_{atarget}$ 

- 1: Translate monitoring data into network model // As formalized in Section 3.2
- 2: Evaluate  $f(QoC_{target})$  to obtain  $f_{atarget}$  with  $f$  given by (F.8), (F.10), or (F.12)
- 3: **return** Network model,  $f_{atarget}$

as we now have  $QoC = 1$  for  $f_{a_{max}}$  and  $QoC = 0$  for  $f_{a_{min}}$ . We then find continuous, smooth, and strictly increasing mappings  $f_a = f(QoC)$  by fitting suitable functions to the 31 pairs  $(QoC, f_a)$ , where  $QoC$  is computed according to (7.6). They are depicted in the bottom rows, respectively, of Figs. 7.4 to 7.6 for the *MPC IMM*, the *MPC Bound*, and the *Robust Controller*. Their expressions are given by (F.8), (F.10), and (F.12) in Appendix F.3.3. Note that these mappings are the inverses of the mappings defined by (7.6). Finally, the mappings  $rate = r(QoC)$  with  $rate$  in bit/s, which are returned by Algorithm 2.1 and required by CoCC, are immediately obtained by multiplication of  $f(QoC)$  with the packet size, which depends on the employed control sequence length. That is, we simply need to compute

$$rate = r(QoC) \triangleq sf(QoC), \quad (7.7)$$

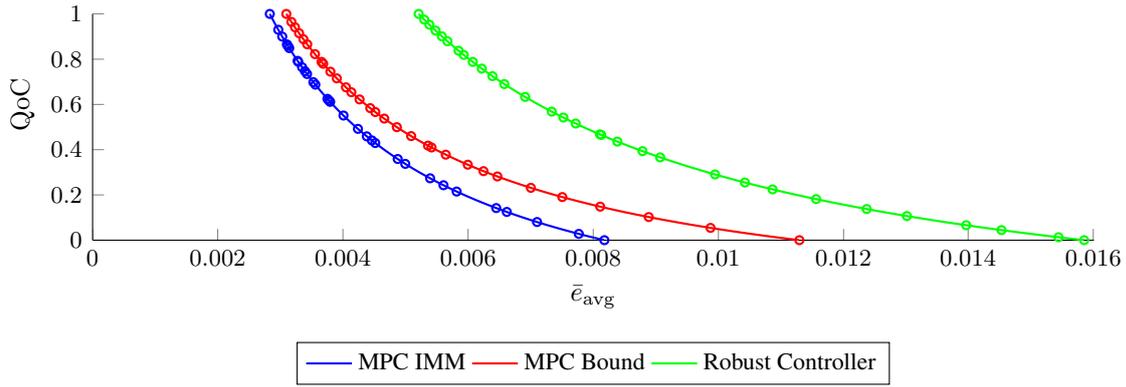
with packet size  $s$  in bits.

The resulting implementations of the CoCPN translator interface are summarized in Algorithm 7.1 (controller  $\rightarrow$  communication system) and Algorithm 7.2 (communication system  $\rightarrow$  controller).

### 7.3 Simulation Scenarios

In this section, we demonstrate the CoCPN architecture in two different simulation scenarios. As outlined at the beginning of this chapter, we focus on its cooperative approach for a fair distribution of the available communication capacity. In particular, we illustrate how the cooperation between our proposed SMPC algorithms and the congestion control CoCC, backed by the CoCPN translator, enables a reliable operation without degradations of the network QoS. In the first simulation scenario, presented in Section 7.3.1, we consider multiple NCS that start and stop at different points in time during the simulation. The communication resources have to be shared with UDP traffic from an unrelated application, which results in a highly utilized bottleneck link. In the second simulation scenario, presented in Section 7.3.2, we are concerned with a single NCS where the controller is located in the cloud and connected to the sensor and the actuator via multiple routers. Co-existing mixed UDP and TCP cross traffic then leads to rapidly changing communication conditions and to end-to-end delays of up to one hundred fifty milliseconds.

Recall from Chapter 2 that CoCPN seeks to implement fairness by computing the same target QoC for all NCS (cf. (2.2)). A fair sharing of the communication capacity with respect to QoC, however, only results when the NCS actually achieve and maintain the given target QoC. Yet, in the previous section



**Figure 7.7:** Mappings  $\text{QoC} = q(\bar{e}_{\text{avg}})$  for the *MPC IMM*, the *MPC Bound*, and the *Robust Controller*.

we defined QoC as a function of  $f_a$ . Hence, we can only obtain the achievable QoC for a given sampling rate by evaluating (7.6).<sup>2</sup> Thus, to verify that CoCPN indeed achieves a fair sharing of the communication resources, we must extend our notion of QoC, so that we can meaningfully quantify the control performance at any time  $k$ , where  $k$  is with respect to the plant sampling rate. That is, we need a mapping that directly translates  $\bar{e}_k$  into QoC, where we compute  $\bar{e}_k$  according to (7.3), i.e.,  $\bar{e}_k$  is the sample mean of the norm of the plant state

$$\bar{e}_k = \frac{1}{N_s} \sum_{n=1}^{N_s} \left\| \underline{x}_k^{[n]} \right\|_2, \quad (7.8)$$

with  $\underline{x}_k^{[n]}$  the plant state at time step  $k$  in the  $n$ -th simulation run.

To obtain this mapping, we first plug (7.5) into (7.6), which shows that for a given  $f_a \in [f_{a_{\min}}, f_{a_{\max}}]$ , the relationship between QoC and  $\bar{e}_{\text{avg}}(f_a)$  is expressed as

$$\begin{aligned} \text{QoC} &= \frac{p(f_a) - p(f_{a_{\min}})}{1 - p(f_{a_{\min}})} \\ &= \left( \frac{\bar{e}_{\text{avg}}(f_{a_{\max}})}{\bar{e}_{\text{avg}}(f_a)} - \frac{\bar{e}_{\text{avg}}(f_{a_{\max}})}{\bar{e}_{\text{avg}}(f_{a_{\min}})} \right) \left( 1 - \frac{\bar{e}_{\text{avg}}(f_{a_{\max}})}{\bar{e}_{\text{avg}}(f_{a_{\min}})} \right)^{-1}, \end{aligned}$$

with  $\bar{e}_{\text{avg}}(f_a)$  as per (7.4). Omitting the dependency on  $f_a$ , we get a function  $q$  that directly maps  $\bar{e}_{\text{avg}}$  onto QoC according to

$$\text{QoC} = q(\bar{e}_{\text{avg}}) \triangleq \frac{a}{\bar{e}_{\text{avg}}} - \frac{a}{\bar{e}_{\text{avg}}(f_{a_{\min}})}, \quad (7.9)$$

with  $a$  given by

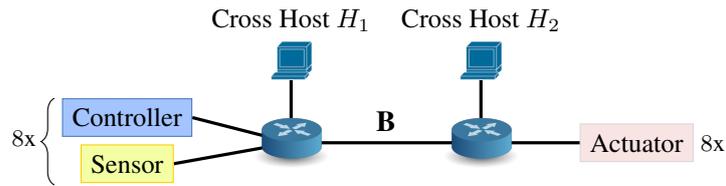
$$a = \bar{e}_{\text{avg}}(f_{a_{\max}}) \left( 1 - \frac{\bar{e}_{\text{avg}}(f_{a_{\max}})}{\bar{e}_{\text{avg}}(f_{a_{\min}})} \right)^{-1}.$$

The values for  $\bar{e}_{\text{avg}}(f_{a_{\min}})$  and  $\bar{e}_{\text{avg}}(f_{a_{\max}})$  obtained in the previous section lead to the functions given by (F.9), (F.11), and (F.13) for the *MPC IMM*, the *MPC Bound*, and the *Robust Controller*. They are plotted in Fig. 7.7.

Based on (7.9), we then define the achieved QoC at time  $k$  as

$$\text{QoC}_k \triangleq q(\bar{e}_k), \quad (7.10)$$

<sup>2</sup>Or for a given data rate rate by evaluating the inverse of  $r$ .



**Figure 7.8:** The network topology considered in the first simulation scenario.

**Table 7.1:** Configuration of the eight NCS in the first simulation scenario.

NCS	Control Algorithm	Time of Operation
NCS 1	MPC IMM	0 s to 450 s
NCS 2	MPC Bound	30 s to 90 s
NCS 3	Robust Controller	130 s to 300 s
NCS 4	MPC Bound	130 s to 250 s
NCS 5	MPC Bound	170 s to 280 s
NCS 6	MPC IMM	190 s to 350 s
NCS 7	MPC IMM	240 s to 410 s
NCS 8	Robust Controller	300 s to 350 s

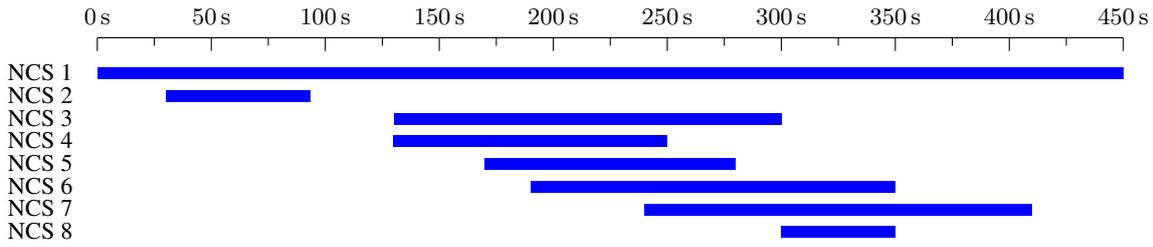
with  $\bar{e}_k$  as introduced above in (7.8). That is, we directly use the sample mean  $\bar{e}_k$  in  $q$  rather than its time average  $\bar{e}_{\text{avg}}$ . This decision is motivated by the results from the previous section, where significant fluctuations of  $\bar{e}_k$  only occurred during the transient response of the NCS to the initial excitation of the plant. Hence, once the NCS is in steady-state operation, we expect only slight gaps between  $\bar{e}_k$  and  $\bar{e}_{\text{avg}}$ .

### 7.3.1 Multiple NCS with UDP Cross Traffic

In the first simulation scenario, we consider a CPS with eight independent NCS that use the network shown in Fig. 7.8. The controller and the sensor of each NCS are connected to the same router. The actuators are connected to this network via a second router. All communication within the individual NCS is based on UDP and IPv6 with the parameters listed in Table F.2. The NCS share the network with two hosts  $H_1$  and  $H_2$ , which we use to model the integration of higher-level applications for, e.g., data visualization.  $H_1$  communicates with  $H_2$  at a fixed rate of 8 Mbit/s via UDP, where each transmitted packet is 400 B. The link between the two routers, denoted by **B** in Fig. 7.8, constitutes a bottleneck because it only provides a data rate of 10 Mbit/s, whereas all others provide 100 Mbit/s.

We deploy the *MPC IMM*, the *MPC Bound*, and the *Robust Controller* as listed in Table 7.1. They are configured as described in Appendix F.3.3. We simulate the CPS over  $t_s = 450$  s, during which the eight NCS start and stop at different points in time. The start and stop times are given in the rightmost column of Table 7.1 and the corresponding times of operation are depicted in Fig. 7.9. They are chosen such that up to six NCS operate concurrently. However, due to the co-existing traffic from  $H_1$  to  $H_2$ , the bottleneck link **B** becomes utilized to the full with already three NCS being active at the same time. Thus, in order to avoid over-utilization of the bottleneck link, the congestion control CoCC must distribute its capacity among the NCS by appropriately adjusting the target performance  $\text{QoC}_{\text{target}}$  as described in Chapter 2. The CoCPN translator then extracts the corresponding target sampling rate  $f_{\text{a,target}}$  by means of Algorithm 7.2, allowing each controller to alter its sampling rate accordingly.

Fig. 7.10 shows the evolution of  $\text{QoC}_{\text{target}}$  over the simulation time. At the beginning, CoCC can keep the target QoC maximal ( $\text{QoC}_{\text{target}} = 1$ ), since only NCS 1 and NCS 2 are active. Then, starting at



**Figure 7.9:** The times of operation of the NCS in the first simulation scenario.

$t_1 = 170$  s, CoCC must gradually lower the target QoC as more NCS begin to operate. The minimum value is reached between  $t_2 = 240$  s and  $t_3 = 250$  s, when six NCS, namely NCS 1 and NCS 3 to NCS 7, are active. After that, the number of active NCS decreases, so that the target QoC gradually rises again. Finally, it spikes back to the peak value ( $\text{QoC}_{\text{target}} = 1$ ) at  $t_4 = 350$  s when NCS 6 and NCS 8 stop, leaving only NCS 1 and NCS 7 active. For some of the NCS, the corresponding controller sampling rates  $f_{\text{a}_{\text{target}}}$  are plotted in Fig. 7.11. Although the target QoC is the same for all NCS, the sampling rates, and, hence, the resulting data rates differ due to the different control algorithms.

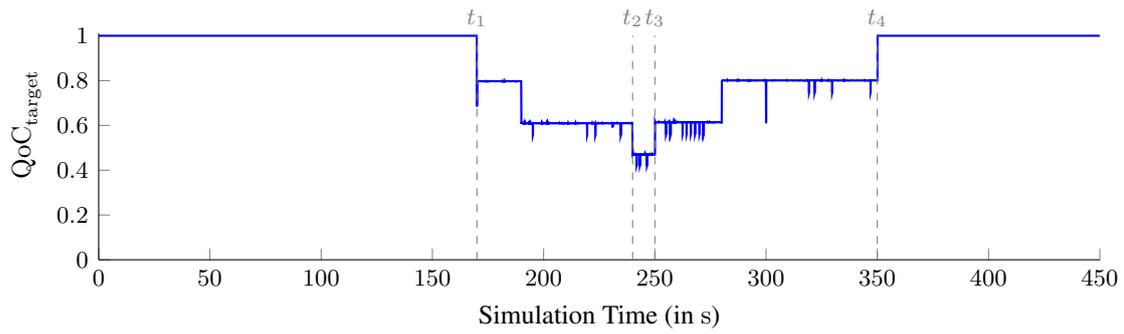
By adjusting the desired target QoC, CoCPN is able to maintain a high utilization of the bottleneck link without overstressing its capacity in spite of the co-existing cross traffic. Thereby, congestion and potentially harmful degradations of the network QoS are avoided. Latency is kept low and, as illustrated in Fig. 7.12, the packets sent from the controllers to the actuators (i.e., the control sequences) only experience end-to-end delays in the single-digit millisecond range even when six NCS operate concurrently between  $t_2$  and  $t_3$ . Note that the cross traffic is unidirectional (from host  $H_1$  to host  $H_2$ ) and, therefore, neither affects the packets sent from the sensors to the controllers (i.e., the measurements) nor the packets sent from the actuators to the controllers (i.e., the ACKs). Thus, their experienced end-to-end delays are smaller and do not disperse much.

To evaluate whether all NCS achieve the desired target QoC, we carry out  $N_s = 200$  simulation runs. In each run, we randomly draw the initial plant state  $\underline{x}_0$  from a Gaussian distribution with mean and covariance given by (7.2). Afterwards, we determine  $\bar{e}_k$  according to (7.8). The evolutions of  $\bar{e}_k$  over time are plotted in Fig. 7.13, together with the corresponding 1-sigma intervals  $\bar{e}_k \pm \sigma_k$ , where the standard deviation  $\sigma_k$  is given by

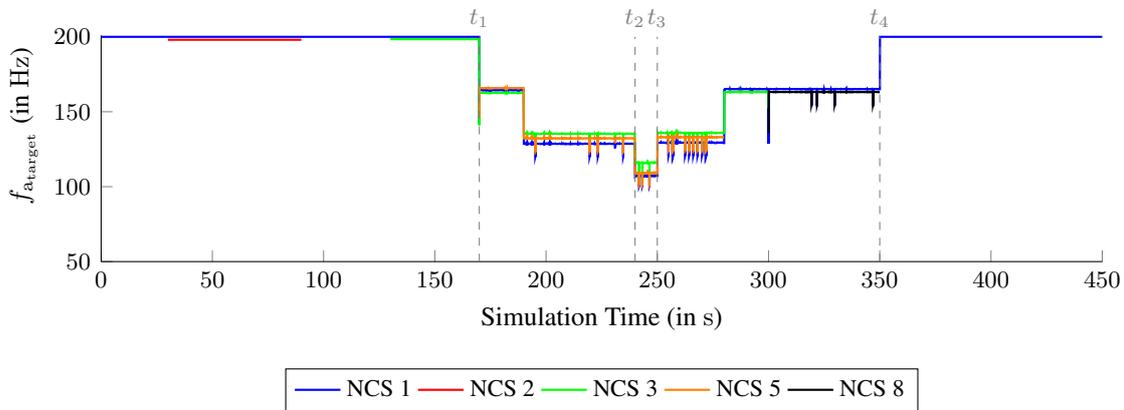
$$\sigma_k = \sqrt{\frac{1}{N_s - 1} \sum_{n=1}^{N_s} \left( \left\| \underline{x}_k^{[n]} \right\|_2 - \bar{e}_k \right)^2}. \quad (7.11)$$

In Fig. 7.13, the adjustments of the target QoC made during operation according to Fig. 7.10 are clearly recognizable. Let us exemplarily consider its top left plot, which shows  $\bar{e}_k$  and  $\bar{e}_k \pm \sigma_k$  over time for NCS 1 that is active from the beginning to the end of the simulation time. After the transient phase (due to the initial excitation of the plant),  $\bar{e}_k$  remains relatively constant until  $t_1 = 170$  s, when CoCC starts to lower the target QoC. Subsequently,  $\bar{e}_k$  reaches its peak value between  $t_2 = 240$  s and  $t_3 = 250$  s, during which the target QoC is minimal. After that,  $\bar{e}_k$  gradually peaks off and, once the target QoC jumps to its maximum again at  $t_4 = 350$  s, eventually returns to the range of values it attained prior to  $t_1$ . Upper and lower boundary of the corresponding 1-sigma interval exhibit similar behavior. Moreover, the width of the 1-sigma interval increases as  $\bar{e}_k$  increases. We can draw alike conclusions for the remaining NCS (NCS 2 to NCS 8) from the other plots in Fig. 7.13.

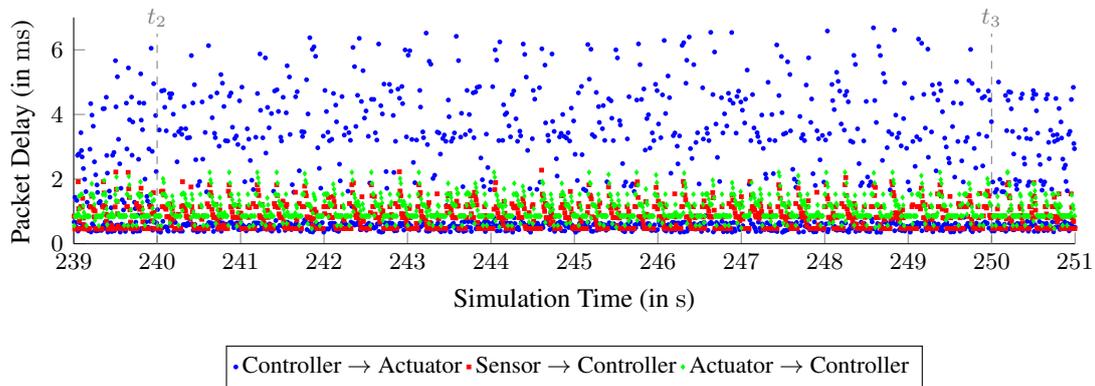
Based on  $\bar{e}_k$ , we then compute  $\text{QoC}_k = 1$  by virtue of (7.10). We set  $\text{QoC}_k = 1$  in case  $q(\bar{e}_k) > 1$  to ensure that  $\text{QoC}_k \in [0, 1]$  for all  $k$ . The eight plots in Fig. 7.14 depict the evolutions of  $\text{QoC}_k$  over



**Figure 7.10:** Evolution of  $QoC_{target}$  computed by the congestion control CoCC for all NCS over the simulation time in the first simulation scenario.

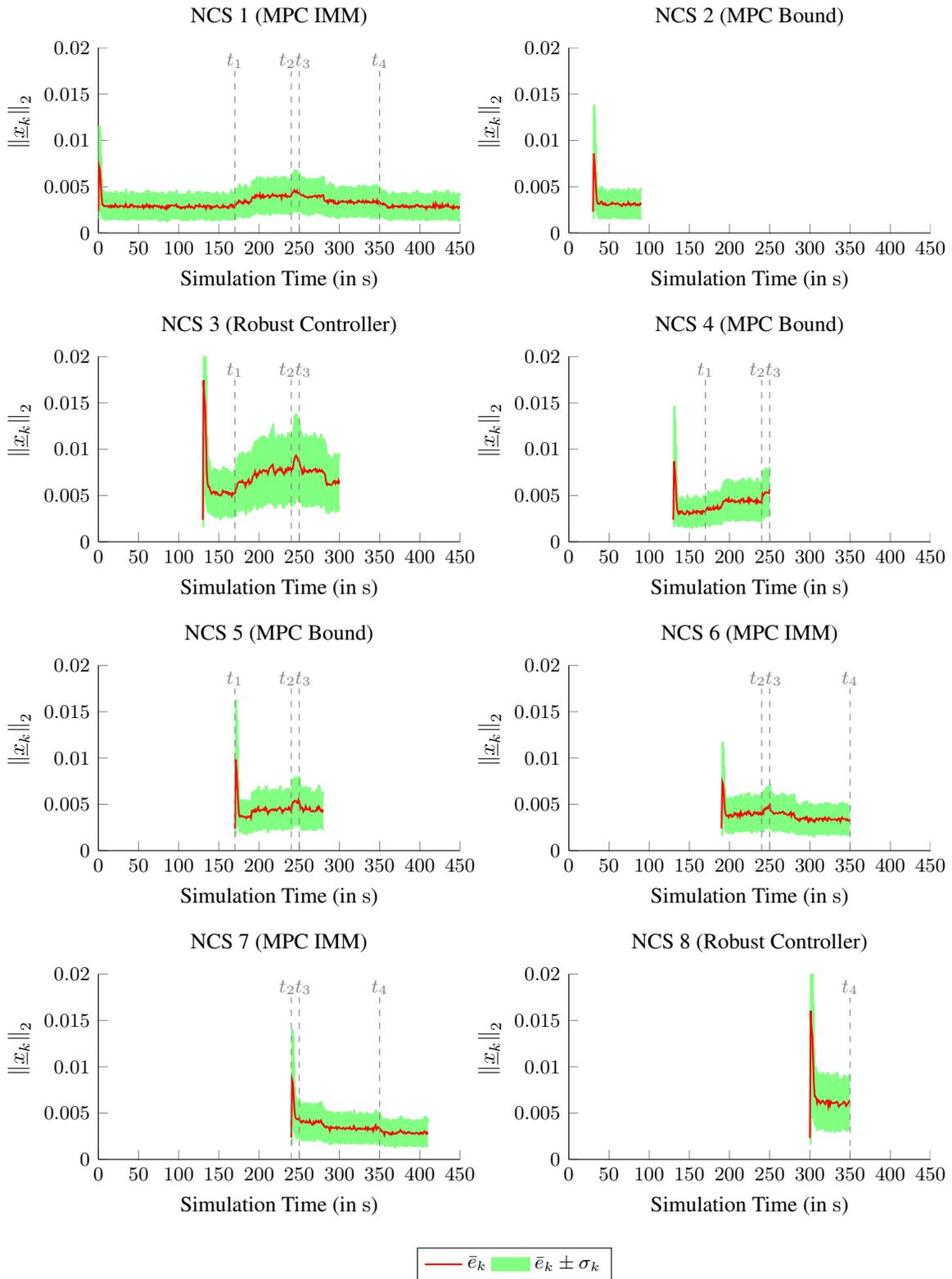


**Figure 7.11:** The corresponding sampling rates  $f_{a_{target}}$  according to Algorithm 7.2 for some of the NCS.



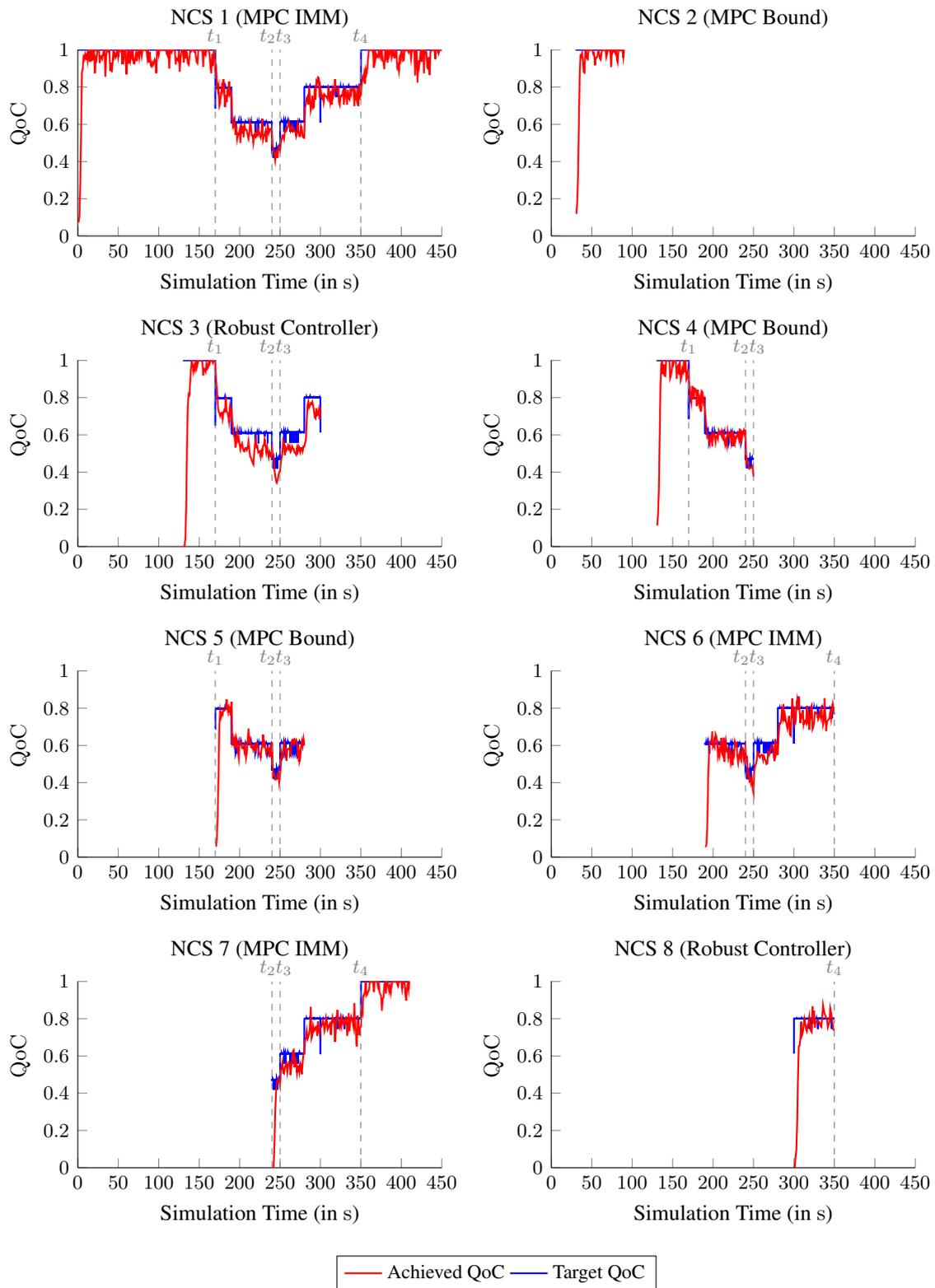
**Figure 7.12:** Excerpt of the recorded end-to-end delays in the first simulation scenario.

time for all NCS. For comparison, the desired target QoC according to Fig. 7.10 is also displayed in each plot. On the whole, the results show that, once the transient response to the initial excitation of the plant is over, all NCS manage to react to changes of the target QoC by decreasing or increasing their achieved QoC. The QoC achieved by NCS 2, NCS 4, and NCS 5, which employ the *MPC Bound*, is in line with the desired target QoC. Similarly, NCS 1, NCS 6, and NCS 7, which use the *MPC IMM*, are also able to quickly adjust their achieved QoC, but with more distinct oscillations of  $QoC_k$  around  $QoC_{target}$ . For NCS 3, which uses the *Robust Controller*, we can observe a gap between the desired target QoC and the achieved QoC. Once the target QoC drops from its maximum at time  $t_1$ ,  $QoC_k$



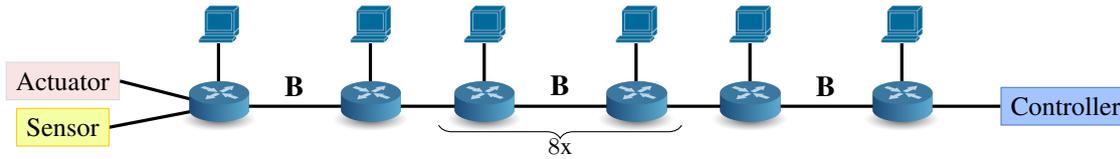
**Figure 7.13:** Sample mean  $\bar{e}_k$  and 1-sigma interval  $\bar{e}_k \pm \sigma_k$  of the norm of the plant state over time for NCS 1 (top left) to NCS 8 (bottom right).

falls below  $QoC_{target}$  for the remaining time of operation. NCS 8, which uses the *Robust Controller* as well, maintains the desired target QoC during its time of operation with only slight oscillations,



**Figure 7.14:** The desired target QoC ( $QoC_{\text{target}}$ ) in comparison to the achieved QoC ( $QoC_k$ ) for NCS 1 (top left) to NCS 8 (bottom right) in the first simulation scenario.

but needs more time to increase its QoC to the desired peak value ( $QoC_{\text{target}} = 0.8$ ) after the initial transient phase.



**Figure 7.15:** The network topology considered in the second simulation scenario.

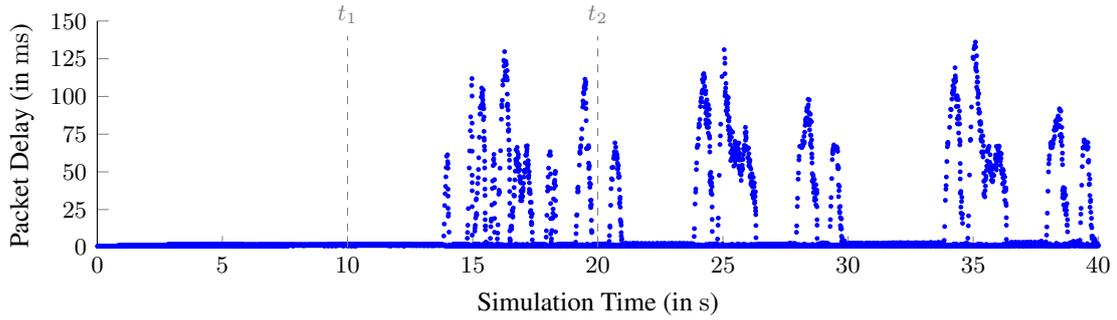
The last observations show that the *Robust Controller* lacks adaptivity compared to the SMPC algorithms and, thus, cannot respond as fast as the latter to changes of the target QoC. This is not completely unexpected since we developed the *Robust Controller* with a different goal – being robust to changing or uncertain/unknown communication conditions. As already discussed at the end of Section 6.4, increased robustness typically leads to controllers that act conservatively and, in particular, respond only slowly to changing operating points. On the other hand, the results underline that both the *MPC IMM* and *MPC Bound* enable NCS to quickly adjust their achieved QoC, allowing for fast responses to changes of the desired target QoC. In turn, in the considered scenario an efficient and fair usage of the communication resources is possible – CoCPN can distribute the capacity of the bottleneck link **B** such that all NCS achieve the same performance.

### 7.3.2 Single NCS with Mixed TCP/UDP Cross Traffic

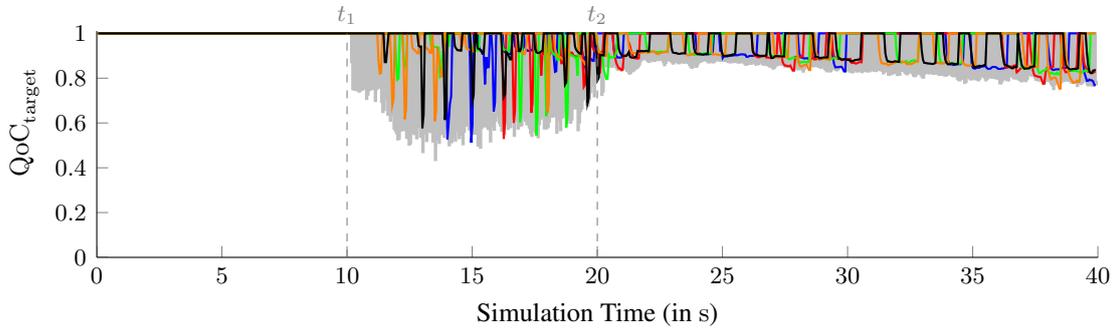
In the second simulation scenario, we consider a CPS with only a single NCS but many other applications that concurrently use the network shown in Fig. 7.15. The actuator and the sensor are physically co-located and connected to the same router. The controller is cloud-based and connected to the sensor and the actuator via multiple routers. All network links have a capacity of 100 Mbit/s and are error-free, that is, no bit errors occur. The propagation delay of the ten links marked with **B** in Fig. 7.15 is set to 250 ns, which corresponds to link lengths of 50 m. The propagation delay of the remaining links is either zero or 50 ns. For their communication, actuator, sensor, and controller use UDP and IPv6 with the parameters listed in Table F.2 in Appendix F.3.2. The cross traffic from the other applications consists of a mix of 924 UDP data flows and nine TCP flows.

We carry out  $N_s = 200$  simulation runs with the *MPC IMM*, the *MPC Bound*, and the *Robust Controller*, respectively. They are configured as described in Appendix F.3.3. In each run, we simulate the CPS over  $t_s = 40$  s, during which the NCS is active the entire time. Prior to each run, we randomly draw the initial plant state  $\underline{x}_0$  from a Gaussian distribution with mean and covariance given by (7.2). The UDP flows are active from the beginning to the end of the simulation time and configured such that they vary their data rates between 49 kbit/s and 2 Mbit/s during operation. The TCP flows start at random between  $t_1 = 10$  s and  $t_2 = 20$  s. They transmit data in chunks of 5 MB every ten seconds. The random starting times are chosen to reflect rapid changes of the communication conditions due to the sudden appearance of unrelated traffic that has high priority or requires high data rates, such as video footage or software updates. TCP is configured with the parameters given in Table F.3 in Appendix F.3.2.

Due to this choice of cross traffic, the packets sent from the cloud-based controller to the actuator (i.e., the control sequences) can experience delays of up to one hundred fifty milliseconds. Fig. 7.16 shows the end-to-end delays recorded in one of the simulation runs and illustrates nicely that the delays spike regularly whenever the TCP flows are active. The TCP cross traffic is unidirectional and, thus, does not impair the communication of the measurements sent from the sensor to the controller and the communication of the ACKs sent back from the actuator. Accordingly, their delays stay in the single-digit millisecond range.



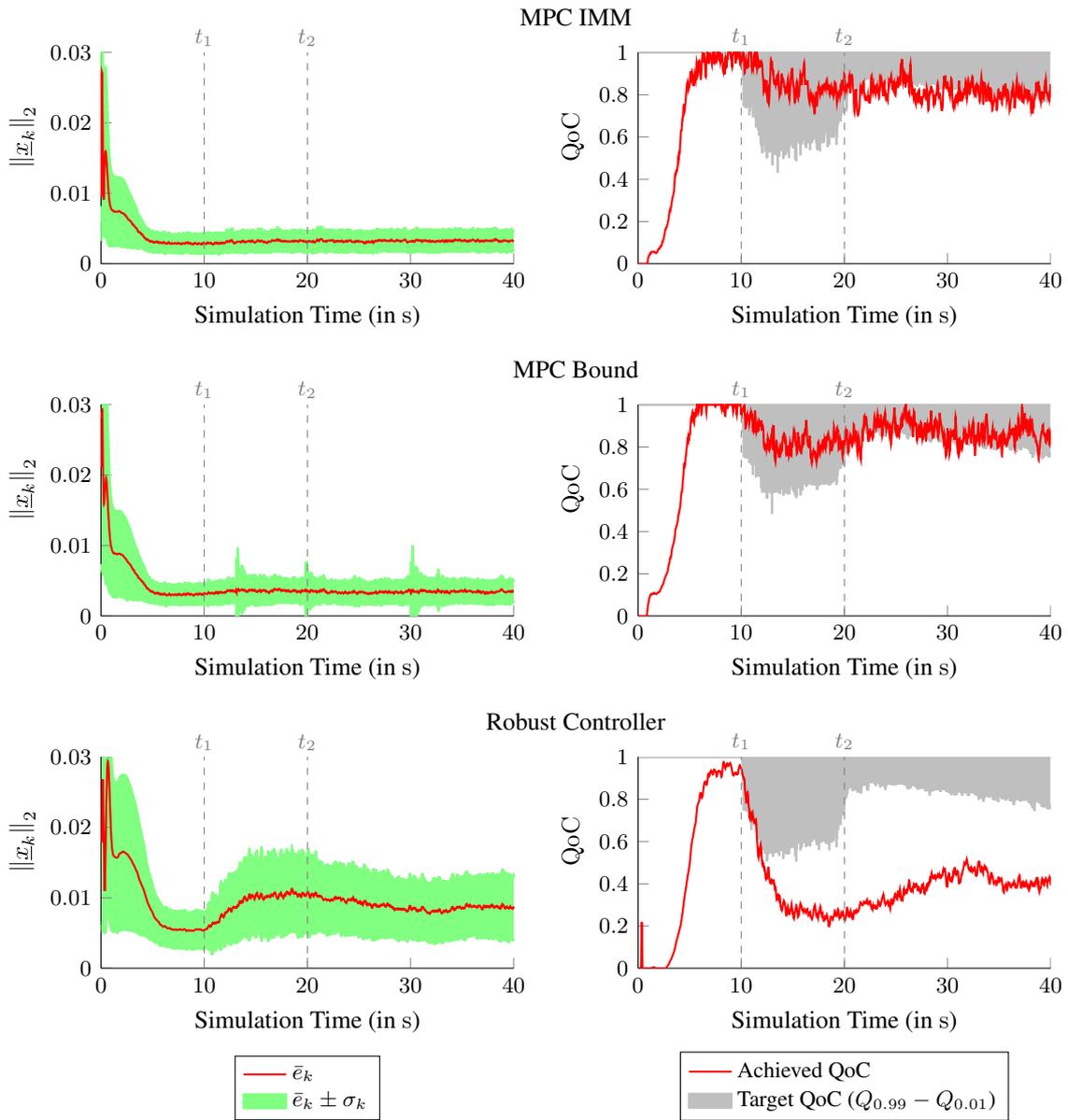
**Figure 7.16:** Recorded end-to-end delays (controller  $\rightarrow$  actuator) in one of the simulation runs of the second simulation scenario.



**Figure 7.17:** Evolution of  $QoC_{target}$  computed by the congestion control CoCC in five exemplary runs with the *MPC IMM*. The range between the 0.99- and 0.01-quantiles  $Q_{0.99}$  and  $Q_{0.01}$  is shaded in gray.

In contrast to the previous simulation scenario, the target QoC computed by the congestion control CoCC now differs from run to run because the TCP cross traffic starts at different times. We illustrate this in Fig. 7.17, which depicts the adjustments of the target QoC made by CoCC in five simulation runs with the *MPC IMM*. The graphs show that the target QoC remains maximal ( $QoC_{target} = 1$ ) at the beginning. Then, between  $t_1 = 10$  s and  $t_2 = 20$  s, we observe short but sudden declines that can reduce the target QoC below 0.6 once one of the TCP data flows becomes active. After that, decreases and increases of the target QoC alternate with each other rather periodically – CoCC is only able to keep the target QoC maximal at times when no TCP flows are active. Consequently, the sampling rate adaptations made by the controller also differ from run to run. Moreover, the computed target QoC is now also dependent on the employed control algorithm because of their different communication requirements, i.e., because of the different relationships  $f_a = f(QoC)$  obtained in Section 7.2. Hence, for a meaningful determination whether the NCS is able to achieve the desired QoC, we must take the dispersion of the computed target QoC into account. For this purpose, we consider the (empirical) 0.99- and 0.01-quantiles  $Q_{0.99}$  and  $Q_{0.01}$  of the computed target QoC. Then, the QoC achieved by the NCS should lie in the range between  $Q_{0.99}$  and  $Q_{0.01}$ . For instance, the QoC achieved in the simulation runs with the *MPC IMM* should reside inside the area shaded in gray in Fig. 7.17.

To obtain the achieved QoC, we first determine  $\bar{e}_k$  according to (7.8). The three plots in the left column of Fig. 7.18 show the evolutions of  $\bar{e}_k$  and the corresponding 1-sigma intervals over time for the *MPC IMM* (top row), the *MPC Bound* (middle row) and the *Robust Controller* (bottom row). The standard deviation  $\sigma_k$  is calculated by virtue (7.11). Based on  $\bar{e}_k$ , we then compute  $QoC_k$  according to (7.10). We again set  $QoC_k = 1$  in case  $q(\bar{e}_k) > 1$  to ensure that  $QoC_k \in [0, 1]$  for all  $k$ . The evolutions of  $QoC_k$  over time are depicted in the plots on the right of Fig. 7.18 in comparison to the desired QoC,



**Figure 7.18:** Sample mean  $\bar{e}_k$  and 1-sigma interval  $\bar{e}_k \pm \sigma_k$  of the norm of the plant state over time (left) and the achieved QoC in comparison to the target QoC (right) for the *MPC IMM* (top row), the *MPC Bound* (middle row) and the *Robust Controller* (bottom row).

which is expressed as explained above, i.e., as the range between the quantiles  $Q_{0.99}$  and  $Q_{0.01}$  of the target QoC in the corresponding simulation runs.

Similar to the previous simulation scenario, the results confirm that our control algorithms enable the NCS to react to changes of the target QoC by decreasing or increasing their achieved QoC. The QoC achieved by the NCS in the simulation runs with the *MPC IMM* lies between  $Q_{0.99}$  and  $Q_{0.01}$ , apart from two short periods after  $t_2$  and a around  $t = 30$  s, where it remains slightly below. The achieved QoC does not decrease much between  $t_1$  and  $t_2$ , during which all TCP flows become active, but stays relatively constant with oscillations around  $QoC_k = 0.8$ . Afterwards, no significant changes can be observed until the end of the simulation time. This is also reflected by  $\bar{e}_k$  and  $\sigma_k$ , which do not vary much after the transient phase has ended. The QoC achieved in the simulation runs with the *MPC Bound* is almost consistently between  $Q_{0.99}$  and  $Q_{0.01}$ , with only few short-lasting drops below. These

drops coincide with sudden increases of  $\sigma_k$ , which other than that does not vary much. Between  $t_1$  and  $t_2$ , the achieved QoC falls below 0.7, which is not the case in the runs with the *MPC IMM*. On the other hand, the QoC achieved with the *MPC Bound* then rises to reach its peak value  $QoC_k = 1$  again. Also, we can observe that the achieved QoC reaches the peak more quickly after the initial transient phase than the QoC achieved with the *MPC IMM*. The results of the simulation runs with the *Robust Controller* resemble the results of the previous simulation scenario since they reveal a gap between the desired QoC and the achieved QoC. As the target QoC starts to diminish at time  $t_1$ ,  $QoC_k$  steadily decreases, so that it falls below  $Q_{0.01}$ , where it remains until the end of the simulation time. The decrease of the achieved QoC between  $t_1$  and  $t_2$  goes along with increases of  $\bar{e}_k$  and  $\sigma_k$ .

We can again conclude that the robustness of the *Robust Controller* makes it less adaptive so that it acts overly conservatively. In particular, the controller does not exploit that a significant portion of the transmitted control sequences experiences only small delays even when the TCP cross traffic is active (cf. Fig. 7.16). Nonetheless, the *Robust Controller* guarantees stability of the closed-loop system and, hence, a reliable operation of the NCS in spite of the changing communication conditions. Regarding the *MPC IMM* and *MPC Bound*, this simulation scenario shows that both algorithms render the NCS flexible enough to cope with rapidly changing communication conditions and enable them to achieve the desired performance even in the presence of a multitude of co-existing data flows with different properties.

## 7.4 Conclusions

In the last part of this thesis, we integrated the networked control algorithms developed in Chapters 5 and 6 into the CoCPN architecture. Our goal was to demonstrate CoCPN and its cooperative concept in simulation scenarios with CoCPN-Sim. Focusing on the collaboration between our control algorithms and the congestion control CoCC, we wanted to illustrate how CoCPN avoids potentially harmful degradations of the network QoS and, at the same time, implements a fair sharing of the available communication capacity.

We considered two simulation scenarios, in which the controllers were tasked with the stabilization of double inverted pendulums over networks with highly utilized links that had to be shared with co-existing cross traffic with different properties and requirements. For this control task, we first defined the notion of QoC, which we then used to implement the CoCPN translator interface defined in Chapter 2 for our developed algorithms. Specifically, we derived the mappings  $\text{rate} = r(QoC)$  needed by the congestion control CoCC to determine the target performance  $QoC_{\text{target}}$  that leads to a fair sharing of the available communication capacity.

The simulation results illustrated that our developed control algorithms enable the NCS to react to changes of the target QoC by decreasing or increasing their achieved QoC. The *Robust Controller*, however, responded only slowly to changes of the target QoC. This was not completely against expectation because, by design, adaptivity was traded for increased robustness against changing or uncertain communication conditions. Still, the stability of the closed-loop system was guaranteed all the time and, consequently, enough elasticity provided to ensure the reliable operation of the NCS. Both SMPC algorithms, the *MPC IMM* and the *MPC Bound*, displayed fast responses to changes of the network QoS and allowed the NCS to achieve the desired performance even when the communication conditions changed rapidly. Supported by the CoCPN translator and its data exchange, undesirable degradation of the control performance was avoided, thereby enabling CoCPN to realize the desired fair distribution of the communication resources.

## Conclusions

The deployment of general-purpose networking equipment that supports the IEEE 802.3 (Ethernet) and IEEE 802.11 (WLAN) standards is becoming increasingly popular in industrial cyber-physical systems (CPS). This development is primarily fueled by the availability of inexpensive devices, the option to reuse already existing infrastructure, and increased flexibility compared to traditional fieldbus networks. On the other hand, the quality of service (QoS) provided by general-purpose networks is characterized by fluctuations, which appear for the end systems in the form of changing latencies, packet loss rates, and usable data rates. Control loops are particularly vulnerable to variations of the network QoS as they typically require guaranteed latencies for the exchange of sensor data and control commands. Consequently, approaches that render control loops elastic, that is, approaches that enable control loops to cope with changing communication conditions, are demanded for a reliable operation within such CPS. In this thesis, we developed approaches for networked control that provide the required elasticity in the context of the research project *CoCPN: Cooperative Cyber-Physical Networking*. We placed special emphasis on the design and analysis of sequence-based control algorithms that allow controllers to quickly respond to changes of the network QoS and to efficiently use the available communication capacity. Based on the results of this thesis, CoCPN realizes a novel, distributed, and cooperative concept for a fair sharing of the available communication capacity without overstressing individual resources. Thereby, potentially harmful degradations of the network QoS can be studiously avoided and the performance of all control loops can be kept in balance.

We summarize the key contributions of this thesis in the next section. The subsequent section, then, discusses potential starting points for future research.

### 8.1 Summary

In the first part of this thesis, we extended an existing modeling approach that compactly expresses the networked control system (NCS) in terms of a Markov jump linear system (MJLS), which is a particular type of hybrid system where a Markov chain governs the “switchings” between different linear dynamics. For a number of reasons, the developed model extension enabled us to pursue a more realistic treatment of the impact of the shared communication resources in relation to the majority

of the literature. First, it does not rely on the assumption that packet delays and losses occur with fixed probabilities. Instead, it models them as realizations of non-stationary processes, so that their probability of occurrence can change over time. Hence, it is flexible enough to reflect changes of the network QoS. Second, the temporal correlation of the packet delays and losses can be expressed, allowing for the integration of the impact of factors such as congestion and queuing at intermediate nodes, which affect successive packets in a similar way. Finally, it forgoes the (theoretically convenient) assumption that acknowledgment packets are transmitted instantaneously and received with negligibly small latency. Instead, our developed extension is based on a more practical point of view and takes delays and losses during their transmission into account.

The derived MJLS enables controllers to predict the influence of packet delays and losses on the control performance, making stochastic model predictive control (SMPC) well-suited for the computation of control sequences. Thus, in the second part of this thesis, we concerned ourselves with the development of algorithms for sequence-based SMPC. In principle, such algorithms determine control laws by formulating optimal control problems that are then solved via the dynamic programming (DP) recursion. Yet, our chosen modeling approach introduced an interdependency between decision-making and state estimation: The computed control sequences influence the quality of the controller's future state estimates, which in turn influence the quality of the control sequences to be computed in the future. This interdependency, typically referred to as dual effect, prevented us from solving the DP recursion in closed-form and, even worse, rendered its numerical evaluation intractable. Accordingly, we focused on computationally tractable approaches for the determination of suboptimal control laws. To that end, we proposed two novel algorithms. The first one relies on the close connection between MJLS and multiple model algorithms for state estimation. These estimation algorithms maintain the state estimate in terms of a Gaussian mixture. We used this representation to derive an algorithm for the computation of a nonlinear control law that combines the output of multiple individual controllers. Each of these is associated with one component of the Gaussian mixture, which is provided by an interacting multiple filter we tailored to our purposes. In the second proposed algorithm, the control law is linear and obtained by means of an iterative procedure that is based on repeated improvements of a reference trajectory. This reference trajectory is created by propagating a given estimate of the plant state over the optimization horizon. Our simulation results displayed that both algorithms can significantly outperform state-of-the-art algorithms for networked SMPC from the literature.

The third part of this thesis was motivated by the observation that the transition probabilities of the Markov chain that governs the switchings between the modes of the MJLS are dependent on the network models, i.e., the probabilistic descriptions of the packet delays and losses. These are forwarded to the controller by the CoCPN translator, which creates them by processing monitoring data gathered inside the network. The monitoring data, however, is itself based on measurements of, e.g., link utilizations or packet loss and error rates, and cannot capture the complete "state" of the network. Consequently, the mode transition probabilities cannot be assumed perfectly known, making it natural to explore conditions for the existence of controllers that are robust to uncertainties in the network models. Our first result in this regard was the insight that uncertain or even completely unknown network models render the MJLS polytopic, that is, the mode transition matrix always lies in a convex polytope. Consequently, we translated the problem of synthesizing robust sequence-based controllers into the problem of finding stabilizing controllers for polytopic MJLS. In the literature, a necessary and sufficient condition for the mean square stability of such systems has to date only been reported for noise-free polytopic MJLS and for polytopic MJLS that are subject to bounded disturbances. We substantially contributed to the theory of hybrid systems by establishing that this condition also holds true for the broader class of polytopic MJLS that are driven by wide-sense stationary noise. Unfortunately, the practical applicability of the condition is limited because it requires to determine whether the joint spectral radius (JSR) of a set of matrices is less than one, which is

$\mathcal{NP}$ -hard and, thus, generally cannot be done efficiently. On the other hand, this result enabled us to obtain a sufficient stability condition in terms of a linear matrix inequality (LMI) feasibility problem. This sort of feasibility problems is convex and, as such, can be evaluated by standard solvers in polynomial time. Finally, we proposed a similar set of inequalities whose feasibility guarantees the existence of stabilizing state feedback laws. This set is less restrictive than the ones already known in the literature and, thus, leads to less conservative controllers.

In the last part of this thesis, we integrated our developed algorithms into the CoCPN architecture. We demonstrated the applicability of CoCPN and its cooperative approach in challenging simulation scenarios with highly utilized links and rapidly changing communication conditions. Especially the collaboration between our proposed SMPC algorithms and the control-aware congestion control CoCC, realized by the data exchange provided by the CoCPN translator, enabled a reliable operation of the control loops without unwanted performance drops even when the available communication capacity had to be shared with multiple data flows from unrelated applications.

To conclude this section, we remark that open source implementations of all algorithms developed in the scope of this thesis are part of CoCPN-Sim [295].<sup>1</sup> CoCPN-Sim also contains implementations of the SMPC algorithms from [152, 169] that we used in Section 5.5 and implementations of other sequence-based control algorithms such as the ones developed in [123, 124]. Furthermore, it also contains standalone functions for the computation of the mode transition matrices according to Theorem 4.1 and for the computation of the lumped dynamics (4.24).

## 8.2 Future Research

Within the scope of this thesis, we touched several different aspects of networked control, each of which is an active research field on its own. Some starting points for follow-up research in these fields, based on the outcome of this thesis, are given below, together with potential future activities in the context of CoCPN.

**Modeling Networked Control Systems** Using an MJLS of the form (4.1) for the design and analysis of sequence-based controllers in this thesis was appealing since this class of hybrid systems is well understood. However, for general nonlinear plants, the augmented dynamics will be a hybrid system of the form

$$\underline{\psi}_{k+1} = \tilde{a}_k^{(\theta_k)}(\underline{\psi}_k, \underline{U}_k, \underline{w}_k), \quad (8.1)$$

with  $\tilde{a}_k^{(i)}$  nonlinear. For general stochastic hybrid systems, the literature is relatively scarce. Even for the important special case that  $\theta_k$  is a Markov chain, optimal control problems have not been explored much, so that only few notable results seem available [262, 263]. Hence, for NCS with nonlinear plants a model of the form (4.26) that expresses the actual plant input in terms of a Dirac mixture PDF might be a more promising starting point for the design of control algorithms. To corroborate (or disprove) this supposition, the pros and cons of the two modeling approaches could be studied in greater detail.

**Sequence-Based Stochastic MPC** Both algorithms for sequence-based stochastic MPC that we proposed within this thesis rely on different approximations of the cost-to-go. The approximations were necessary to overcome the main obstacle – the presence of the dual effect rendered the computation of optimal control laws intractable. Future research could check the reasonableness of the made approximations and, based thereon, seek to find conditions under which (mean square) stability of

<sup>1</sup> CoCPN-Sim is released under the GNU General Public License (GPL) and available on github: <https://github.com/spp1914-cocpn/cocpn-sim>

the closed-loop system can be established. In this regard, investigation of the convergence properties of IMM filter associated with the control algorithm developed in Section 5.3 based on the recent results [126, 188, 264] would be a suitable starting point. For the algorithm developed in Section 5.4, future work in this regard should attempt to reveal the connection between the convergence of the control law parameters according to Theorem 5.3 and the stability of the closed-loop system.

Additionally, the development of the sequence-based stochastic MPC approaches for nonlinear plants based on the augmented dynamics (8.1) could draw inspiration from the control algorithm from Section 5.3. Combining the multiple model estimator for hybrid systems from [265] with the quadratic approximation of the cost-to-go proposed in [293] should result in a set of mode-conditioned affine control laws. On the other hand, with a model of the form (4.26) at our disposal, it might be worthwhile to research scenario-based approaches [266, 267].

**Sequence-Based Stochastic Robust Control** In Chapter 6, we showed that polytopic MJLS constitute a natural way to tackle uncertainties in the network models provided by the CoCPN translator. Consequently, the synthesis of sequence-based controllers that are robust to such uncertainties has been tantamount to the synthesis of stabilizing controllers for polytopic MJLS. For stochastic systems, several definitions of stability exist in the literature [268]. For ordinary MJLS and for noise-free polytopic MJLS, the notion of mean square stability considered in this thesis is known to be equivalent to the notions of stochastic stability and exponential mean square stability. Although our numerical example in Section 6.5 indicated that this equivalence also holds for polytopic MJLS driven by wide-sense stationary noise, a formal verification is, to the best of our knowledge, still missing.

In the simulation studies with CoCPN-Sim presented in Chapter 7, the augmented state  $\underline{\psi}_k$  was not directly available. Hence, we made use of the certainty equivalence principle [158] and replaced the mode-independent state feedback law

$$\underline{U}_k = \mathbf{L}\underline{\psi}_k,$$

that resulted from the application of Corollary 6.3 by

$$\underline{U}_k = \mathbf{L}\hat{\underline{\psi}}_k,$$

i.e., we used the estimate  $\hat{\underline{\psi}}_k$  in place of the true state  $\underline{\psi}_k$ . This control law is suboptimal and, in particular, not guaranteed to stabilize the plant. Thus, future research activities should concentrate on the design of (mean square) stabilizing dynamic output feedback laws. Using our results, it should be possible to show that the existence of such laws depends on the feasibility of matrix inequalities similar to the ones we obtained in Theorem 6.8 and Corollary 6.3.

**CoCPN** To date, the CoCPN architecture is implemented in our simulation and evaluation framework CoCPN-Sim [295]. We used CoCPN-Sim to demonstrate the applicability of CoCPN and its cooperative concept in different simulation scenarios. Experimental validation of the results constitutes an important next steps towards the deployment of CoCPN in real-world applications. The integration of wireless networks based on the IEEE 802.11 (WLAN) standards into the CoCPN architecture, which has been out of scope so far, could accompany the experimental validation as such networks become more and more popular in industrial applications.

Currently, controllers react to changes of the target QoC by adapting their sampling rates. However, it is known that variations in the controller's sampling rate can render the closed-loop system unstable [269, 270]. Thus, future work should investigate the impact of these rate adaptations in greater detail and research conditions under which (mean square) stability of the closed-loop dynamics can be guaranteed. Additionally, it might be interesting to explore other strategies to adjust the sending rates of the controllers. In this regard, the recently developed resource-aware MPC approaches,

which incorporate given traffic specifications such as desired average packet rates directly into the optimization, seem promising [271–273].

Our definition of QoC in Chapter 7 was straightforward and grounded on the time average of the control error in steady-state operation of the NCS. Future work could research alternative definitions of QoC based on other statistical measures. In particular, measures used for control performance monitoring in industrial applications could be explored [274, 275]. Similarly, it might be worthwhile to find ways to detect mismatches between the actually achieved QoC and the desired QoC computed by CoCC according to the mapping  $\text{rate} = r(\text{QoC})$  provided by the CoCPN translator. Based thereon, one could then research methods to adapt this mapping at runtime.

Finally, we observed in the second evaluation scenario in Chapter 7 that a considerable proportion of the packets sent from the controller to the actuator experienced only small delays even when the TCP cross traffic was active (cf. Fig. 7.16)). This is due to the fact CoCC always prioritizes a certain amount of packets to guarantee some minimum control performance. It might be useful to devise network models for the controllers that incorporate the effects caused by mechanisms inside the communication protocols. Based on the gathered monitoring information, the CoCPN translator could also obtain more sophisticated network models by applying approaches based on nonparametric statistics. In this regard, methods for nonparametric density estimation based on Pitman-Yor processes, which are especially suited for heavy-tailed distributions, could be studied [276].



# Mathematical Concepts and Results

In this appendix, we state mathematical concepts, terminology, and expressions that are used throughout this thesis. To keep the presentation concise, we present all results without proof.

## A.1 Metric Space Concepts

The notion of a metric, as defined below, extends the fundamental concept of distances in the Euclidean plane to elements of arbitrary sets.

**Definition A.1:** (Naylor and Sell [277, p. 45])

*The pair  $(X, d_x)$  is called a metric space if  $X$  is a set and  $d_x(x_1, x_2)$  is a real-valued function, called the metric, defined for  $x_1, x_2 \in X$  that satisfies the following conditions:*

- (i)  $d_x(x_1, x_2) \geq 0$  and  $d_x(x_1, x_1) = 0$  for all  $x_1, x_2 \in X$ .
- (ii) If  $d_x(x_1, x_2) = 0$ , then  $x_1 = x_2$  for all  $x_1, x_2 \in X$ .
- (iii)  $d_x(x_1, x_2) = d_x(x_2, x_1)$  for all  $x_1, x_2 \in X$ .
- (iv)  $d_x(x_1, x_2) \leq d_x(x_1, x_3) + d_x(x_2, x_3)$  for all  $x_1, x_2, x_3 \in X$ .

Let  $(X, d_x)$  and  $(Y, d_y)$  be arbitrary metric spaces and  $F: X \mapsto Y$  a mapping from  $(X, d_x)$  into  $(Y, d_y)$ . Furthermore, let  $G: X \mapsto Y$  be a bijection, i.e.,  $G$  is a one-to-one and surjective mapping from  $(X, d_x)$  onto  $(Y, d_y)$  and thus invertible. Finally, let  $\{x_k\} = \{x_0, x_1, \dots\}$  be a sequence of points in  $(X, d_x)$ .

**Definition A.2:** (Naylor and Sell [277, Definition 3.5.1])

*The mapping  $F$  is said to be continuous at the point  $x_0$  in  $X$  if for every real number  $\epsilon > 0$ , there exists a real number  $\delta = \delta(\epsilon, x_0) > 0$  such that  $d_y(F(x), F(x_0)) < \epsilon$  whenever  $d_x(x, x_0) < \delta$ . The mapping  $F$  is said to be continuous if it is continuous at each point in its domain.*

**Definition A.3:** (Naylor and Sell [277, Definition 3.5.2])

The mapping  $F$  is said to be uniformly continuous if for each  $\epsilon > 0$ , there exists  $\delta = \delta(\epsilon) > 0$  such that for any  $x_0$  one has  $d_y(F(x), F(x_0)) < \epsilon$  whenever  $d_x(x, x_0) < \delta$ .

In particular, any uniformly continuous mapping is continuous.

**Definition A.4:** (Naylor and Sell [277, Definitions 3.10.1 and 3.13.7])

The bijection  $G$  is called a homeomorphism if both  $G$  and  $G^{-1}$  are continuous. If both  $G$  and  $G^{-1}$  are uniformly continuous, then  $G$  is said to be a uniform homeomorphism.

If a (uniform) homeomorphism exists,  $(X, d_x)$  and  $(Y, d_y)$  are called (uniformly) homeomorphic. Definitions A.3 and A.4 imply that every isometry, i.e., every bijection  $G$  from  $(X, d_x)$  onto  $(Y, d_y)$  for which  $d_y(G(x_1), G(x_2)) = d_x(x_1, x_2)$  holds for every  $x_1, x_2 \in X$ , is a uniform homeomorphism.

**Definition A.5:** (Naylor and Sell [277, Definition 3.6.1])

The sequence  $\{x_k\}$  in  $(X, d_x)$  is said to be convergent if there is a point  $x$  in  $(X, d_x)$  with the property that for each  $\epsilon > 0$  there is an integer  $N$  such that  $d_x(x_n, x) < \epsilon$  whenever  $n \geq N$ . The point  $x$  is called the limit of the sequence  $\{x_k\}$ .

For a convergent sequence  $\{x_k\}$ , we write  $\lim_{k \rightarrow \infty} x_k = x$ .

**Theorem A.1:** (Naylor and Sell [277, Theorem 3.10.2])

The following statements are equivalent:

- (i) The bijection  $G$  is a homeomorphism.
- (ii) A sequence  $\{x_k\}$  in  $(X, d_x)$  converges to a point  $x$  if and only if the sequence  $\{G(x_k)\}$  in  $(Y, d_y)$  converges to  $G(x)$ .

**Definition A.6:** (Naylor and Sell [277, Definition 3.13.1])

The sequence  $\{x_k\}$  in  $(X, d_x)$  is said to be a Cauchy sequence if for each  $\epsilon > 0$  there exists an integer  $N = N(\epsilon)$  such that  $d_x(x_n, x_m) < \epsilon$  for any  $n, m \geq N$ .

It is evident from the definition that any convergent sequence is also a Cauchy sequence. The opposite, however, is generally not true. For instance, the sequence defined by  $x_{k+1} = \frac{x_k}{2} + \frac{1}{x_k}$  with  $x_0 = 1$  is a Cauchy sequence in the metric space  $(\mathbb{Q}, |\cdot|)$ , but it is not convergent in  $(\mathbb{Q}, |\cdot|)$ , since its limit  $\lim_{k \rightarrow \infty} x_k = \sqrt{2}$  is not a rational number. Metric spaces possessing the property that any convergent sequence is a Cauchy sequence are called *complete* according to the following definition.

**Definition A.7:** (Naylor and Sell [277, Definition 3.13.3])

A metric space  $(X, d_x)$  is called complete if each Cauchy sequence in  $(X, d_x)$  is also convergent in  $(X, d_x)$ .

## A.2 Norms and Banach Spaces

The notion of a norm, as defined below, extends the fundamental concept of length of vectors in the Euclidean space to elements of arbitrary vector spaces.

**Definition A.8:** (Naylor and Sell [277, Definition 5.2.1])

A real-valued function  $\|x\|$  defined on a vector space  $X$  is a norm if

- (i)  $\|x\| \geq 0$  and  $\|x\| = 0$  if and only if  $x = 0$ ,
- (ii)  $\|x + y\| \leq \|x\| + \|y\|$ ,
- (iii)  $\|\alpha x\| = |\alpha| \|x\|$  for any scalar  $\alpha$ .

A pair  $(X, \|\cdot\|)$ , where  $X$  is a vector space and  $\|\cdot\|$  a norm defined on  $X$ , is called a *normed vector space*. Since any norm on  $X$  generates a metric by the function  $d_x(x, y) = \|x - y\|$ , every normed vector space is also a metric space. A normed vector space that is also complete according to Definition A.7 is called a *Banach space*. In particular, any finite-dimensional normed vector space is a Banach space [277, Theorem 5.10.2]. Another remarkable property of finite-dimensional normed vector spaces is that *all* linear mappings are continuous.

**Theorem A.2:** (Naylor and Sell [277, Theorem 5.10.4])

Let  $X$  and  $Y$  be normed vector spaces and let  $L: X \mapsto Y$  be a linear mapping. If  $X$  is finite-dimensional, then  $L$  is continuous.

To conclude this section, we state some further properties of norms in general and, in particular, of some matrix norms. It follows directly from the definition that every norm is a convex mapping. Hence, by Jensen’s inequality we have for any random variable  $\underline{x}$  that  $\|E\{\underline{x}\}\| \leq E\{\|\underline{x}\|\}$  [256, p. 77]. Although there are typically many norms that can be defined on a vector space  $X$ , if  $X$  is finite-dimensional any two of them are *equivalent* [277, Theorem 5.10.6]. That is, if  $\|\cdot\|_a$  and  $\|\cdot\|_b$  are two norms on a finite-dimensional vector space  $X$ , then for any  $x \in X$

$$m \|x\|_a \leq \|x\|_b \leq M \|x\|_a ,$$

with  $m, M > 0$ .

Finally, in the case of  $X = \mathbb{R}^{n \times n}$ , the vector space of  $n$ -by- $n$  matrices, some norms are also *submultiplicative* [252, p. 290], i.e., it holds for all  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$

$$\|\mathbf{AB}\| \leq \|\mathbf{A}\| \|\mathbf{B}\| . \tag{A.1}$$

One prominent submultiplicative norm is the *Frobenius norm*  $\|\cdot\|_F$  which for a general, not necessarily square matrix  $\mathbf{A} \in \mathbb{R}^{n \times m}$  is given by

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m a_{ij}^2} = \sqrt{\text{tr}[\mathbf{A}^T \mathbf{A}]} . \tag{A.2}$$

The Frobenius norm is also *compatible* with the standard Euclidean norm on  $\mathbb{R}^n$  because we have

$$\|\mathbf{A}\underline{x}\|_2 \leq \|\mathbf{A}\|_F \|\underline{x}\|_2 . \tag{A.3}$$

for all  $\underline{x} \in \mathbb{R}^n, \mathbf{A} \in \mathbb{R}^{n \times n}$ .

### A.3 Kronecker Product and Vectorization

The Kronecker product  $\mathbf{A} \otimes \mathbf{B}$  of two arbitrarily sized matrices  $\mathbf{A} \in \mathbb{R}^{n \times m}$  and  $\mathbf{B} \in \mathbb{R}^{p \times q}$  produces the  $np$ -by- $mq$  block matrix [278]

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1m}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2m}\mathbf{B} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1}\mathbf{B} & a_{n2}\mathbf{B} & \dots & a_{nm}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{np \times mq}. \quad (\text{A.4})$$

The following properties of the Kronecker product are easily verified and frequently used within this thesis. For matrices of  $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$  of appropriate dimensions and any scalar  $\alpha$ , it holds [278]

$$\begin{aligned} (\mathbf{A} \otimes \mathbf{B})^T &= \mathbf{A}^T \otimes \mathbf{B}^T, \\ (\mathbf{A} + \mathbf{B}) \otimes (\mathbf{C} + \mathbf{D}) &= \mathbf{A} \otimes \mathbf{C} + \mathbf{A} \otimes \mathbf{D} + \mathbf{B} \otimes \mathbf{C} + \mathbf{B} \otimes \mathbf{D}, \\ (\alpha\mathbf{A}) \otimes \mathbf{B} &= \mathbf{A} \otimes (\alpha\mathbf{B}) = \alpha(\mathbf{A} \otimes \mathbf{B}). \end{aligned}$$

Next, we note that an expression of the form

$$\underline{y} = a_1\mathbf{M}_1\underline{z}_1 + a_2\mathbf{M}_2\underline{z}_2 + \dots + a_l\mathbf{M}_l\underline{z}_l,$$

with  $\mathbf{M}_j \in \mathbb{R}^{n \times m}$ ,  $\underline{z}_j \in \mathbb{R}^m$ , and  $a_j$  arbitrary coefficients can be written by means of the Kronecker product according to

$$\underline{y} = (\underline{a}^T \otimes \mathbf{I}_n) \tilde{\mathbf{M}} \tilde{\underline{z}},$$

with

$$\underline{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_l \end{bmatrix}, \quad \tilde{\underline{z}} = \begin{bmatrix} \underline{z}_1 \\ \underline{z}_2 \\ \vdots \\ \underline{z}_l \end{bmatrix}, \quad \tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{M}_l \end{bmatrix}. \quad (\text{A.5})$$

Accordingly, the set of equations

$$\begin{aligned} \underline{y}^{(1)} &= a_1^{(1)}\mathbf{M}_1\underline{z}_1 + a_2^{(1)}\mathbf{M}_2\underline{z}_2 + \dots + a_l^{(1)}\mathbf{M}_l\underline{z}_l, \\ \underline{y}^{(2)} &= a_1^{(2)}\mathbf{M}_1\underline{z}_1 + a_2^{(2)}\mathbf{M}_2\underline{z}_2 + \dots + a_l^{(2)}\mathbf{M}_l\underline{z}_l, \\ &\vdots \\ \underline{y}^{(r)} &= a_1^{(r)}\mathbf{M}_1\underline{z}_1 + a_2^{(r)}\mathbf{M}_2\underline{z}_2 + \dots + a_l^{(r)}\mathbf{M}_l\underline{z}_l, \end{aligned} \quad (\text{A.6})$$

can be compactly expressed in terms of the stacked vector  $\tilde{\underline{y}} = [(\underline{y}^{(1)})^T \ (\underline{y}^{(2)})^T \ \dots \ (\underline{y}^{(r)})^T]^T$

$$\tilde{\underline{y}} = (\mathbf{A}^T \otimes \mathbf{I}_n) \tilde{\mathbf{M}} \tilde{\underline{z}}, \quad (\text{A.7})$$

where  $\tilde{\mathbf{M}}$  and  $\tilde{\underline{z}}$  as in (A.5), and

$$\mathbf{A} = \begin{bmatrix} a_1^{(1)} & a_1^{(2)} & \dots & a_1^{(r)} \\ a_2^{(1)} & a_2^{(2)} & \dots & a_2^{(r)} \\ \vdots & \vdots & \vdots & \vdots \\ a_l^{(1)} & a_l^{(2)} & \dots & a_l^{(r)} \end{bmatrix}.$$

In the same vein, a weighted sum of  $l$  square matrices

$$\mathbf{Y} = a_1 \mathbf{M}_1 + a_2 \mathbf{M}_2 + \cdots + a_l \mathbf{M}_l, \quad (\text{A.8})$$

with  $\mathbf{M}_j \in \mathbb{R}^{n \times n}$  and  $a_j$  nonnegative coefficients is equivalently expressed as

$$\mathbf{Y} = (\underline{\alpha} \otimes \mathbf{I}_n) \tilde{\mathbf{M}} (\underline{\alpha} \otimes \mathbf{I}_n)^T, \quad (\text{A.9})$$

where  $\tilde{\mathbf{M}}$  is as in (A.5) and  $\underline{\alpha} = [\sqrt{a_1} \ \sqrt{a_2} \ \cdots \ \sqrt{a_l}]$ .

In applications, the Kronecker product is frequently encountered in conjunction with the vectorization operation, which converts a matrix into a vector by stacking its columns. Formally, the vectorization of a matrix  $\mathbf{A} \in \mathbb{R}^{n \times m}$  is defined as the mapping  $\text{vec}: \mathbb{R}^{n \times m} \mapsto \mathbb{R}^{nm}$  with [279]

$$\text{vec}(\mathbf{A}) = \text{vec} \left( \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \right) = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \\ a_{12} \\ \vdots \\ a_{n2} \\ \vdots \\ \vdots \\ a_{1m} \\ \vdots \\ a_{nm} \end{bmatrix}. \quad (\text{A.10})$$

It is evident from (A.10) that vectorization is a linear mapping, since  $\text{vec}(\alpha \mathbf{A}) = \alpha \text{vec}(\mathbf{A})$  and  $\text{vec}(\mathbf{A} + \mathbf{B}) = \text{vec}(\mathbf{A}) + \text{vec}(\mathbf{B})$  for all  $\mathbf{A}, \mathbf{B}$  and all scalars  $\alpha$ . Moreover, because  $\|\text{vec}(\mathbf{A})\|_2 = \|\mathbf{A}\|_F$ , vectorization is an isometry and thus uniformly continuous (cf. Definition A.3) with a uniformly continuous inverse  $\text{vec}^{-1}: \mathbb{R}^{nm} \mapsto \mathbb{R}^{n \times m}$  [277, Theorem 5.7.1].<sup>1</sup> Hence, vectorization is a uniformly homeomorphic mapping from the vector space of all  $n$ -by- $m$  matrices onto the Euclidean space  $\mathbb{R}^{nm}$ . Using Theorem A.1 this means that, if we shall determine whether a sequence of matrices  $\{\mathbf{X}_k\}$  converges, it is enough to study the corresponding sequence of vectors  $\{\text{vec}(\mathbf{X}_k)\}$ .

Additionally, vectorization allows to express the usual matrix multiplication as a linear transformation. In particular, for matrices  $\mathbf{A}, \mathbf{B}$ , and  $\mathbf{C}$  of conformable dimensions we have

$$\text{vec}(\mathbf{ABC}) = [\mathbf{C}^T \otimes \mathbf{A}] \text{vec}(\mathbf{B}). \quad (\text{A.11})$$

Eq. (A.11) has found widespread application in control engineering and system theory. For instance, a solution (if existent) of the Lyapunov equation

$$\mathbf{AXA}^T + \mathbf{Q} = \mathbf{X},$$

with  $\mathbf{X}$  unknown, is obtained by appropriately reshaping the solution of the linear system

$$[\mathbf{I} - (\mathbf{A} \otimes \mathbf{A})] \text{vec}(\mathbf{X}) = \text{vec}(\mathbf{Q}).$$

<sup>1</sup> An explicit expression for the inverse is  $\text{vec}^{-1}(\underline{x}) = (\text{vec}(\mathbf{I}_m)^T \otimes \mathbf{I}_n)(\mathbf{I}_m \otimes \underline{x})$ .

## A.4 Positive Definite and Positive Semidefinite Matrices

In this section, we state some properties of positive definite and positive semidefinite matrices, respectively. We start with an immediate consequence of Sylvester's law of inertia [252, Theorem 4.5.8].

### Theorem A.3:

Let  $\mathbf{A}, \mathbf{P} \in \mathbb{R}^{n \times n}$  such that  $\mathbf{A}$  is symmetric and  $\mathbf{P}$  is invertible. Then  $\mathbf{A}$  is positive definite if and only if  $\mathbf{B} = \mathbf{PAP}^T$  is.

Submultiplicative matrix norms give an explicit upper bound for the *spectral radius* of a matrix  $\mathbf{A}$  since  $\rho(\mathbf{A}) \leq \|\mathbf{A}\|$  [252, Theorem. 5.6.9]. The following lemma is an immediate consequence.

### Lemma A.1:

Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be positive semidefinite with eigenvalues  $\lambda_j \geq 0$ . Then it holds

$$\operatorname{tr}[\mathbf{A}] = \sum_{j=1}^n \lambda_j \leq \sum_{j=1}^n \rho(\mathbf{A}) \leq n \|\mathbf{A}\| ,$$

for any submultiplicative matrix norm  $\|\cdot\|$ .

The next result follows from the fact that the set of positive semidefinite matrices, when equipped with the inner product  $\langle \mathbf{A}, \mathbf{B} \rangle = \operatorname{tr}[\mathbf{AB}]$ , is a self-dual cone [256, p. 52].

### Theorem A.4:

Let  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  be positive semidefinite. Then  $\operatorname{tr}[\mathbf{AB}] \geq 0$ .

The following fact provides explicit bounds for  $\operatorname{tr}[\mathbf{AB}]$ , when one of the matrices is positive definite [280, Lemma 5].

### Lemma A.2:

Let  $\mathbf{A}$  be positive definite and let its minimal and maximal eigenvalue be denoted by  $\lambda_{\min}(\mathbf{A})$  and  $\lambda_{\max}(\mathbf{A})$ . Then for any  $\mathbf{B} \succcurlyeq 0$ , we have  $\lambda_{\min}(\mathbf{A}) \operatorname{tr}[\mathbf{B}] \leq \operatorname{tr}[\mathbf{AB}] \leq \lambda_{\max}(\mathbf{A}) \operatorname{tr}[\mathbf{B}]$ .

The next result shows that for any positive semidefinite matrix  $\mathbf{B}$ , we can always find a positive definite matrix  $\mathbf{A}$  such that  $\mathbf{A} \succ \mathbf{B}$  [252, Theorem 7.7.3].

### Theorem A.5:

Let  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  such that  $\mathbf{A} \succ 0$  and  $\mathbf{B} \succcurlyeq 0$ . Then  $\mathbf{A} \succ \mathbf{B}$  if and only if  $\rho(\mathbf{BA}^{-1}) < 1$ .

The following basic property of positive definite matrices is easily verified and has found widespread application in the literature [198, 281].

### Lemma A.3:

Let  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  such that  $\mathbf{A} \succ 0$ . Then it holds  $\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} \succcurlyeq \mathbf{B} + \mathbf{B}^T - \mathbf{A}$ .

The following result proves that the *Cholesky decomposition* exists for any positive semidefinite matrix [282, Lemma 1.1].

**Theorem A.6:**

Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  such that  $\mathbf{A} \succcurlyeq 0$ . Then, there exists at least one upper triangular matrix  $\mathbf{R} \in \mathbb{R}^{n \times n}$  with nonnegative diagonal elements such that  $\mathbf{A} = \mathbf{R}^T \mathbf{R}$ .

Finally, we consider a handy characterization of positive definite block matrices, namely the *Schur complement*, and its generalization to positive semidefinite matrices. Both have found widespread application in control engineering and concern conformably partitioned matrices of the form

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{12}^T & \mathbf{Q}_{22} \end{bmatrix}, \quad (\text{A.12})$$

with  $\mathbf{Q}_{11}$  and  $\mathbf{Q}_{22}$  square.

**Theorem A.7:** (Kreindler and Jameson [283])

Let  $\mathbf{Q}$  be partitioned according to (A.12). Then:

$\mathbf{Q} \succ 0$  if and only if

$$\mathbf{Q}_{22} \succ 0, \quad \mathbf{Q}_{11} - \mathbf{Q}_{12} \mathbf{Q}_{22}^{-1} \mathbf{Q}_{12}^T \succ 0,$$

or

$$\mathbf{Q}_{11} \succ 0, \quad \mathbf{Q}_{22} - \mathbf{Q}_{12}^T \mathbf{Q}_{11}^{-1} \mathbf{Q}_{12} \succ 0.$$

$\mathbf{Q} \succcurlyeq 0$  if and only if

$$\mathbf{Q}_{22} \succcurlyeq 0, \quad \mathbf{Q}_{12} = \mathbf{Q}_{12} \mathbf{Q}_{22}^\dagger \mathbf{Q}_{22}, \quad \mathbf{Q}_{11} - \mathbf{Q}_{12} \mathbf{Q}_{22}^\dagger \mathbf{Q}_{12}^T \succcurlyeq 0,$$

or

$$\mathbf{Q}_{11} \succcurlyeq 0, \quad \mathbf{Q}_{12} = \mathbf{Q}_{11} \mathbf{Q}_{11}^\dagger \mathbf{Q}_{12}, \quad \mathbf{Q}_{22} - \mathbf{Q}_{12}^T \mathbf{Q}_{11}^\dagger \mathbf{Q}_{12} \succcurlyeq 0,$$

where  $\dagger$  denotes the Moore-Penrose pseudoinverse (cf. Appendix A.5).

## A.5 The Moore-Penrose Pseudoinverse

The Moore-Penrose pseudoinverse of a matrix  $\mathbf{A}$ , denoted by  $\mathbf{A}^\dagger$  throughout this thesis, satisfies [284]

$$\begin{aligned} \mathbf{A} \mathbf{A}^\dagger \mathbf{A} &= \mathbf{A}, \\ \mathbf{A}^\dagger \mathbf{A} \mathbf{A}^\dagger &= \mathbf{A}^\dagger, \\ (\mathbf{A} \mathbf{A}^\dagger)^T &= \mathbf{A} \mathbf{A}^\dagger, \\ (\mathbf{A}^\dagger \mathbf{A})^T &= \mathbf{A}^\dagger \mathbf{A}. \end{aligned} \quad (\text{A.13})$$

It always exists and is unique but coincides with  $\mathbf{A}^{-1}$  if  $\mathbf{A}$  is square and nonsingular.  $\mathbf{A}^\dagger$  is symmetric if  $\mathbf{A}$  is. Additionally,  $\mathbf{A}^\dagger$  enjoys the property that the equation  $\mathbf{A} \underline{x} = \underline{b}$  has solutions if and only if  $\mathbf{A} \mathbf{A}^\dagger \underline{b} = \underline{b}$ . In the affirmative case, any solution is of the form

$$\underline{x} = \mathbf{A}^\dagger \underline{b} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \underline{v}, \quad (\text{A.14})$$

with  $\underline{v}$  an arbitrary vector of conformable dimension. Eq. (A.14) yields a unique solution  $\underline{x}^+ = \mathbf{A}^\dagger \underline{b}$  only if  $\mathbf{A}$  has full column rank, in which case  $\mathbf{I} - \mathbf{A}^\dagger \mathbf{A} = \mathbf{0}$ . Otherwise,  $\underline{x}^+$  is the *minimum norm solution* to  $\mathbf{A} \underline{x} = \underline{b}$ , i.e.,  $\|\underline{x}^+\|_2 \leq \|\underline{x}\|_2$  for all solutions  $\underline{x}$ . To see this, we use (A.13) to establish that  $\mathbf{A}^\dagger \underline{b}$  and  $(\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \underline{v}$  are orthogonal since  $(\mathbf{A}^\dagger \underline{b})^T (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \underline{v} = 0$  for any  $\underline{v}$ . It thus follows

$$\|\underline{x}\|_2^2 = \left\| \mathbf{A}^\dagger \underline{b} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \underline{v} \right\|_2^2 = \left\| \mathbf{A}^\dagger \underline{b} \right\|_2^2 + \left\| (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \underline{v} \right\|_2^2 \geq \left\| \mathbf{A}^\dagger \underline{b} \right\|_2^2 = \|\underline{x}^+\|_2^2.$$

A typical way to compute the Moore-Penrose pseudoinverse is to use the *singular value decomposition*, according to which any  $\mathbf{A} \in \mathbb{R}^{n \times m}$  can be factored as  $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$  with orthogonal matrices  $\mathbf{U} \in \mathbb{R}^{n \times n}$  and  $\mathbf{V} \in \mathbb{R}^{m \times m}$ , and where  $\mathbf{\Sigma}$  is a conformable (rectangular) diagonal matrix with nonnegative elements. Then,  $\mathbf{A}^\dagger = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^T$ , where  $\mathbf{\Sigma}^\dagger$  is simply the transpose of  $\mathbf{\Sigma}$  with all positive diagonal elements replaced by their reciprocals [252, p. 421].

## A.6 The Perron-Frobenius Theorem and Stochastic Matrices

The *Perron-Frobenius theorem* is a well-known result from matrix theory with many applications. In the scope of this thesis, we apply it to stochastic matrices (i.e., transition matrices of Markov chains) in Chapters 3 and 4.

We state the theorem in its version for square matrices with positive entries [252, Theorem 8.2.11]. However, it remains valid for *primitive* matrices, i.e., nonnegative matrices with a strictly positive power [252, p. 516].<sup>2</sup>

### Theorem A.8: (Perron-Frobenius)

Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  have only positive entries, that is,  $\mathbf{A} > 0$ , and denote by  $\rho(\mathbf{A})$  its spectral radius. Then:

- (i)  $\rho(\mathbf{A}) > 0$ .
- (ii)  $\rho(\mathbf{A})$  is an eigenvalue of  $\mathbf{A}$ .
- (iii) There exists an eigenvector  $\underline{v}$  of  $\mathbf{A}$  with eigenvalue  $\rho(\mathbf{A})$  and only positive entries, i.e.,  $\underline{v} > 0$  and  $\mathbf{A}\underline{v} = \rho(\mathbf{A})\underline{v}$ . Likewise, there exists a left eigenvector  $\underline{w}$  of  $\mathbf{A}$  with eigenvalue  $\rho(\mathbf{A})$  and only positive entries, i.e.,  $\underline{w} > 0$  and  $\underline{w}^T\mathbf{A} = \rho(\mathbf{A})\underline{w}^T$ .
- (iv)  $\rho(\mathbf{A})$  is an algebraically simple eigenvalue of  $\mathbf{A}$ , i.e., its eigenspace is one-dimensional.
- (v)  $\rho(\mathbf{A})$  is the unique eigenvalue with maximum modulus, i.e.,  $|\lambda| < \rho(\mathbf{A})$  for every eigenvalue  $\lambda \neq \rho(\mathbf{A})$ .
- (vi)  $\lim_{k \rightarrow \infty} \frac{\mathbf{A}^k}{\rho(\mathbf{A})^k} = \underline{v}\underline{w}^T$  where  $\underline{v}, \underline{w}$  are as in (iii), normalized such that  $\underline{v}^T\underline{w} = 1$ .

Recall that a *stationary distribution*  $\underline{p}$  of a Markov chain with transition matrix  $\mathbf{T}$  is a stochastic vector (i.e.,  $\underline{p} > 0$  and  $\sum_i p_i = 1$ ) that satisfies the relation  $\underline{p}^T\mathbf{T} = \underline{p}^T$ , i.e.,  $\underline{p}$  is a left eigenvector of  $\mathbf{T}$  with eigenvalue 1. The following result confirms that every Markov chain<sup>3</sup> has at least one such distribution [285, Lemma 9.1].

### Lemma A.4:

Let  $\mathbf{T}$  be a stochastic matrix. Then 1 is an eigenvalue and every eigenvalue  $\lambda$  satisfies  $|\lambda| \leq 1$ .

In general, the stationary distribution  $\underline{p}$  is not unique. However, if all transition probabilities are positive, i.e.,  $\mathbf{T} > 0$ , then the uniqueness of  $\underline{p}$  is guaranteed by the Perron-Frobenius theorem. Moreover,  $\underline{p} > 0$  holds. Finally, we can conclude that  $\underline{p}$  is also the *limiting distribution* of the chain, that is,  $\underline{p}_{k+1}^T = \underline{p}_k^T\mathbf{T}$  converges to  $\underline{p}$  for any initial distribution  $\underline{p}_0$ . To see this, we note that  $\underline{1}$  is an eigenvector of  $\mathbf{T}$  with eigenvalue 1 satisfying  $\underline{1}^T\mathbf{T} = \underline{1}^T$ . Hence, we may write using the last item of the Perron-Frobenius

<sup>2</sup>A square matrix  $\mathbf{A}$  is called primitive if  $\mathbf{A} \geq 0$  and  $\mathbf{A}^m > 0$  for some integer  $m \geq 1$ .

<sup>3</sup>We tacitly assume a finite state space here.

theorem

$$\lim_{k \rightarrow \infty} \underline{p}_k^T \mathbf{T} = \underline{p}_0^T \lim_{k \rightarrow \infty} \mathbf{T}^k = \underline{p}_0^T \underline{1} \underline{p}^T = \underline{p}^T.$$

## A.7 The Banach Space of Tuples of Square Matrices

It is common practice in the literature to analyze Markov jump linear systems by investigating the properties of the second moment of the state variable. As in Chapters 5 and 6 of this thesis, this is conveniently done by breaking down the second moment into a tuple of matrices, each of which is associated with one mode of the system. Throughout this thesis, we consider jump systems with  $N + 1$  modes, resulting in tuples consisting of  $N + 1$  square matrices.

In formal terms, such a tuple  $\mathcal{X} = (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)})$ , where each  $\mathbf{X}^{(i)}$  is  $n$ -by- $n$ , is an element of the space

$$\mathbb{H}^n \triangleq \left\{ (\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)}) \mid \mathbf{X}^{(i)} \in \mathbb{R}^{n \times n} \right\},$$

that is composed of all  $(N + 1)$ -tuples of square matrices. It is a routine matter to verify that  $\mathbb{H}^n$  is a vector space with addition and scalar multiplication defined as

$$\begin{aligned} \mathcal{X} + \mathcal{Y} &\triangleq (\mathbf{X}^{(0)} + \mathbf{Y}^{(0)}, \mathbf{X}^{(1)} + \mathbf{Y}^{(1)}, \dots, \mathbf{X}^{(N)} + \mathbf{Y}^{(N)}), \\ \alpha \mathcal{X} &\triangleq (\alpha \mathbf{X}^{(0)}, \alpha \mathbf{X}^{(1)}, \dots, \alpha \mathbf{X}^{(N)}), \end{aligned}$$

for  $\mathcal{X}, \mathcal{Y} \in \mathbb{H}^n$  and  $\alpha \in \mathbb{R}$ , i.e., the usual addition and scalar multiplication on  $\mathbb{R}^{n \times n}$  is applied elementwise. It is also readily verified that  $\mathbb{H}^n$  is finite-dimensional and that for any matrix norm

$$\|\mathcal{X}\|_{\mathbb{H}} \triangleq \sum_{i=0}^N \|\mathbf{X}^{(i)}\|,$$

is a norm on  $\mathbb{H}^n$  according to Definition A.8, so that  $\mathbb{H}^n$  is a Banach space (cf. Appendix A.2).

Just as  $\mathbb{R}^{n \times m}$  is uniformly homeomorphic to  $\mathbb{R}^{nm}$ ,  $\mathbb{H}^n$  is uniformly homeomorphic to  $\mathbb{R}^{(N+1)n^2}$ . To see this, we first note that the elementwise application of the vectorization operation yields an  $n^2$ -by- $(N + 1)$ -dimensional matrix if the resulting vectors,  $\text{vec}(\mathbf{X}^{(i)})$ , are column-wise arranged. Vectorizing this matrix then results in a vector in  $\mathbb{R}^{(N+1)n^2}$ . This idea is formalized by the mapping  $\hat{\varphi}: \mathbb{H}^n \mapsto \mathbb{R}^{(N+1)n^2}$  defined as [127, 258]

$$\hat{\varphi}(\mathcal{X}) \triangleq \begin{bmatrix} \text{vec}(\mathbf{X}^{(0)}) \\ \text{vec}(\mathbf{X}^{(1)}) \\ \vdots \\ \text{vec}(\mathbf{X}^{(N)}) \end{bmatrix}. \quad (\text{A.15})$$

It is easy to verify that (A.15) is a uniformly homeomorphic mapping from the vector space of all  $(N + 1)$ -tuples of  $n$ -by- $n$  matrices onto the Euclidean space  $\mathbb{R}^{(N+1)n^2}$  according to Definition A.4. By Theorem A.1 this implies that, if we are asked to determine whether a sequence of tuples  $\{\mathcal{X}_k\}$  is convergent, it is enough to consider the corresponding sequence of vectors  $\{\hat{\varphi}(\mathcal{X}_k)\}$ .



## Proofs of the Results in Chapter 4

In the following three sections, we provide the proofs of Theorem 4.1, Theorem 4.2, and Lemma 4.1, each of which is restated here for convenience. The proofs are presented in the order of appearance of the results.

### **B.1** Proof of Theorem 4.1

**Theorem 4.1:**

Let  $\tau_k^{\text{ca}}$  be an independent process as defined in Section 3.2.1. Then,  $\theta_k$  forms a time-inhomogeneous Markov chain with state space  $\{0, 1, \dots, N\}$  and transition probabilities  $t_{k,ij} = \mathbb{P}[\theta_{k+1} = j | \theta_k = i]$  given by

$$t_{k,ij} = \begin{cases} p_{k+1}^{(0)} & j = 0 \\ \left(1 - p_{k+1}^{(0)}\right) \prod_{m=0}^{i-1} \left(1 - q_k^{(m)}\right) & j = i + 1 \\ 0 & j > i + 1 \\ q_k^{(j-1)} \left(1 - p_{k+1}^{(0)}\right) \prod_{m=0}^{j-2} \left(1 - q_k^{(m)}\right) & 1 \leq j \leq i \leq N - 1 \\ \left(1 - p_{k+1}^{(0)}\right) \prod_{m=0}^{N-2} \left(1 - q_k^{(m)}\right) & i = j = N \end{cases},$$

with  $p_k^{(i)} = \mathbb{P}[\tau_k^{\text{ca}} = i]$  and where  $q_k^{(j)}$  is the conditional probability that  $\underline{U}_{k-j}$  arrives at time  $k + 1$  given that it has not been received up to time  $k$

$$q_k^{(j)} = \mathbb{P}[\tau_{k-j}^{\text{ca}} = j + 1 | \tau_{k-j}^{\text{ca}} > j] = \frac{p_{k-j}^{(j+1)}}{1 - \sum_{m=0}^j p_{k-j}^{(m)}}. \quad (4.11)$$

*Proof.* The proof is similar to the time-homogeneous case discussed in [123, Lemma 3.1] and exploits that  $\tau_k^{\text{ca}}$  is an independent process. First,  $j = 0$  indicates a transition from  $\theta_k = i$  to  $\theta_{k+1} = 0$ , which means that  $\underline{U}_{k+1}$  arrives at the actuator without delay. The corresponding probability is thus  $p_{k+1}^{(0)}$ . For the remaining cases, we note that the probability that  $\underline{U}_{k-j}$  arrives at time  $k+1$  given that it has not been received up to time  $k$  is equal to the conditional probability  $\text{P}[\tau_{k-j}^{\text{ca}} = j+1 | \tau_{k-j}^{\text{ca}} > j]$ . This probability is given by

$$\text{P}[\tau_{k-j}^{\text{ca}} = j+1 | \tau_{k-j}^{\text{ca}} > j] = \frac{\text{P}[\tau_{k-j}^{\text{ca}} = j+1, \tau_{k-j}^{\text{ca}} > j]}{\text{P}[\tau_{k-j}^{\text{ca}} > j]} = \frac{\text{P}[\tau_{k-j}^{\text{ca}} = j+1]}{1 - \text{P}[\tau_{k-j}^{\text{ca}} \leq j]},$$

which yields (4.11). Thus,  $1 - q_k^{(j)}$  denotes the probability that  $\underline{U}_{k-j}$  does not arrive at the actuator at time  $k+1$ , given that it has not arrived earlier. Then we note that a transition from  $\theta_k = i$  to  $\theta_{k+1} = i+1$  corresponds to the event that the currently buffered sequence  $\underline{U}_k^{\text{bf}} = \underline{U}_{k-i}$  is not replaced. Hence, none of the sequences  $\underline{U}_{k-(i-1)}, \dots, \underline{U}_{k+1}$  will arrive at time  $k+1$  and we get

$$t_{k,i(i+1)} = \left(1 - p_{k+1}^{(0)}\right) \prod_{m=0}^{i-1} \left(1 - q_k^{(m)}\right).$$

Transitions from  $\theta_k = i$  to  $\theta_{k+1} \geq i+2$  are impossible since the age of the buffered sequence can only increase by one, namely in case it is not replaced. Hence,  $t_{k,i,j} = 0$  for  $j > i+1$ .

For  $1 \leq j \leq i \leq N-1$ , we have transitions from  $\theta_k = i$  to  $\theta_{k+1} = j$  that indicate a replacement of the buffered sequence  $\underline{U}_k^{\text{bf}} = \underline{U}_{k-i}$  by a newer sequence  $\underline{U}_{k-(j-1)}$ . The corresponding probability is

$$t_{k,i,j} = q_k^{(j-1)} \left(1 - p_{k+1}^{(0)}\right) \prod_{m=0}^{j-2} \left(1 - q_k^{(m)}\right),$$

since the control sequences  $\underline{U}_{k-(j-2)}, \dots, \underline{U}_k$  cannot be available yet.

Finally, we note that the case  $i = j = N$  corresponds to the event that at time  $k+1$  no valid control sequence is buffered given that no valid control sequence was available at time  $k$  either. This means that none of the sequences  $\underline{U}_{k-(N-2)}, \dots, \underline{U}_k, \underline{U}_{k+1}$  that provide inputs for time  $k+1$  will be received. Hence,

$$t_{k,NN} = \left(1 - p_{k+1}^{(0)}\right) \prod_{m=0}^{N-2} \left(1 - q_k^{(m)}\right),$$

which concludes the proof.  $\square$

## B.2 Proof of Lemma 4.1

### Lemma 4.1:

For every initial distribution, the expanded chain  $\tau_k^{\text{ca}}$  converges to its unique stationary distribution  $\bar{p}$  with elements  $\bar{p}^{(i_0, i_1, \dots, i_{N-1})}$  given by

$$\bar{p}^{(i_0, i_1, \dots, i_{N-1})} = p^{(i_{N-1})} p_{i_{N-1} i_{N-2}}^{\text{ca}} p_{i_{N-2} i_{N-3}}^{\text{ca}} \cdots p_{i_1 i_0}^{\text{ca}}, \quad (4.17)$$

with  $i_0, i_1, \dots, i_{N-1} \in \{0, \dots, M\}$ , and where  $\underline{p} = [p^{(0)} p^{(1)} \dots p^{(M)}]^T$  is the stationary distribution of  $\tau_k^{\text{ca}}$ .

*Proof.* We proceed similar to the proof of Theorem 6.5.2 in [129]. We have  $\bar{p}^{(i_0, i_1, \dots, i_{N-1})} > 0$  by Assumption 3.5 and

$$\begin{aligned} \sum_{(i_0, i_1, \dots, i_{N-1})} \bar{p}^{(i_0, i_1, \dots, i_{N-1})} &= \sum_{i_0} \sum_{i_1} \dots \sum_{i_{N-2} i_{N-1}} p^{(i_{N-1})} p_{i_{N-1} i_{N-2}}^{\text{ca}} p_{i_{N-2} i_{N-3}}^{\text{ca}} \dots p_{i_1 i_0}^{\text{ca}} \\ &= \sum_{i_0} \sum_{i_1} \dots \sum_{i_{N-2}} p^{(i_{N-2})} p_{i_{N-2} i_{N-3}}^{\text{ca}} \dots p_{i_1 i_0}^{\text{ca}} \\ &= \sum_{i_0} \sum_{i_1} p^{(i_1)} p_{i_1 i_0}^{\text{ca}} \\ &= \sum_{i_0} p^{(i_0)} = 1, \end{aligned}$$

hence  $\bar{p}$  is a stochastic vector. To conclude that  $\bar{p}$  is a stationary distribution, it remains to show that

$$\sum_{(i_0, i_1, \dots, i_{N-1})} \bar{p}^{(i_0, i_1, \dots, i_{N-1})} \tilde{p}_{(i_0, i_1, \dots, i_{N-1})(j_0, j_1, \dots, j_{N-1})} = \bar{p}^{(j_0, j_1, \dots, j_{N-1})}, \quad (\text{B.1})$$

with  $\tilde{p}_{(i_0, i_1, \dots, i_{N-1})(j_0, j_1, \dots, j_{N-1})}$  the transition probabilities of  $\underline{\tau}_k^{\text{ca}}$ . These are given by

$$\begin{aligned} &\tilde{p}_{(i_0, i_1, \dots, i_{N-1})(j_0, j_1, \dots, j_{N-1})} \\ &= \text{P}[\underline{\tau}_{k+1}^{\text{ca}} = (j_0, j_1, \dots, j_{N-1}) \mid \underline{\tau}_k^{\text{ca}} = (i_0, i_1, \dots, i_{N-1})] \\ &= \text{P}[\tau_{k+1}^{\text{ca}} = j_0, \tau_k^{\text{ca}} = j_1, \dots, \tau_{k-(N-2)}^{\text{ca}} = j_{N-1} \mid \tau_k^{\text{ca}} = i_0, \tau_{k-1}^{\text{ca}} = i_1, \dots, \tau_{k-(N-1)}^{\text{ca}} = i_{N-1}] \\ &= \text{P}[\tau_{k+1}^{\text{ca}} = j_0 \mid \tau_k^{\text{ca}} = i_0, \tau_k^{\text{ca}} = j_1] \\ &\quad \cdot \text{P}[\tau_k^{\text{ca}} = j_1, \tau_{k-1}^{\text{ca}} = j_2, \dots, \tau_{k-(N-2)}^{\text{ca}} = j_{N-1}] \\ &\quad \mid \tau_k^{\text{ca}} = i_0, \tau_{k-1}^{\text{ca}} = i_1, \dots, \tau_{k-(N-2)}^{\text{ca}} = i_{N-2}, \tau_{k-(N-1)}^{\text{ca}} = i_{N-1}] \\ &= p_{i_0 j_0}^{\text{ca}} \delta_{i_0, j_1} \text{P}[\tau_k^{\text{ca}} = j_1, \tau_{k-1}^{\text{ca}} = j_2, \dots, \tau_{k-(N-2)}^{\text{ca}} = j_{N-1} \\ &\quad \mid \tau_k^{\text{ca}} = j_1, \tau_{k-1}^{\text{ca}} = i_1, \dots, \tau_{k-(N-2)}^{\text{ca}} = i_{N-2}, \tau_{k-(N-1)}^{\text{ca}} = i_{N-1}] \\ &= p_{i_0 j_0}^{\text{ca}} \delta_{i_0, j_1} \delta_{i_1, j_2} \dots \delta_{i_{N-3}, j_{N-2}} \delta_{i_{N-2}, j_{N-1}}, \end{aligned} \quad (\text{B.2})$$

with  $\delta_{i,j}$  the Kronecker delta, i.e.,  $\delta_{i,j} = 1$  if  $i = j$ , and 0 otherwise. But using (4.17) and (B.2) in (B.1) yields

$$\begin{aligned} &\sum_{i_0} \sum_{i_1} \dots \sum_{i_{N-2} i_{N-1}} p^{(i_{N-1})} p_{i_{N-1} i_{N-2}}^{\text{ca}} p_{i_{N-2} i_{N-3}}^{\text{ca}} \dots p_{i_1 i_0}^{\text{ca}} p_{i_0 j_0}^{\text{ca}} \delta_{i_0, j_1} \delta_{i_1, j_2} \dots \delta_{i_{N-3}, j_{N-2}} \delta_{i_{N-2}, j_{N-1}} \\ &= \sum_{i_0} \sum_{i_1} \dots \sum_{i_{N-2}} p^{(i_{N-2})} p_{i_{N-2} i_{N-3}}^{\text{ca}} \dots p_{i_1 i_0}^{\text{ca}} p_{i_0 j_0}^{\text{ca}} \delta_{i_0, j_1} \delta_{i_1, j_2} \dots \delta_{i_{N-3}, j_{N-2}} \delta_{i_{N-2}, j_{N-1}} \\ &= p^{(j_{N-1})} p_{j_{N-1} j_{N-2}}^{\text{ca}} p_{j_{N-2} j_{N-3}}^{\text{ca}} \dots p_{j_1 j_0}^{\text{ca}} = \bar{p}^{(j_0, j_1, \dots, j_{N-1})}. \end{aligned}$$

The Perron-Frobenius theorem (cf. Appendix A.6) ensures that  $\bar{p}$  is the only stationary distribution and that  $\underline{\tau}_k^{\text{ca}}$  converges towards it for any initial  $\tilde{p}_0$  provided the transition matrix of  $\underline{\tau}_k^{\text{ca}}$ ,  $\tilde{\mathbf{P}}^{\text{ca}}$ , is primitive, i.e., provided that there exists an  $l \in \mathbb{N}$  such that  $(\tilde{\mathbf{P}}^{\text{ca}})^l > 0$ . To show this, we use that the element  $a_{ij}^{(l)}$  of the  $l$ -th power of a general square matrix  $\mathbf{A}$  is given by the expression

$$a_{ij}^{(l)} = \sum_{k_1} \sum_{k_2} \dots \sum_{k_{l-1}} a_{i k_{l-1}} a_{k_{l-1} k_{l-2}} a_{k_{l-2} k_{l-3}} \dots a_{k_1 j}. \quad (\text{B.3})$$

Using (B.2) in (B.3) reveals that

$$\tilde{p}_{(i_0, i_1, \dots, i_{N-1})(j_0, j_1, \dots, j_{N-1})}^{(N)} = p_{i_0 j_{N-1}}^{\text{ca}} p_{j_{N-1} j_{N-2}}^{\text{ca}} \dots p_{j_1 j_0}^{\text{ca}} > 0,$$

holds for any element of  $(\tilde{\mathbf{P}}^{\text{ca}})^N$  due to Assumption 3.5 and hence  $(\tilde{\mathbf{P}}^{\text{ca}})^N > 0$ .  $\square$

### B.3 Proof of Theorem 4.2

#### Theorem 4.2:

The stationary distribution of the lumped dynamics (4.24) for  $\theta_k$  is unique and given by  $\bar{\mathbf{V}}^T \bar{\underline{\mu}}$ , where  $\bar{\underline{\mu}}$  is the stationary distribution (4.17) of  $\tau_k^{\text{ca}}$  and  $\bar{\mathbf{V}}$  is given by (4.20). Moreover,

$$\lim_{k \rightarrow \infty} \underline{\mu}_0^T \mathbf{T}^k = \lim_{k \rightarrow \infty} \underline{\mu}_0^T \left( \bar{\mathbf{U}} \tilde{\mathbf{P}}^{\text{ca}} \bar{\mathbf{V}} \right)^k = \bar{\underline{\mu}}^T \bar{\mathbf{V}}, \quad (4.25)$$

for any initial distribution  $\underline{\mu}_0$ .

*Proof.* Due to the particular choice of  $\bar{\mathbf{U}}$ , it holds  $\bar{\underline{\mu}}^T = \bar{\underline{\mu}}^T \bar{\mathbf{V}} \bar{\mathbf{U}}$ .<sup>1</sup> Multiplying both sides to the right by  $\tilde{\mathbf{P}}^{\text{ca}} \bar{\mathbf{V}}$  gives

$$\bar{\underline{\mu}}^T \tilde{\mathbf{P}}^{\text{ca}} \bar{\mathbf{V}} = \bar{\underline{\mu}}^T \bar{\mathbf{V}} \bar{\mathbf{U}} \tilde{\mathbf{P}}^{\text{ca}} \bar{\mathbf{V}}.$$

Because  $\bar{\underline{\mu}}$  is the stationary distribution of  $\tau_k^{\text{ca}}$ , we have  $\bar{\underline{\mu}}^T = \bar{\underline{\mu}}^T \tilde{\mathbf{P}}^{\text{ca}}$ , and hence

$$\bar{\underline{\mu}}^T \bar{\mathbf{V}} = \bar{\underline{\mu}}^T \tilde{\mathbf{P}}^{\text{ca}} \bar{\mathbf{V}} = \bar{\underline{\mu}}^T \bar{\mathbf{V}} \bar{\mathbf{U}} \tilde{\mathbf{P}}^{\text{ca}} \bar{\mathbf{V}} = \bar{\underline{\mu}}^T \bar{\mathbf{V}} \mathbf{T},$$

with (4.24), so that  $\bar{\mathbf{V}}^T \bar{\underline{\mu}}$  is a stationary distribution. Now assume that  $\underline{\mu}$  is another stationary distribution. Then, there must exist some  $\tilde{\underline{\mu}}$  such that  $\tilde{\underline{\mu}}^T \bar{\mathbf{V}} = \underline{\mu}^T$  and  $\underline{\mu}^T \bar{\mathbf{U}} = \tilde{\underline{\mu}}^T$ , so that

$$\begin{aligned} \tilde{\underline{\mu}}^T \bar{\mathbf{V}} &= \underline{\mu}^T \\ &= \underline{\mu}^T \mathbf{T} \\ &= \underline{\mu}^T \bar{\mathbf{U}} \tilde{\mathbf{P}}^{\text{ca}} \bar{\mathbf{V}} \\ &= \tilde{\underline{\mu}}^T \tilde{\mathbf{P}}^{\text{ca}} \bar{\mathbf{V}}. \end{aligned}$$

This implies  $\tilde{\underline{\mu}}^T = \tilde{\underline{\mu}}^T \tilde{\mathbf{P}}^{\text{ca}}$  because by construction (cf. (4.20))  $\bar{\mathbf{V}}$  has full rank. Thus, by Lemma 4.1,  $\tilde{\underline{\mu}} = \bar{\underline{\mu}}$  must hold and hence  $\underline{\mu}^T = \bar{\underline{\mu}}^T \bar{\mathbf{V}} = \bar{\underline{\mu}}^T$ , proving uniqueness of the stationary distribution. Finally, (4.25) follows from (4.19) and (4.21) since

$$\underline{\mu}_{k+1}^T = \underline{\mu}_k^T \mathbf{T} = \underline{\mu}_k^T \bar{\mathbf{U}} \tilde{\mathbf{P}}^{\text{ca}} \bar{\mathbf{V}} = \tilde{\underline{\mu}}_k^T \tilde{\mathbf{P}}^{\text{ca}} \bar{\mathbf{V}} = \tilde{\underline{\mu}}_0^T \left( \tilde{\mathbf{P}}^{\text{ca}} \right)^{k+1} \bar{\mathbf{V}} \xrightarrow{k \rightarrow \infty} \bar{\underline{\mu}}^T \bar{\mathbf{V}}.$$

□

<sup>1</sup>On the other hand, we have  $\underline{z}^T \bar{\mathbf{U}} \bar{\mathbf{V}} = \underline{z}^T$  for any  $\underline{z} \in \mathbb{R}^{N+1}$ , since  $\bar{\mathbf{U}} \bar{\mathbf{V}} = \mathbf{I}_{N+1}$ , which motivates the naming:  $\bar{\mathbf{V}}$  collects what  $\bar{\mathbf{U}}$  distributes [141].

## Proofs of the Results in Chapter 5

In the following sections, we give the proofs of Lemmas 5.1, 5.2, 5.3, and 5.4 and of Theorems 5.1, 5.2, and 5.3, each of which is restated here for convenience. The proofs are presented in the order of appearance of the results.

### C.1 Proof of Lemma 5.1

**Lemma 5.1:**

For every time step  $k$ , the estimation error  $\tilde{e}_k = \underline{\psi}_k - \mathbb{E}\{\underline{\psi}_k \mid \mathcal{I}_k\}$  is a function of the computed control sequences  $\underline{U}_0, \dots, \underline{U}_{k-1}$ .

*Proof.* Fix two arbitrary sequences of control sequences  $\underline{U}_0, \dots, \underline{U}_{k-1}$  and  $\underline{U}'_0, \dots, \underline{U}'_{k-1}$ . We proceed similar to the proof of Lemma A.3 in [123] and consider two versions of the augmented dynamics (4.1), namely

$$\begin{aligned} \underline{\psi}_{k+1} &= \tilde{\mathbf{A}}^{(\theta_k)} \underline{\psi}_k + \tilde{\mathbf{B}}^{(\theta_k)} \underline{U}_k + \tilde{\underline{w}}_k, \\ \underline{y}_k &= [\mathbf{C} \ \mathbf{0}] \underline{\psi}_k + \underline{v}_k, \end{aligned} \tag{C.1}$$

and

$$\begin{aligned} \underline{\psi}'_{k+1} &= \tilde{\mathbf{A}}^{(\theta'_k)} \underline{\psi}'_k + \tilde{\mathbf{B}}^{(\theta'_k)} \underline{U}'_k + \tilde{\underline{w}}'_k, \\ \underline{y}'_k &= [\mathbf{C} \ \mathbf{0}] \underline{\psi}'_k + \underline{v}'_k. \end{aligned} \tag{C.2}$$

We consider the temporal evolution of these systems when the initial conditions and the noise vectors are identical, i.e., we have  $\underline{\psi}_0 = \underline{\psi}'_0$ ,  $\theta_0 = \theta'_0$ ,  $\tilde{\underline{w}}_t = \tilde{\underline{w}}'_t$ , and  $\underline{v}_t = \underline{v}'_t$  for  $t = 0, 1, \dots, k-1$ . Also, let the packet delays and losses be identical for both systems. Hence, the mode trajectories  $\theta_{0:k-1}$  and  $\theta'_{0:k-1}$  are fixed and identical, so that (C.1) and (C.2) become linear time-varying systems. We may

thus write

$$\begin{aligned}\underline{\psi}_k &= \mathbf{\Omega}_k \underline{\psi}_0 + \mathbf{\Lambda}_k \begin{bmatrix} \underline{U}_0 \\ \vdots \\ \underline{U}_{k-1} \end{bmatrix} + \mathbf{\Gamma}_k \begin{bmatrix} \tilde{w}_0 \\ \vdots \\ \tilde{w}_{k-1} \end{bmatrix}, \\ \underline{\psi}'_k &= \mathbf{\Omega}_k \underline{\psi}'_0 + \mathbf{\Lambda}_k \begin{bmatrix} \underline{U}'_0 \\ \vdots \\ \underline{U}'_{k-1} \end{bmatrix} + \mathbf{\Gamma}_k \begin{bmatrix} \tilde{w}_0 \\ \vdots \\ \tilde{w}_{k-1} \end{bmatrix},\end{aligned}$$

for some matrices  $\mathbf{\Omega}_k$ ,  $\mathbf{\Lambda}_k$ , and  $\mathbf{\Gamma}_k$  that depend on  $\theta_{0:k-1}$ . Then, it holds for the conditional expectations

$$\begin{aligned}\mathbb{E}\{\underline{\psi}_k \mid \mathcal{I}_k\} &= \mathbb{E}\{\mathbf{\Omega}_k \mid \mathcal{I}_k\} \mathbb{E}\{\underline{\psi}_0 \mid \mathcal{I}_k\} + \mathbb{E}\{\mathbf{\Lambda}_k \mid \mathcal{I}_k\} \begin{bmatrix} \underline{U}_0 \\ \vdots \\ \underline{U}_{k-1} \end{bmatrix}, \\ \mathbb{E}\{\underline{\psi}'_k \mid \mathcal{I}_k\} &= \mathbb{E}\{\mathbf{\Omega}_k \mid \mathcal{I}_k\} \mathbb{E}\{\underline{\psi}'_0 \mid \mathcal{I}_k\} + \mathbb{E}\{\mathbf{\Lambda}_k \mid \mathcal{I}_k\} \begin{bmatrix} \underline{U}'_0 \\ \vdots \\ \underline{U}'_{k-1} \end{bmatrix}.\end{aligned}$$

We thus obtain for the estimation errors  $\tilde{e}_k = \underline{\psi}_k - \mathbb{E}\{\underline{\psi}_k \mid \mathcal{I}_k\}$  and  $\tilde{e}'_k = \underline{\psi}'_k - \mathbb{E}\{\underline{\psi}'_k \mid \mathcal{I}_k\}$

$$\tilde{e}_k = \mathbf{\Omega}_k \underline{\psi}_0 - \mathbb{E}\{\mathbf{\Omega}_k \mid \mathcal{I}_k\} \mathbb{E}\{\underline{\psi}_0 \mid \mathcal{I}_k\} + (\mathbf{\Lambda}_k - \mathbb{E}\{\mathbf{\Lambda}_k \mid \mathcal{I}_k\}) \begin{bmatrix} \underline{U}_0 \\ \vdots \\ \underline{U}_{k-1} \end{bmatrix} + \mathbf{\Gamma}_k \begin{bmatrix} \tilde{w}_0 \\ \vdots \\ \tilde{w}_{k-1} \end{bmatrix}, \quad (\text{C.3})$$

$$\tilde{e}'_k = \mathbf{\Omega}_k \underline{\psi}'_0 - \mathbb{E}\{\mathbf{\Omega}_k \mid \mathcal{I}_k\} \mathbb{E}\{\underline{\psi}'_0 \mid \mathcal{I}_k\} + (\mathbf{\Lambda}_k - \mathbb{E}\{\mathbf{\Lambda}_k \mid \mathcal{I}_k\}) \begin{bmatrix} \underline{U}'_0 \\ \vdots \\ \underline{U}'_{k-1} \end{bmatrix} + \mathbf{\Gamma}_k \begin{bmatrix} \tilde{w}_0 \\ \vdots \\ \tilde{w}_{k-1} \end{bmatrix}. \quad (\text{C.4})$$

We see that the estimation errors are only equal if  $\underline{U}_t = \underline{U}'_t$  for all  $t = 0, \dots, k-1$ . Consequently,  $\tilde{e}_k$  is not independent of the previous control sequences  $\underline{U}_0, \dots, \underline{U}_{k-1}$ .  $\square$

We note that the expressions for  $\tilde{e}_k$  and  $\tilde{e}'_k$  exhibit why the assumption of TCP-like communication between controller and actuator drastically simplifies the dynamic programming recursion (5.6). This assumption implies that  $\theta_{0:k-1} \in \mathcal{I}_k$  and hence  $\mathbb{E}\{\mathbf{\Lambda}_k \mid \mathcal{I}_k\} = \mathbf{\Lambda}_k$ , so that the third summands in (C.3) and (C.4), respectively, vanish. Hence, the estimation error is independent of past control actions and its contribution to the cost-to-go can be excluded from the minimization, which in turn eliminates the dual effect.

## C.2 Proof of Lemma 5.2

### Lemma 5.2:

For  $t = 0, 1, \dots, K-1$ , it holds for all  $j = 0, 1, \dots, N$

$$\tilde{\mathbf{X}}_{t+1}^{(j)} = \sum_{i=0}^N t_{ij} \left( \mathbf{\Gamma}_t^{(i)} \tilde{\mathbf{X}}_t^{(i)} \left( \mathbf{\Gamma}_t^{(i)} \right)^\top + \mu_t^{(i)} \mathbf{N}_t \right). \quad (5.49)$$

*Proof.* The result is similar to Proposition 3.14 in [175] and proved in an akin fashion. Denote by  $f(\underline{z})$  the probability density function of  $\underline{z}$ , conditioned on the available information  $\mathcal{I}_0$ . The direct evaluation of the underlying expectation yields

$$\begin{aligned}
 \tilde{\mathbf{X}}_{t+1}^{(j)} &= \mathbb{E}_{\substack{\tilde{\mathbf{x}}_{t+1}, \\ \theta_{t+1}}} \left\{ \tilde{\mathbf{x}}_{t+1} \tilde{\mathbf{x}}_{t+1}^T \mathbb{1}_{\{\theta_{t+1}=j\}} \mid \mathcal{I}_0 \right\} \\
 &= \int_{\mathbb{R}^{2n_\xi}} \sum_{r=0}^N \tilde{\mathbf{x}}_{t+1} \tilde{\mathbf{x}}_{t+1}^T \mathbb{1}_{\{\theta_{t+1}=j\}} f(\tilde{\mathbf{x}}_{t+1}, \theta_{t+1} = r) d\tilde{\mathbf{x}}_{t+1} \\
 &= \int_{\mathbb{R}^{2n_\xi}} \int_{\mathbb{R}^{2n_\xi}} \int_{\mathbb{R}^{2n_\xi}} \sum_{r=0}^N \sum_{i=0}^N \tilde{\mathbf{x}}_{t+1} \tilde{\mathbf{x}}_{t+1}^T \mathbb{1}_{\{\theta_{t+1}=j\}} f(\tilde{\mathbf{x}}_{t+1}, \tilde{\mathbf{x}}_t, \underline{o}_t, \theta_{t+1} = r, \theta_t = i) d\tilde{\mathbf{x}}_{t+1} d\tilde{\mathbf{x}}_t d\underline{o}_t \\
 &= \int_{\mathbb{R}^{2n_\xi}} \int_{\mathbb{R}^{2n_\xi}} \int_{\mathbb{R}^{2n_\xi}} \sum_{r=0}^N \sum_{i=0}^N \tilde{\mathbf{x}}_{t+1} \tilde{\mathbf{x}}_{t+1}^T \mathbb{1}_{\{\theta_{t+1}=j\}} \\
 &\quad \cdot t_{ir} f(\tilde{\mathbf{x}}_{t+1} \mid \tilde{\mathbf{x}}_t, \theta_t = i, \underline{o}_t) f(\tilde{\mathbf{x}}_t, \theta_t = i) f(\underline{o}_t) d\tilde{\mathbf{x}}_{t+1} d\tilde{\mathbf{x}}_t d\underline{o}_t.
 \end{aligned}$$

From (5.44), we have that for given realizations of  $\tilde{\mathbf{x}}_t$  and  $\underline{o}_t$  it holds

$$f(\tilde{\mathbf{x}}_{t+1} \mid \tilde{\mathbf{x}}_t, \theta_t = i, \underline{o}_t) = \delta \left( \tilde{\mathbf{x}}_{t+1} - \left( \mathbf{\Gamma}_t^{(i)} \tilde{\mathbf{x}}_t + \underline{o}_t \right) \right),$$

for any  $i = 0, 1, \dots, N$ . Hence, exploiting the sifting property of the Dirac delta function, we get

$$\begin{aligned}
 \tilde{\mathbf{X}}_{t+1}^{(j)} &= \int_{\mathbb{R}^{2n_\xi}} \int_{\mathbb{R}^{2n_\xi}} \int_{\mathbb{R}^{2n_\xi}} \sum_{r=0}^N \sum_{i=0}^N \tilde{\mathbf{x}}_{t+1} \tilde{\mathbf{x}}_{t+1}^T \mathbb{1}_{\{\theta_{t+1}=j\}} \\
 &\quad \cdot t_{ir} \delta \left( \tilde{\mathbf{x}}_{t+1} - \left( \mathbf{\Gamma}_t^{(i)} \tilde{\mathbf{x}}_t + \underline{o}_t \right) \right) f(\tilde{\mathbf{x}}_t, \theta_t = i) f(\underline{o}_t) d\tilde{\mathbf{x}}_{t+1} d\tilde{\mathbf{x}}_t d\underline{o}_t \\
 &= \int_{\mathbb{R}^{2n_\xi}} \int_{\mathbb{R}^{2n_\xi}} \sum_{r=0}^N \sum_{i=0}^N \left( \mathbf{\Gamma}_t^{(i)} \tilde{\mathbf{x}}_t + \underline{o}_t \right) \left( \mathbf{\Gamma}_t^{(i)} \tilde{\mathbf{x}}_t + \underline{o}_t \right)^T \mathbb{1}_{\{\theta_{t+1}=j\}} t_{ir} f(\tilde{\mathbf{x}}_t, \theta_t = i) f(\underline{o}_t) d\tilde{\mathbf{x}}_t d\underline{o}_t \\
 &= \int_{\mathbb{R}^{2n_\xi}} \int_{\mathbb{R}^{2n_\xi}} \sum_{i=0}^N \left( \mathbf{\Gamma}_t^{(i)} \tilde{\mathbf{x}}_t + \underline{o}_t \right) \left( \mathbf{\Gamma}_t^{(i)} \tilde{\mathbf{x}}_t + \underline{o}_t \right)^T t_{ij} f(\tilde{\mathbf{x}}_t, \theta_t = i) f(\underline{o}_t) d\tilde{\mathbf{x}}_t d\underline{o}_t,
 \end{aligned}$$

which gives (5.49). □

## C.3 Proof of Theorem 5.1

### Theorem 5.1:

Fix a sequence of control law parameters  $(\mathbf{K}_0, \mathbf{L}_0), \dots, (\mathbf{K}_{K-1}, \mathbf{L}_{K-1})$ . Then, the cost-to-go at each stage  $t = 0, 1, \dots, K$  is given by

$$\mathcal{V}_t = \sum_{i=0}^N \text{tr} \left[ \tilde{\mathbf{P}}_t^{(i)} \tilde{\mathbf{X}}_t^{(i)} \right] + \mu_t^{(i)} \tilde{\omega}_t^{(i)}, \quad (5.53)$$

with  $\tilde{\mathbf{P}}_t^{(i)}$  and  $\tilde{\omega}_t^{(i)}$  computed according to the backward recursions

$$\tilde{\mathbf{P}}_t^{(i)} = \hat{\mathbf{Q}}_t^{(i)} + \left( \mathbf{\Gamma}_t^{(i)} \right)^T \mathcal{E}^{(i)} \left( \tilde{\mathcal{P}}_{t+1} \right) \mathbf{\Gamma}_t^{(i)}, \quad (5.54)$$

$$\tilde{\omega}_t^{(i)} = \mathcal{E}^{(i)} \left( \tilde{\omega}_{t+1} \right) + \text{tr} \left[ \mathcal{E}^{(i)} \left( \tilde{\mathcal{P}}_{t+1} \right) \mathbf{N}_t \right], \quad (5.55)$$

that are initialized with  $\tilde{\mathbf{P}}_K^{(i)} = \hat{\mathbf{Q}}_K$  and  $\tilde{\omega}_K^{(i)} = 0$  for  $i = 0, 1, \dots, N$ , and where

$$\begin{aligned}\tilde{\mathcal{P}}_t &= \left( \tilde{\mathbf{P}}_t^{(0)}, \tilde{\mathbf{P}}_t^{(1)}, \dots, \tilde{\mathbf{P}}_t^{(N)} \right) \in \mathbb{H}^{2n\xi}, \\ \tilde{\omega}_t &= \left( \tilde{\omega}_t^{(0)}, \tilde{\omega}_t^{(1)}, \dots, \tilde{\omega}_t^{(N)} \right) \in \mathbb{H}^1.\end{aligned}$$

*Proof.* We will prove the theorem by induction over “tail subproblems” of increasing length. It follows from (5.52) that the cost-to-go for the subproblem that starts at stage  $t = t'$  is given by

$$\begin{aligned}\mathcal{V}_{t'} &= \sum_{j=0}^N \text{tr} \left[ \hat{\mathbf{Q}}_K \tilde{\mathbf{X}}_K^{(j)} \right] + \sum_{t=t'}^{K-1} \sum_{j=0}^N \text{tr} \left[ \hat{\mathbf{Q}}_t^{(j)} \tilde{\mathbf{X}}_t^{(j)} \right] \\ &= \sum_{j=0}^N \text{tr} \left[ \hat{\mathbf{Q}}_K \tilde{\mathbf{X}}_K^{(j)} \right] + \sum_{t=t'+1}^{K-1} \sum_{j=0}^N \text{tr} \left[ \hat{\mathbf{Q}}_t^{(j)} \tilde{\mathbf{X}}_t^{(j)} \right] + \sum_{j=0}^N \text{tr} \left[ \hat{\mathbf{Q}}_{t'}^{(j)} \tilde{\mathbf{X}}_{t'}^{(j)} \right] \\ &= \mathcal{V}_{t'+1} + \sum_{j=0}^N \text{tr} \left[ \hat{\mathbf{Q}}_{t'}^{(j)} \tilde{\mathbf{X}}_{t'}^{(j)} \right].\end{aligned}\tag{C.5}$$

For the first subproblem that starts at the terminal stage  $t = K$ , we directly get from (C.5) that

$$\mathcal{V}_K = \sum_{i=0}^N \text{tr} \left[ \hat{\mathbf{Q}}_K \tilde{\mathbf{X}}_K^{(i)} \right] = \sum_{i=0}^N \text{tr} \left[ \tilde{\mathbf{P}}_K^{(i)} \tilde{\mathbf{X}}_K^{(i)} \right].$$

Now let (5.53) hold for the subproblem that starts at stage  $t' + 1$  for some  $0 \leq t' < K - 1$ , i.e., let it hold

$$\mathcal{V}_{t'+1} = \sum_{i=0}^N \text{tr} \left[ \tilde{\mathbf{P}}_{t'+1}^{(i)} \tilde{\mathbf{X}}_{t'+1}^{(i)} \right] + \mu_{t'+1}^{(i)} \tilde{\omega}_{t'+1}^{(i)}.$$

Then, it follows from (C.5) for the subproblem that starts at stage  $t = t'$

$$\begin{aligned}\mathcal{V}_{t'} &= \sum_{j=0}^N \left( \text{tr} \left[ \tilde{\mathbf{P}}_{t'+1}^{(j)} \tilde{\mathbf{X}}_{t'+1}^{(j)} \right] + \mu_{t'+1}^{(j)} \tilde{\omega}_{t'+1}^{(j)} \right) + \sum_{j=0}^N \text{tr} \left[ \hat{\mathbf{Q}}_{t'}^{(j)} \tilde{\mathbf{X}}_{t'}^{(j)} \right] \\ &= \sum_{j=0}^N \left( \text{tr} \left[ \tilde{\mathbf{P}}_{t'+1}^{(j)} \tilde{\mathbf{X}}_{t'+1}^{(j)} + \hat{\mathbf{Q}}_{t'}^{(j)} \tilde{\mathbf{X}}_{t'}^{(j)} \right] + \sum_{i=0}^N t_{ij} \mu_{t'}^{(i)} \tilde{\omega}_{t'+1}^{(j)} \right).\end{aligned}$$

Plugging in the dynamics (5.49) then leads to

$$\begin{aligned}\mathcal{V}_{t'} &= \sum_{j=0}^N \left( \text{tr} \left[ \sum_{i=0}^N t_{ij} \tilde{\mathbf{P}}_{t'+1}^{(j)} \left( \mathbf{\Gamma}_{t'}^{(i)} \tilde{\mathbf{X}}_{t'}^{(i)} \left( \mathbf{\Gamma}_{t'}^{(i)} \right)^\top + \mu_{t'}^{(i)} \mathbf{N}_{t'} \right) + \hat{\mathbf{Q}}_{t'}^{(j)} \tilde{\mathbf{X}}_{t'}^{(j)} \right] + \sum_{i=0}^N t_{ij} \mu_{t'}^{(i)} \tilde{\omega}_{t'+1}^{(j)} \right) \\ &= \sum_{i=0}^N \left( \text{tr} \left[ \left( \sum_{j=0}^N t_{ij} \left( \mathbf{\Gamma}_{t'}^{(i)} \right)^\top \tilde{\mathbf{P}}_{t'+1}^{(j)} \mathbf{\Gamma}_{t'}^{(i)} \right) \tilde{\mathbf{X}}_{t'}^{(i)} + \hat{\mathbf{Q}}_{t'}^{(i)} \tilde{\mathbf{X}}_{t'}^{(i)} \right] + \mu_{t'}^{(i)} \sum_{j=0}^N t_{ij} \tilde{\mathbf{P}}_{t'+1}^{(j)} \mathbf{N}_{t'} + t_{ij} \tilde{\omega}_{t'+1}^{(j)} \right) \\ &= \sum_{i=0}^N \text{tr} \left[ \left( \left( \mathbf{\Gamma}_{t'}^{(i)} \right)^\top \mathcal{E}^{(i)} \left( \tilde{\mathcal{P}}_{t'+1} \right) \mathbf{\Gamma}_{t'}^{(i)} + \hat{\mathbf{Q}}_{t'}^{(i)} \right) \tilde{\mathbf{X}}_{t'}^{(i)} \right] + \mu_{t'}^{(i)} \left( \text{tr} \left[ \mathcal{E}^{(i)} \left( \tilde{\mathcal{P}}_{t'+1} \right) \mathbf{N}_{t'} \right] + \mathcal{E}^{(i)} \left( \tilde{\omega}_{t'+1} \right) \right) \\ &= \sum_{i=0}^N \text{tr} \left[ \tilde{\mathbf{P}}_{t'}^{(i)} \tilde{\mathbf{X}}_{t'}^{(i)} \right] + \mu_{t'}^{(i)} \tilde{\omega}_{t'}^{(i)},\end{aligned}$$

which concludes the proof.  $\square$

## C.4 Proof of Lemma 5.3

### Lemma 5.3:

Let  $\tilde{\mathbf{X}}_t^{(i)}$  be parameterized according to (5.65) and let (5.66) hold. Then, the dynamics of  $\underline{\mathbf{X}}_t^{(i)}$  and  $\overline{\mathbf{X}}_t^{(i)}$  are given by

$$\begin{aligned}
\underline{\mathbf{X}}_{t+1}^{(j)} &= \sum_{i=0}^N t_{ij} \left[ \mu_t^{(i)} \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{V}} \left( \mathbf{K}_t \hat{\mathbf{S}}_t \right)^\top \right. \\
&\quad \left. + \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t \right) \underline{\mathbf{X}}_t^{(i)} \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t \right)^\top + \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \overline{\mathbf{X}}_t^{(i)} \left( \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \right], \\
\overline{\mathbf{X}}_{t+1}^{(j)} &= \sum_{i=0}^N t_{ij} \left[ \mu_t^{(i)} \left( \overline{\mathbf{W}} + \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{V}} \left( \mathbf{K}_t \hat{\mathbf{S}}_t \right)^\top \right) \right. \\
&\quad \left. + \left( \overline{\mathbf{A}}_t^{(i)} - \hat{\mathbf{A}}_t + \left( \overline{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t \right) \mathbf{L}_t \right) \underline{\mathbf{X}}_t^{(i)} \left( \overline{\mathbf{A}}_t^{(i)} - \hat{\mathbf{A}}_t + \left( \overline{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t \right) \mathbf{L}_t \right)^\top \right. \\
&\quad \left. + \left( \overline{\mathbf{A}}_t^{(i)} - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right) \overline{\mathbf{X}}_t^{(i)} \left( \overline{\mathbf{A}}_t^{(i)} - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \right], \tag{5.67}
\end{aligned}$$

for  $t = 0, 1, \dots, K$  and  $j = 0, 1, \dots, N$ .

*Proof.* To obtain the dynamics of  $\underline{\mathbf{X}}_t^{(i)}$  and  $\overline{\mathbf{X}}_t^{(i)}$ , we first evaluate (5.49) by plugging in the definitions (5.45) and (5.48) of  $\tilde{\Gamma}_t^{(i)}$  and  $\mathbf{N}_t$ , which yields for dynamics of the blocks of  $\tilde{\mathbf{X}}_t^{(i)}$

$$\begin{aligned}
\tilde{\mathbf{X}}_{t+1,1}^{(j)} &= \sum_{i=0}^N t_{ij} \left[ \mu_t^{(i)} \overline{\mathbf{W}} + \overline{\mathbf{A}}_t^{(i)} \tilde{\mathbf{X}}_{t,1}^{(i)} \left( \overline{\mathbf{A}}_t^{(i)} \right)^\top + \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \tilde{\mathbf{X}}_{t,2}^{(i)} \left( \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \right)^\top \right. \\
&\quad \left. + \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \left( \tilde{\mathbf{X}}_{t,12}^{(i)} \right)^\top \left( \overline{\mathbf{A}}_t^{(i)} \right)^\top + \overline{\mathbf{A}}_t^{(i)} \tilde{\mathbf{X}}_{t,12}^{(i)} \left( \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \right)^\top \right], \\
\tilde{\mathbf{X}}_{t+1,12}^{(j)} &= \sum_{i=0}^N t_{ij} \left[ \left( \overline{\mathbf{A}}_t^{(i)} \tilde{\mathbf{X}}_{t,1}^{(i)} + \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \left( \tilde{\mathbf{X}}_{t,12}^{(i)} \right)^\top \right) \left( \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \right. \\
&\quad \left. + \left( \overline{\mathbf{A}}_t^{(i)} \tilde{\mathbf{X}}_{t,12}^{(i)} + \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \tilde{\mathbf{X}}_{t,2}^{(i)} \right) \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \right], \\
\tilde{\mathbf{X}}_{t+1,2}^{(j)} &= \sum_{i=0}^N t_{ij} \left[ \mu_t^{(i)} \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{V}} \left( \mathbf{K}_t \hat{\mathbf{S}}_t \right)^\top + \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \tilde{\mathbf{X}}_{t,1}^{(i)} \left( \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \right. \\
&\quad \left. + \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right) \tilde{\mathbf{X}}_{t,2}^{(i)} \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \right. \\
&\quad \left. + \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right) \left( \tilde{\mathbf{X}}_{t,12}^{(i)} \right)^\top \left( \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \right. \\
&\quad \left. + \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \tilde{\mathbf{X}}_{t,12}^{(i)} \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \right],
\end{aligned}$$

for  $j = 0, 1, \dots, N$ . Then, we compute  $\overline{\mathbf{X}}_{t+1}^{(j)} = \tilde{\mathbf{X}}_{t+1,1}^{(j)} + \tilde{\mathbf{X}}_{t+1,2}^{(j)} - \tilde{\mathbf{X}}_{t+1,12}^{(j)} - \left( \tilde{\mathbf{X}}_{t+1,12}^{(j)} \right)^\top$  and after replacing all occurrences of  $\tilde{\mathbf{X}}_{t,1}^{(i)}$  by  $\underline{\mathbf{X}}_t^{(i)} + \overline{\mathbf{X}}_t^{(i)}$ , all occurrences of  $\tilde{\mathbf{X}}_{t+1,2}^{(i)}$  by  $\underline{\mathbf{X}}_{t+1}^{(i)}$ , and all occurrences of  $\tilde{\mathbf{X}}_{t,12}^{(i)}$  and  $\tilde{\mathbf{X}}_{t,2}^{(i)}$  by  $\underline{\mathbf{X}}_t^{(i)}$  according to (5.66), we arrive at (5.67).  $\square$

## C.5

 Proof of Theorem 5.2

### Theorem 5.2:

Fix a sequence of control law parameters  $(\mathbf{K}_0, \mathbf{L}_0), \dots, (\mathbf{K}_{K-1}, \mathbf{L}_{K-1})$  and let  $\tilde{\mathbf{X}}_t^{(i)}$  and  $\tilde{\mathbf{P}}_t^{(i)}$  be parameterized as per (5.65) and (5.70), respectively. Further, let denote

$$\begin{aligned}\underline{\mathcal{P}}_t &= \left( \underline{\mathbf{P}}_t^{(0)}, \underline{\mathbf{P}}_t^{(1)}, \dots, \underline{\mathbf{P}}_t^{(N)} \right) \in \mathbb{H}^{n_\xi}, \\ \overline{\mathcal{P}}_t &= \left( \overline{\mathbf{P}}_t^{(0)}, \overline{\mathbf{P}}_t^{(1)}, \dots, \overline{\mathbf{P}}_t^{(N)} \right) \in \mathbb{H}^{n_\xi}, \\ \overline{\omega}_t &= \left( \overline{\omega}_t^{(0)}, \overline{\omega}_t^{(1)}, \dots, \overline{\omega}_t^{(N)} \right) \in \mathbb{H}^1.\end{aligned}$$

Then, the cost-to-go at each stage  $t = 0, 1, \dots, K$  is given by

$$\overline{V}_t = \sum_{i=0}^N \text{tr} \left[ \underline{\mathbf{P}}_t^{(i)} \overline{\mathbf{X}}_t^{(i)} + \overline{\mathbf{P}}_t^{(i)} \left( \underline{\mathbf{X}}_t^{(i)} + \overline{\mathbf{X}}_t^{(i)} \right) \right] + \mu_t^{(i)} \overline{\omega}_t^{(i)}, \quad (5.72)$$

with  $\underline{\mathbf{X}}_t^{(i)}$  and  $\overline{\mathbf{X}}_t^{(i)}$  given by Lemma 5.3, and where  $\underline{\mathbf{P}}_t^{(i)}$ ,  $\overline{\mathbf{P}}_t^{(i)}$ , and  $\overline{\omega}_t^{(i)}$  are computed by means of the backward recursions

$$\begin{aligned}\underline{\mathbf{P}}_t^{(i)} &= \left( \mathbf{J}^{(i)} \mathbf{L}_t \right)^\top \mathbf{R}_t \mathbf{J}^{(i)} \mathbf{L}_t + \left( \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \right)^\top \mathcal{E}^{(i)}(\overline{\mathcal{P}}_{t+1}) \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \\ &\quad + \left( \hat{\mathbf{A}}_t + \left( \hat{\mathbf{B}}_t - \overline{\mathbf{B}}_t^{(i)} \right) \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}) \left( \hat{\mathbf{A}}_t + \left( \hat{\mathbf{B}}_t - \overline{\mathbf{B}}_t^{(i)} \right) \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right), \\ \overline{\mathbf{P}}_t^{(i)} &= \overline{\mathbf{Q}}_t^{(i)} + \left( \mathbf{J}^{(i)} \mathbf{L}_t \right)^\top \mathbf{R}_t \mathbf{J}^{(i)} \mathbf{L}_t + \left( \overline{\mathbf{A}}_t^{(i)} + \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \right)^\top \mathcal{E}^{(i)}(\overline{\mathcal{P}}_{t+1}) \left( \overline{\mathbf{A}}_t^{(i)} + \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \right) \\ &\quad + \left( \overline{\mathbf{A}}_t^{(i)} - \hat{\mathbf{A}}_t + \left( \overline{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t \right) \mathbf{L}_t \right)^\top \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}) \left( \overline{\mathbf{A}}_t^{(i)} - \hat{\mathbf{A}}_t + \left( \overline{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t \right) \mathbf{L}_t \right), \\ \overline{\omega}_t^{(i)} &= \mathcal{E}^{(i)}(\overline{\omega}_{t+1}) + \text{tr} \left[ \mathcal{E}^{(i)}(\overline{\mathcal{P}}_{t+1} + \underline{\mathcal{P}}_{t+1}) \overline{\mathbf{W}} + \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}) \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{V}} \left( \mathbf{K}_t \hat{\mathbf{S}}_t \right)^\top \right],\end{aligned} \quad (5.73)$$

that are initialized with  $\underline{\mathbf{P}}_K^{(i)} = \mathbf{0}_{n_\xi}$ ,  $\overline{\mathbf{P}}_K^{(i)} = \overline{\mathbf{Q}}_K$ , and  $\overline{\omega}_K^{(i)} = 0$  for  $i = 0, 1, \dots, N$ .

*Proof.* We first evaluate the dynamics (5.54) of  $\tilde{\mathbf{P}}_t^{(i)}$  by plugging in the definitions (5.45) and (5.50) of  $\mathbf{\Gamma}_t^{(i)}$  and  $\hat{\mathbf{Q}}_t^{(i)}$ , which yields for its blocks

$$\begin{aligned}\tilde{\mathbf{P}}_{t,1}^{(i)} &= \overline{\mathbf{Q}}_t^{(i)} + \left( \overline{\mathbf{A}}_t^{(i)} \right)^\top \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,1}) \overline{\mathbf{A}}_t^{(i)} + \left( \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,2}) \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \\ &\quad + \left( \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \left( \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,12}) \right)^\top \overline{\mathbf{A}}_t^{(i)} + \left( \overline{\mathbf{A}}_t^{(i)} \right)^\top \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,12}) \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}}, \\ \tilde{\mathbf{P}}_{t,12}^{(i)} &= \left( \overline{\mathbf{A}}_t^{(i)} \right)^\top \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,1}) \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t + \left( \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,2}) \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right) \\ &\quad + \left( \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \left( \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,12}) \right)^\top \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \\ &\quad + \left( \overline{\mathbf{A}}_t^{(i)} \right)^\top \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,12}) \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right), \\ \tilde{\mathbf{P}}_{t,2}^{(i)} &= \left( \mathbf{J}^{(i)} \mathbf{L}_t \right)^\top \mathbf{R}_t \mathbf{J}^{(i)} \mathbf{L}_t + \left( \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \right)^\top \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,1}) \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \\ &\quad + \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,2}) \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right) \\ &\quad + \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right)^\top \left( \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,12}) \right)^\top \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \\ &\quad + \left( \overline{\mathbf{B}}_t^{(i)} \mathbf{L}_t \right)^\top \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,12}) \left( \hat{\mathbf{A}}_t + \hat{\mathbf{B}}_t \mathbf{L}_t - \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{C}} \right),\end{aligned}$$

for  $i = 0, 1, \dots, N$ , with

$$\begin{aligned}\tilde{\mathcal{P}}_{t,1} &= \left( \tilde{\mathbf{P}}_{t,1}^{(0)}, \tilde{\mathbf{P}}_{t,1}^{(1)}, \dots, \tilde{\mathbf{P}}_{t,1}^{(N)} \right) \in \mathbb{H}^{n_\xi}, \\ \tilde{\mathcal{P}}_{t,12} &= \left( \tilde{\mathbf{P}}_{t,12}^{(0)}, \tilde{\mathbf{P}}_{t,12}^{(1)}, \dots, \tilde{\mathbf{P}}_{t,12}^{(N)} \right) \in \mathbb{H}^{n_\xi}, \\ \tilde{\mathcal{P}}_{t,2} &= \left( \tilde{\mathbf{P}}_{t,2}^{(0)}, \tilde{\mathbf{P}}_{t,2}^{(1)}, \dots, \tilde{\mathbf{P}}_{t,2}^{(N)} \right) \in \mathbb{H}^{n_\xi}.\end{aligned}$$

Similarly, the evaluation of (5.55) using the definition of  $\mathbf{N}_t$  (5.48) results in

$$\tilde{\omega}_t^{(i)} = \mathcal{E}^{(i)}(\tilde{\omega}_{t+1}) + \text{tr} \left[ \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,1}) \overline{\mathbf{W}} + \mathcal{E}^{(i)}(\tilde{\mathcal{P}}_{t+1,2}) \mathbf{K}_t \hat{\mathbf{S}}_t \overline{\mathbf{V}} \left( \mathbf{K}_t \hat{\mathbf{S}}_t \right)^\top \right],$$

for  $i = 0, 1, \dots, N$ . Then, we compute  $\overline{\mathbf{P}}_t^{(i)} = \tilde{\mathbf{P}}_{t,1}^{(i)} + \tilde{\mathbf{P}}_{t,2}^{(i)} + \tilde{\mathbf{P}}_{t,12}^{(i)} + \left( \tilde{\mathbf{P}}_{t,12}^{(i)} \right)^\top$  and after setting  $\tilde{\mathbf{P}}_{t,2}^{(i)} = \underline{\mathbf{P}}_t^{(i)}$ ,  $\tilde{\mathbf{P}}_{t+1,1}^{(i)} = \underline{\mathbf{P}}_{t+1}^{(i)} + \overline{\mathbf{P}}_{t+1}^{(i)}$ ,  $\tilde{\mathbf{P}}_{t+1,2}^{(i)} = \underline{\mathbf{P}}_{t+1}^{(i)}$ , and  $\tilde{\mathbf{P}}_{t+1,12}^{(i)} = -\underline{\mathbf{P}}_{t+1}^{(i)}$  according to (5.71), we arrive at (5.73). Finally, we obtain the expression (5.72) for the cost-to-go by direct evaluation of (5.53) using the parametrizations (5.65) and (5.70) of  $\tilde{\mathbf{X}}_t^{(i)}$  and  $\tilde{\mathbf{P}}_t^{(i)}$ .  $\square$

## C.6 Proof of Lemma 5.4

### Lemma 5.4:

$\overline{\mathbf{V}}_t$  is convex with regards to  $(\mathbf{K}_t, \mathbf{L}_t)$ .

*Proof.* Note that  $\mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1})$ ,  $\mathbf{D}_{t,1}^{(i)}$ ,  $\mathbf{D}_{t,2}^{(i)}$ , and  $\underline{\mathbf{X}}_t^{(i)}$  are positive semidefinite for  $i = 0, 1, \dots, N$ . Since any positive semidefinite matrix  $\mathbf{Z}$  can be written as  $\mathbf{Z} = \tilde{\mathbf{Z}}^\top \tilde{\mathbf{Z}}$  for some  $\tilde{\mathbf{Z}}$  (cf. Theorem A.6 in Appendix A.4), we can write

$$\begin{aligned}\mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}) &= \left( \mathbf{E}_{t,1}^{(i)} \right)^\top \mathbf{E}_{t,1}^{(i)}, & \mathbf{D}_{t,1}^{(i)} &= \left( \mathbf{E}_{t,2}^{(i)} \right)^\top \mathbf{E}_{t,2}^{(i)}, \\ \mathbf{D}_{t,2}^{(i)} &= \left( \mathbf{E}_{t,3}^{(i)} \right)^\top \mathbf{E}_{t,3}^{(i)}, & \underline{\mathbf{X}}_t^{(i)} &= \left( \mathbf{E}_{t,4}^{(i)} \right)^\top \mathbf{E}_{t,4}^{(i)}.\end{aligned}$$

Then, (5.74) becomes

$$\begin{aligned}\overline{\mathbf{V}}_t &= \sum_{i=0}^N \text{tr} \left[ \mathbf{K}_t^\top \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}) \mathbf{K}_t \mathbf{D}_{t,1}^{(i)} \right] + \text{tr} \left[ \mathbf{L}_t^\top \mathbf{D}_{t,2}^{(i)} \mathbf{L}_t \underline{\mathbf{X}}_t^{(i)} \right] \\ &\quad + 2 \text{tr} \left[ \left( \mathbf{D}_{t,3}^{(i)} \right)^\top \mathbf{K}_t \right] + 2 \text{tr} \left[ \left( \mathbf{D}_{t,4}^{(i)} \right)^\top \mathbf{L}_t \right] + c_t^{(i)} \\ &= \sum_{i=0}^N \text{tr} \left[ \mathbf{K}_t^\top \left( \mathbf{E}_{t,1}^{(i)} \right)^\top \mathbf{E}_{t,1}^{(i)} \mathbf{K}_t \left( \mathbf{E}_{t,2}^{(i)} \right)^\top \mathbf{E}_{t,2}^{(i)} \right] + \text{tr} \left[ \mathbf{L}_t^\top \left( \mathbf{E}_{t,3}^{(i)} \right)^\top \mathbf{E}_{t,3}^{(i)} \mathbf{L}_t \left( \mathbf{E}_{t,4}^{(i)} \right)^\top \mathbf{E}_{t,4}^{(i)} \right] \\ &\quad + 2 \text{tr} \left[ \left( \mathbf{D}_{t,3}^{(i)} \right)^\top \mathbf{K}_t \right] + 2 \text{tr} \left[ \left( \mathbf{D}_{t,4}^{(i)} \right)^\top \mathbf{L}_t \right] + c_t^{(i)} \\ &= \sum_{i=0}^N \text{tr} \left[ \mathbf{E}_{t,2}^{(i)} \mathbf{K}_t^\top \left( \mathbf{E}_{t,1}^{(i)} \right)^\top \mathbf{E}_{t,1}^{(i)} \mathbf{K}_t \left( \mathbf{E}_{t,2}^{(i)} \right)^\top \right] + \text{tr} \left[ \mathbf{E}_{t,4}^{(i)} \mathbf{L}_t^\top \left( \mathbf{E}_{t,3}^{(i)} \right)^\top \mathbf{E}_{t,3}^{(i)} \mathbf{L}_t \left( \mathbf{E}_{t,4}^{(i)} \right)^\top \right] \\ &\quad + 2 \text{tr} \left[ \left( \mathbf{D}_{t,3}^{(i)} \right)^\top \mathbf{K}_t \right] + 2 \text{tr} \left[ \left( \mathbf{D}_{t,4}^{(i)} \right)^\top \mathbf{L}_t \right] + c_t^{(i)} \\ &= \sum_{i=0}^N \left\| \tilde{\mathbf{E}}_{t,1}^{(i)} \right\|_F^2 + \left\| \tilde{\mathbf{E}}_{t,2}^{(i)} \right\|_F^2 + 2 \text{tr} \left[ \left( \mathbf{D}_{t,3}^{(i)} \right)^\top \mathbf{K}_t \right] + 2 \text{tr} \left[ \left( \mathbf{D}_{t,4}^{(i)} \right)^\top \mathbf{L}_t \right] + c_t^{(i)},\end{aligned}$$

with  $\|\cdot\|_F$  the Frobenius norm (A.2) and

$$\tilde{\mathbf{E}}_{t,1}^{(i)} = \mathbf{E}_{t,1}^{(i)} \mathbf{K}_t \left( \mathbf{E}_{t,2}^{(i)} \right)^\top, \quad \tilde{\mathbf{E}}_{t,2}^{(i)} = \mathbf{E}_{t,3}^{(i)} \mathbf{L}_t \left( \mathbf{E}_{t,4}^{(i)} \right)^\top.$$

Since  $\tilde{\mathbf{E}}_{t,1}^{(i)}$  and  $\tilde{\mathbf{E}}_{t,2}^{(i)}$  are linear functions of  $\mathbf{K}_t$  and  $\mathbf{L}_t$ , the result follows from the convexity of  $\|\cdot\|_F$ .  $\square$

## C.7 Proof of Theorem 5.3

### Theorem 5.3:

Let  $(\mathbf{K}_0^{[c]}, \mathbf{L}_0^{[c]}), \dots, (\mathbf{K}_{K-1}^{[c]}, \mathbf{L}_{K-1}^{[c]})$  be the control law parameters at the end of iteration  $c$  in Algorithm 5.4, i.e., after the completion of the backward pass of the algorithm. Then, it holds  $\bar{\mathcal{V}}_0^{[c]} \leq \bar{\mathcal{V}}_0^{[c-1]}$  and, moreover,  $(\mathbf{K}_0, \mathbf{L}_0), \dots, (\mathbf{K}_{K-1}, \mathbf{L}_{K-1}) = \lim_{c \rightarrow \infty} (\mathbf{K}_0^{[c]}, \mathbf{L}_0^{[c]}), \dots, (\mathbf{K}_{K-1}^{[c]}, \mathbf{L}_{K-1}^{[c]})$  exists.

*Proof.* The proof works similar to the proofs of related results presented in [148, 177, 178, 198, 200]. We calculate the cost-to-go  $\bar{\mathcal{V}}_t^{[c]}$  at stage  $t$  of iteration  $c$  after the computation of  $(\mathbf{K}_t^{[c]}, \mathbf{L}_t^{[c]})$  in line 20 of Algorithm 5.4 and the cost-to-go  $\bar{\mathcal{V}}_t^{[c]}$  that would have resulted if  $(\mathbf{K}_t^{[c-1]}, \mathbf{L}_t^{[c-1]})$  had been reused. We show that the difference  $\bar{\mathcal{V}}_t^{[c]} - \bar{\mathcal{V}}_t^{[c-1]}$  is always nonnegative for all stages  $t$ , implying a decrease of the cost. Since the cost are always nonnegative, the limit  $\lim_{c \rightarrow \infty} \bar{\mathcal{V}}_0^{[c]}$  exists and the convergence of the proposed method follows.

For a formal proof, let us first assume that iteration  $c - 1$  of the algorithm has already completed. Let the corresponding control law parameters be  $(\mathbf{K}_0^{[c-1]}, \mathbf{L}_0^{[c-1]}), \dots, (\mathbf{K}_{K-1}^{[c-1]}, \mathbf{L}_{K-1}^{[c-1]})$ . Now consider the backward pass of iteration  $c$  at some stage  $t$ , prior to the computation of the control law parameters  $(\mathbf{K}_t^{[c]}, \mathbf{L}_t^{[c]})$  in line 20. From the current iteration, we already have  $\underline{\mathbf{X}}_{0:K}^{(i,[c])}$  and  $\bar{\mathbf{X}}_{0:K}^{(i,[c])}$  available from the forward pass. Similarly, during the backward pass,  $\underline{\mathbf{P}}_{t+1:K}^{(i,[c])}$ ,  $\bar{\mathbf{P}}_{t+1:K}^{(i,[c])}$ ,  $\bar{\omega}_{t+1:K}^{(i,[c])}$ , and the control law parameters  $(\mathbf{K}_{t+1}^{[c]}, \mathbf{L}_{t+1}^{[c]}), \dots, (\mathbf{K}_{K-1}^{[c]}, \mathbf{L}_{K-1}^{[c]})$  have already been already computed. From (5.74), we get for the cost-to-go  $\bar{\mathcal{V}}_t^{[c]}$  before we compute the control law parameters  $(\mathbf{K}_t^{[c]}, \mathbf{L}_t^{[c]})$

$$\begin{aligned} \bar{\mathcal{V}}_t^{[c]} &= \sum_{i=0}^N \text{tr} \left[ \left( \mathbf{K}_t^{[c-1]} \right)^\top \mathcal{E}^{(i)} \left( \underline{\mathbf{P}}_{t+1}^{[c]} \right) \mathbf{K}_t^{[c-1]} \mathbf{D}_{t,1}^{(i,[c])} \right] + \text{tr} \left[ \left( \mathbf{L}_t^{[c-1]} \right)^\top \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c-1]} \underline{\mathbf{X}}_t^{(i,[c])} \right] \\ &\quad + 2 \text{tr} \left[ \left( \mathbf{D}_{t,3}^{(i,[c])} \right)^\top \mathbf{K}_t^{[c-1]} \right] + 2 \text{tr} \left[ \left( \mathbf{D}_{t,4}^{(i,[c])} \right)^\top \mathbf{L}_t^{[c-1]} \right] + c_t^{(i,[c])}, \end{aligned}$$

with

$$\begin{aligned} \mathbf{D}_{t,1}^{(i,[c])} &= \hat{\mathbf{S}}_t \left( \mu_t^{(i)} \bar{\mathbf{V}} + \bar{\mathbf{C}} \bar{\mathbf{X}}_t^{(i,[c])} \bar{\mathbf{C}}^\top \right) \hat{\mathbf{S}}_t^\top, \\ \mathbf{D}_{t,2}^{(i,[c])} &= \left( \mathbf{J}^{(i)} \right)^\top \mathbf{R}_t \mathbf{J}^{(i)} + \left( \bar{\mathbf{B}}_t^{(i)} \right)^\top \mathcal{E}^{(i)} \left( \bar{\mathcal{P}}_{t+1}^{[c]} \right) \bar{\mathbf{B}}_t^{(i)} + \left( \bar{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t \right)^\top \mathcal{E}^{(i)} \left( \underline{\mathcal{P}}_{t+1}^{[c]} \right) \left( \bar{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t \right), \\ \mathbf{D}_{t,3}^{(i,[c])} &= -\mathcal{E}^{(i)} \left( \underline{\mathcal{P}}_{t+1}^{[c]} \right) \bar{\mathbf{A}}_t^{(i)} \bar{\mathbf{X}}_t^{(i,[c])} \left( \hat{\mathbf{S}}_t \bar{\mathbf{C}} \right)^\top, \\ \mathbf{D}_{t,4}^{(i,[c])} &= \left( \left( \bar{\mathbf{B}}_t^{(i)} - \hat{\mathbf{B}}_t \right)^\top \mathcal{E}^{(i)} \left( \underline{\mathcal{P}}_{t+1}^{[c]} \right) \left( \bar{\mathbf{A}}_t^{(i)} - \hat{\mathbf{A}}_t \right) + \left( \bar{\mathbf{B}}_t^{(i)} \right)^\top \mathcal{E}^{(i)} \left( \bar{\mathcal{P}}_{t+1}^{[c]} \right) \bar{\mathbf{A}}_t^{(i)} \right) \underline{\mathbf{X}}_t^{(i,[c])}, \end{aligned}$$

$c_t^{(i,[c])} \geq 0$  a constant that contains only terms independent of the control law parameters, and where

$$\begin{aligned} \underline{\mathcal{P}}_{t+1}^{[c]} &= \left( \underline{\mathbf{P}}_{t+1}^{(0,[c])}, \underline{\mathbf{P}}_{t+1}^{(1,[c])}, \dots, \underline{\mathbf{P}}_{t+1}^{(N,[c])} \right) \in \mathbb{H}^{n_\xi}, \\ \bar{\mathcal{P}}_{t+1}^{[c]} &= \left( \bar{\mathbf{P}}_{t+1}^{(0,[c])}, \bar{\mathbf{P}}_{t+1}^{(1,[c])}, \dots, \bar{\mathbf{P}}_{t+1}^{(N,[c])} \right) \in \mathbb{H}^{n_\xi}, \end{aligned}$$

and  $\bar{\omega}_{t+1}^{[c]} = (\bar{\omega}_{t+1}^{(0,[c])}, \bar{\omega}_{t+1}^{(1,[c])}, \dots, \bar{\omega}_{t+1}^{(N,[c])}) \in \mathbb{H}^1$ .  $\bar{V}_t'^{[c]}$  expresses the cost-to-go at stage  $t$  if we would skip line 20 of Algorithm 5.4 and reuse the control law parameters  $(\mathbf{K}_t^{[c-1]}, \mathbf{L}_t^{[c-1]})$  from the previous iteration. In contrast, after the execution of line 20 the cost-to-go is given by

$$\begin{aligned} \bar{V}_t^{[c]} = & \sum_{i=0}^N \text{tr} \left[ \left( \mathbf{K}_t^{[c]} \right)^T \mathcal{E}^{(i)} \left( \mathcal{P}_{t+1}^{[c]} \right) \mathbf{K}_t^{[c]} \mathbf{D}_{t,1}^{(i,[c])} \right] + \text{tr} \left[ \left( \mathbf{L}_t^{[c]} \right)^T \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c]} \underline{\mathbf{x}}_t^{(i,[c])} \right] \\ & + 2 \text{tr} \left[ \left( \mathbf{D}_{t,3}^{(i,[c])} \right)^T \mathbf{K}_t^{[c]} \right] + 2 \text{tr} \left[ \left( \mathbf{D}_{t,4}^{(i,[c])} \right)^T \mathbf{L}_t^{[c]} \right] + c_t^{(i,[c])}, \end{aligned}$$

which is the expression for  $\bar{V}_t'^{[c]}$  with  $(\mathbf{K}_t^{[c]}, \mathbf{L}_t^{[c]})$  used in place of  $(\mathbf{K}_t^{[c-1]}, \mathbf{L}_t^{[c-1]})$ . Their difference  $\Delta_t^{[c]} = \bar{V}_t'^{[c]} - \bar{V}_t^{[c]}$  is given by

$$\begin{aligned} \Delta_t^{[c]} = & \sum_{i=0}^N \text{tr} \left[ \left( \mathbf{K}_t^{[c-1]} \right)^T \mathcal{E}^{(i)} \left( \mathcal{P}_{t+1}^{[c]} \right) \mathbf{K}_t^{[c-1]} \mathbf{D}_{t,1}^{(i,[c])} \right] - \text{tr} \left[ \left( \mathbf{K}_t^{[c]} \right)^T \mathcal{E}^{(i)} \left( \mathcal{P}_{t+1}^{[c]} \right) \mathbf{K}_t^{[c]} \mathbf{D}_{t,1}^{(i,[c])} \right] \\ & + \text{tr} \left[ \left( \mathbf{L}_t^{[c-1]} \right)^T \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c-1]} \underline{\mathbf{x}}_t^{(i,[c])} \right] - \text{tr} \left[ \left( \mathbf{L}_t^{[c]} \right)^T \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c]} \underline{\mathbf{x}}_t^{(i,[c])} \right] \\ & + 2 \text{tr} \left[ \left( \mathbf{D}_{t,3}^{(i,[c])} \right)^T \mathbf{K}_t^{[c-1]} \right] - 2 \text{tr} \left[ \left( \mathbf{D}_{t,3}^{(i,[c])} \right)^T \mathbf{K}_t^{[c]} \right] \\ & + 2 \text{tr} \left[ \left( \mathbf{D}_{t,4}^{(i,[c])} \right)^T \mathbf{L}_t^{[c-1]} \right] - 2 \text{tr} \left[ \left( \mathbf{D}_{t,4}^{(i,[c])} \right)^T \mathbf{L}_t^{[c]} \right]. \end{aligned}$$

The updated control law parameters  $(\mathbf{K}_t^{[c]}, \mathbf{L}_t^{[c]})$  fulfill the optimality conditions (5.75) and (5.76) , i.e., it holds

$$\begin{aligned} \sum_{i=0}^N \mathcal{E}^{(i)} \left( \mathcal{P}_{t+1}^{[c]} \right) \mathbf{K}_t^{[c]} \mathbf{D}_{t,1}^{(i,[c])} &= - \sum_{i=0}^N \mathbf{D}_{t,3}^{(i,[c])}, \\ \sum_{i=0}^N \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c]} \underline{\mathbf{x}}_t^{(i,[c])} &= - \sum_{i=0}^N \mathbf{D}_{t,4}^{(i,[c])}. \end{aligned}$$

With these identities, we can rewrite the last four terms in  $\Delta_t^{[c]}$  to obtain

$$\begin{aligned} \Delta_t^{[c]} = & \sum_{i=0}^N \text{tr} \left[ \left( \mathbf{K}_t^{[c-1]} \right)^T \mathcal{E}^{(i)} \left( \mathcal{P}_{t+1}^{[c]} \right) \mathbf{K}_t^{[c-1]} \mathbf{D}_{t,1}^{(i,[c])} \right] - \text{tr} \left[ \left( \mathbf{K}_t^{[c]} \right)^T \mathcal{E}^{(i)} \left( \mathcal{P}_{t+1}^{[c]} \right) \mathbf{K}_t^{[c]} \mathbf{D}_{t,1}^{(i,[c])} \right] \\ & + \text{tr} \left[ \left( \mathbf{L}_t^{[c-1]} \right)^T \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c-1]} \underline{\mathbf{x}}_t^{(i,[c])} \right] - \text{tr} \left[ \left( \mathbf{L}_t^{[c]} \right)^T \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c]} \underline{\mathbf{x}}_t^{(i,[c])} \right] \\ & - 2 \text{tr} \left[ \left( \mathcal{E}^{(i)} \left( \mathcal{P}_{t+1}^{[c]} \right) \mathbf{K}_t^{[c]} \mathbf{D}_{t,1}^{(i,[c])} \right)^T \mathbf{K}_t^{[c-1]} \right] + 2 \text{tr} \left[ \left( \mathcal{E}^{(i)} \left( \mathcal{P}_{t+1}^{[c]} \right) \mathbf{K}_t^{[c]} \mathbf{D}_{t,1}^{(i,[c])} \right)^T \mathbf{K}_t^{[c]} \right] \\ & - 2 \text{tr} \left[ \left( \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c]} \underline{\mathbf{x}}_t^{(i,[c])} \right)^T \mathbf{L}_t^{[c-1]} \right] + 2 \text{tr} \left[ \left( \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c]} \underline{\mathbf{x}}_t^{(i,[c])} \right)^T \mathbf{L}_t^{[c]} \right]. \end{aligned}$$

Rearranging using the symmetry of  $\mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}^{[c]})$ ,  $\mathbf{D}_{t,1}^{(i,[c])}$ ,  $\mathbf{D}_{t,2}^{(i,[c])}$ , and  $\underline{\mathbf{X}}_t^{(i,[c])}$  results in

$$\begin{aligned} \Delta_t^{[c]} &= \sum_{i=0}^N \text{tr} \left[ \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}^{[c]}) \mathbf{K}_t^{[c-1]} \mathbf{D}_{t,1}^{(i,[c])} \left( \mathbf{K}_t^{[c-1]} \right)^{\text{T}} \right] - \left[ \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}^{[c]}) \mathbf{K}_t^{[c]} \mathbf{D}_{t,1}^{(i,[c])} \left( \mathbf{K}_t^{[c]} \right)^{\text{T}} \right] \\ &\quad - 2 \text{tr} \left[ \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}^{[c]}) \mathbf{K}_t^{[c-1]} \mathbf{D}_{t,1}^{(i,[c])} \left( \mathbf{K}_t^{[c]} \right)^{\text{T}} \right] + 2 \text{tr} \left[ \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}^{[c]}) \mathbf{K}_t^{[c]} \mathbf{D}_{t,1}^{(i,[c])} \left( \mathbf{K}_t^{[c]} \right)^{\text{T}} \right] \\ &\quad + \text{tr} \left[ \underline{\mathbf{X}}_t^{(i,[c])} \left( \mathbf{L}_t^{[c-1]} \right)^{\text{T}} \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c-1]} \right] - \text{tr} \left[ \underline{\mathbf{X}}_t^{(i,[c])} \left( \mathbf{L}_t^{[c]} \right)^{\text{T}} \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c]} \right] \\ &\quad - 2 \text{tr} \left[ \underline{\mathbf{X}}_t^{(i,[c])} \left( \mathbf{L}_t^{[c]} \right)^{\text{T}} \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c-1]} \right] + 2 \text{tr} \left[ \underline{\mathbf{X}}_t^{(i,[c])} \left( \mathbf{L}_t^{[c]} \right)^{\text{T}} \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c]} \right], \end{aligned}$$

which gives

$$\begin{aligned} \Delta_t^{[c]} &= \sum_{i=0}^N \text{tr} \left[ \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}^{[c]}) \mathbf{K}_t^{[c-1]} \mathbf{D}_{t,1}^{(i,[c])} \left( \mathbf{K}_t^{[c-1]} \right)^{\text{T}} \right] + \left[ \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}^{[c]}) \mathbf{K}_t^{[c]} \mathbf{D}_{t,1}^{(i,[c])} \left( \mathbf{K}_t^{[c]} \right)^{\text{T}} \right] \\ &\quad - 2 \text{tr} \left[ \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}^{[c]}) \mathbf{K}_t^{[c-1]} \mathbf{D}_{t,1}^{(i,[c])} \left( \mathbf{K}_t^{[c]} \right)^{\text{T}} \right] \\ &\quad + \text{tr} \left[ \underline{\mathbf{X}}_t^{(i,[c])} \left( \mathbf{L}_t^{[c-1]} \right)^{\text{T}} \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c-1]} \right] + \text{tr} \left[ \underline{\mathbf{X}}_t^{(i,[c])} \left( \mathbf{L}_t^{[c]} \right)^{\text{T}} \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c]} \right] \\ &\quad - 2 \text{tr} \left[ \underline{\mathbf{X}}_t^{(i,[c])} \left( \mathbf{L}_t^{[c]} \right)^{\text{T}} \mathbf{D}_{t,2}^{(i,[c])} \mathbf{L}_t^{[c-1]} \right]. \end{aligned}$$

This in turn yields the quadratic expression

$$\begin{aligned} \Delta_t^{[c]} &= \sum_{i=0}^N \text{tr} \left[ \mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}^{[c]}) \left( \mathbf{K}_t^{[c]} - \mathbf{K}_t^{[c-1]} \right) \mathbf{D}_{t,1}^{(i,[c])} \left( \mathbf{K}_t^{[c]} - \mathbf{K}_t^{[c-1]} \right)^{\text{T}} \right] \\ &\quad + \text{tr} \left[ \underline{\mathbf{X}}_t^{(i,[c])} \left( \mathbf{L}_t^{[c]} - \mathbf{L}_t^{[c-1]} \right)^{\text{T}} \mathbf{D}_{t,2}^{(i,[c])} \left( \mathbf{L}_t^{[c]} - \mathbf{L}_t^{[c-1]} \right) \right]. \end{aligned}$$

We have  $\mathcal{E}^{(i)}(\underline{\mathcal{P}}_{t+1}^{[c]})$ ,  $\underline{\mathbf{X}}_t^{(i,[c])} \succcurlyeq 0$  and also  $\mathbf{D}_{t,1}^{(i,[c])}$ ,  $\mathbf{D}_{t,2}^{(i,[c])} \succcurlyeq 0$ . Consequently, we get that

$$\begin{aligned} \left( \mathbf{K}_t^{[c]} - \mathbf{K}_t^{[c-1]} \right) \mathbf{D}_{t,1}^{(i,[c])} \left( \mathbf{K}_t^{[c]} - \mathbf{K}_t^{[c-1]} \right)^{\text{T}} &\succcurlyeq 0, \\ \left( \mathbf{L}_t^{[c]} - \mathbf{L}_t^{[c-1]} \right)^{\text{T}} \mathbf{D}_{t,2}^{(i,[c])} \left( \mathbf{L}_t^{[c]} - \mathbf{L}_t^{[c-1]} \right) &\succcurlyeq 0. \end{aligned}$$

Since the trace of the product of two positive semidefinite matrices is always nonnegative (cf. Theorem A.4 in Appendix A.4), we can conclude that  $\Delta_t^{[c]} \geq 0$ .

Hence, in each iteration, the update of the control law parameters  $(\mathbf{K}_t^{[c]}, \mathbf{L}_t^{[c]})$  during the backward pass leads to an improvement of the cost-to-go at every stage  $t = 0, 1, \dots, K-1$ . Consequently, for the cost we have  $\bar{\mathcal{V}}_0^{[c]} \leq \bar{\mathcal{V}}_0^{[c-1]}$ . The cost function is nonnegative by construction and thus bounded, implying convergence of the sequence  $\bar{\mathcal{V}}_0^{[1]}, \bar{\mathcal{V}}_0^{[2]}, \dots$  and, consequently, of the control law parameters  $(\mathbf{K}_0^{[c]}, \mathbf{L}_0^{[c]}), \dots, (\mathbf{K}_{K-1}^{[c]}, \mathbf{L}_{K-1}^{[c]})$ .  $\square$

## Proofs of the Results in Chapter 6

In the following sections, we give the proofs of Lemmas 6.1 and 6.2 and of Theorems 6.5, 6.6, 6.7, and 6.8, each of which is restated here for convenience. The proofs are presented in the order of appearance of the results.

### **D.1** Proof of Lemma 6.1

**Lemma 6.1:**

Let  $\mathcal{M}$  be a finite set of  $n$ -by- $n$  matrices and suppose  $\hat{\rho}(\mathcal{M}) < 1$ . Then the following propositions are true:

- (i) There exist  $\xi \geq 1$  and  $\beta \in (0, 1)$  such that for all  $\mathbf{P}_k \in \Pi_k(\text{conv}(\mathcal{M}))$  it holds  $\|\mathbf{P}_k\| \leq \xi\beta^k$  for all  $k \in \mathbb{N}$ .
- (ii) For any  $\mathbf{P}_k \in \Pi_k(\text{conv}(\mathcal{M}))$ ,  $\|\mathbf{P}_k\| \rightarrow 0$  as  $k \rightarrow \infty$ .
- (iii) The series  $\sum_{k=0}^{\infty} \|\mathbf{P}_k\|$  is convergent for all possible  $\mathbf{P}_k \in \Pi_k(\text{conv}(\mathcal{M}))$ .

*Proof.* In [244] it was proved that  $\hat{\rho}(\mathcal{M}) < 1$  implies proposition (i). It is trivially verified that (i)  $\Rightarrow$  (ii). To verify that (i) implies (iii), it is enough show that the sequence of partial sums  $\{s_m\}$  with  $s_m = \sum_{k=0}^m \|\mathbf{P}_k\|$  is a Cauchy sequence according to Definition A.6. To show this, we note that for  $m \geq l$  we have  $s_m - s_l = \sum_{k=l+1}^m \|\mathbf{P}_k\|$ , and thus

$$\|s_m - s_l\| = \sum_{k=l+1}^m \|\mathbf{P}_k\| \leq \xi \sum_{k=l+1}^m \beta^k = \xi \frac{\beta^{m+1} - \beta^{l+1}}{\beta - 1}.$$

The fraction on the right approaches zero as  $m$  and  $l$  increase. Hence, for any  $\epsilon > 0$ , we can always find an integer  $N$  such that  $\|s_m - s_l\| \leq \xi \frac{\beta^{m+1} - \beta^{l+1}}{\beta - 1} < \epsilon$  for any  $m, l \geq N$  and the claim (iii) follows. □

## D.2 Proof of Lemma 6.2

### Lemma 6.2:

Let  $\underline{z}_k = \hat{\varphi}(\underline{\mathcal{Z}}_k)$  with  $\underline{\mathcal{Z}}_k$  and  $\hat{\varphi}(\cdot)$  according to (6.18) and (6.20). Then, the dynamics of  $\underline{z}_k$  is linear and given by

$$\underline{z}_{k+1} = \mathbf{A}'_k \underline{z}_k + \mathbf{G}'_k \underline{w}'_k, \quad \underline{z}_0 = \hat{\varphi}(\underline{\mathcal{Z}}_0), \quad (6.23)$$

with  $\underline{w}'_k$  as per (6.21),  $\mathbf{A}'_k, \mathbf{G}'_k \in \mathbb{R}^{(N+1)n_\psi^2 \times (N+1)n_\psi^2}$  according to

$$\mathbf{A}'_k = \sum_{r=1}^R \alpha_k^{(r)} \left( \left( \boldsymbol{\Lambda}^{(r)} \right)^T \otimes \mathbf{I}_{n_\psi^2} \right) \underline{\mathbf{A}}, \quad (6.24)$$

$$\mathbf{G}'_k = \sum_{r=1}^R \alpha_k^{(r)} \left( \left( \boldsymbol{\Lambda}^{(r)} \right)^T \otimes \mathbf{I}_{n_\psi^2} \right), \quad (6.25)$$

and where

$$\sum_{r=1}^R \alpha_k^{(r)} = 1, \quad \alpha_k^{(1)}, \alpha_k^{(2)}, \dots, \alpha_k^{(R)} \geq 0.$$

*Proof.* With arguments similar those used in the proof of Lemma 5.2 (cf. Appendix C.2), the evaluation of the expectation in (6.16) reveals that the dynamics of  $\boldsymbol{\Psi}_k^{(i)}$  is given by

$$\boldsymbol{\Psi}_{k+1}^{(j)} = \sum_{i=0}^N t_{k,ij} \left( \tilde{\mathbf{A}}^{(i)} \boldsymbol{\Psi}_k^{(i)} \left( \tilde{\mathbf{A}}^{(i)} \right)^T + \tilde{\mathbf{W}}_k^{(i)} \right),$$

for  $j = 0, 1, \dots, N$ , and hence by (6.15)

$$\boldsymbol{\Psi}_{k+1}^{(j)} = \sum_{i=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)} \left( \tilde{\mathbf{A}}^{(i)} \boldsymbol{\Psi}_k^{(i)} \left( \tilde{\mathbf{A}}^{(i)} \right)^T + \tilde{\mathbf{W}}_k^{(i)} \right). \quad (\text{D.1})$$

With the aid of (A.11), the corresponding vectorized dynamics is

$$\begin{aligned} \text{vec} \left( \boldsymbol{\Psi}_{k+1}^{(j)} \right) &= \sum_{i=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)} \left[ \text{vec} \left( \tilde{\mathbf{A}}^{(i)} \boldsymbol{\Psi}_k^{(i)} \left( \tilde{\mathbf{A}}^{(i)} \right)^T \right) + \text{vec} \left( \tilde{\mathbf{W}}_k^{(i)} \right) \right] \\ &= \sum_{i=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)} \left[ \left( \tilde{\mathbf{A}}^{(i)} \otimes \tilde{\mathbf{A}}^{(i)} \right) \text{vec} \left( \boldsymbol{\Psi}_k^{(i)} \right) + \text{vec} \left( \tilde{\mathbf{W}}_k^{(i)} \right) \right], \end{aligned}$$

resulting in a set of  $N + 1$  similar equations

$$\begin{aligned} \text{vec} \left( \boldsymbol{\Psi}_{k+1}^{(0)} \right) &= \sum_{i=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{i0}^{(r)} \left[ \left( \tilde{\mathbf{A}}^{(i)} \otimes \tilde{\mathbf{A}}^{(i)} \right) \text{vec} \left( \boldsymbol{\Psi}_k^{(i)} \right) + \text{vec} \left( \tilde{\mathbf{W}}_k^{(i)} \right) \right], \\ \text{vec} \left( \boldsymbol{\Psi}_{k+1}^{(1)} \right) &= \sum_{i=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{i1}^{(r)} \left[ \left( \tilde{\mathbf{A}}^{(i)} \otimes \tilde{\mathbf{A}}^{(i)} \right) \text{vec} \left( \boldsymbol{\Psi}_k^{(i)} \right) + \text{vec} \left( \tilde{\mathbf{W}}_k^{(i)} \right) \right], \\ &\vdots \\ \text{vec} \left( \boldsymbol{\Psi}_{k+1}^{(N)} \right) &= \sum_{i=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{iN}^{(r)} \left[ \left( \tilde{\mathbf{A}}^{(i)} \otimes \tilde{\mathbf{A}}^{(i)} \right) \text{vec} \left( \boldsymbol{\Psi}_k^{(i)} \right) + \text{vec} \left( \tilde{\mathbf{W}}_k^{(i)} \right) \right], \end{aligned}$$

that are of the form (A.6). Rewriting these equations as indicated by (A.7) then yields (6.23) using the definitions of  $\underline{z}_k$ ,  $\underline{w}'_k$ ,  $\mathbf{A}'_k$ ,  $\mathbf{G}'_k$ , and  $\underline{\mathbf{A}}$ .  $\square$

### D.3 Proof of Theorem 6.5

#### Theorem 6.5:

Consider the dynamics (6.23) with  $\underline{z}_k = \hat{\varphi}(\underline{\mathbf{Z}}_k)$  and  $\underline{\mathbf{Z}}_k$  and  $\hat{\varphi}(\cdot)$  according to (6.18) and (6.20), and let  $\mathcal{A}_R$  be given by (6.26). Then  $\lim_{k \rightarrow \infty} \underline{z}_k = \underline{z}$  for some  $\underline{z} \in \mathbb{R}^{(N+1)n_\psi^2}$  that is independent of  $\underline{z}_0$  if and only if  $\hat{\rho}(\mathcal{A}_R) < 1$ .

*Proof.* The dynamics (6.23) of  $\underline{z}_k$  is linear. Thus, repeated application gives

$$\underline{z}_k = \mathbf{A}'_{k-1} \mathbf{A}'_{k-2} \cdots \mathbf{A}'_0 \underline{z}_0 + \sum_{j=0}^{k-1} \mathbf{A}'_{k-1} \mathbf{A}'_{k-2} \cdots \mathbf{A}'_{j+1} \mathbf{G}'_j \underline{w}'_j, \quad (\text{D.2})$$

for any initial condition  $\underline{z}_0$ .

To show necessity, suppose that for some  $\underline{z} \in \mathbb{R}^{(N+1)n_\psi^2}$

$$\lim_{k \rightarrow \infty} \underline{z}_k = \underline{z}, \quad (\text{D.3})$$

holds for any initial condition  $\underline{z}_0$ . For the particular initial condition  $\underline{z}_0 = \underline{0}$ , the first term on the right side of (D.2) vanishes for any  $k$ , implying that the sum on the right must converge to the limit given in (D.3). This term is, however, independent of  $\underline{z}_0$ , we thus conclude that the first term on the right side of (D.2) must vanish for any initial condition. Hence, the product  $\mathbf{A}'_{k-1} \mathbf{A}'_{k-2} \cdots \mathbf{A}'_0 \in \Pi_k(\text{conv}(\mathcal{A}_R))$ , with  $\Pi_k(\cdot)$  given by (6.5), must converge to  $\mathbf{0}$ . The application of Theorems 6.1 and 6.2 then yields that  $\hat{\rho}(\mathcal{A}_R) < 1$  must hold.

For sufficiency, suppose now that  $\hat{\rho}(\mathcal{A}_R) < 1$ . Then, Theorems 6.1 and 6.2 allow us to conclude that the first term on the right of (D.2) must vanish as  $k \rightarrow \infty$ . Hence, we must show that the sum on the right converges as  $k \rightarrow \infty$ . To that end, we first note that the definition (6.25) of  $\mathbf{G}'_k$  implies

$$\begin{aligned} \|\mathbf{G}'_k\|_{\text{F}} &= \left\| \sum_{r=1}^R \alpha_k^{(r)} \left( (\mathbf{\Lambda}^{(r)})^{\text{T}} \otimes \mathbf{I}_{n_\psi^2} \right) \right\|_{\text{F}} \\ &\leq \sum_{r=1}^R \alpha_k^{(r)} \left\| \left( (\mathbf{\Lambda}^{(r)})^{\text{T}} \otimes \mathbf{I}_{n_\psi^2} \right) \right\|_{\text{F}} \\ &\leq \sum_{r=1}^R \left\| \left( (\mathbf{\Lambda}^{(r)})^{\text{T}} \otimes \mathbf{I}_{n_\psi^2} \right) \right\|_{\text{F}} = c, \end{aligned}$$

for some  $c > 0$  since  $\alpha_k^{(r)} \leq 1$ . Similarly, we get from (6.21)

$$\begin{aligned} \|\underline{w}'_k\|_2 = \hat{\varphi}(\tilde{\mathbf{W}}_k) &= \left\| \begin{bmatrix} \text{vec}(\tilde{\mathbf{W}}_k^{(0)}) \\ \text{vec}(\tilde{\mathbf{W}}_k^{(1)}) \\ \vdots \\ \text{vec}(\tilde{\mathbf{W}}_k^{(N)}) \end{bmatrix} \right\|_2 = \sqrt{\sum_{i=0}^N \|\text{vec}(\tilde{\mathbf{W}}_k^{(i)})\|_2^2} = \sqrt{\sum_{i=0}^N \|\tilde{\mathbf{W}}_k^{(i)}\|_{\text{F}}^2} \\ &= \sqrt{\sum_{i=0}^N \left\| \mathbb{E}_{\tilde{w}_k, \theta_k} \{ \tilde{w}_k \tilde{w}_k^{\text{T}} \mathbf{1}_{\{\theta_k=i\}} \} \right\|_{\text{F}}^2} = \sqrt{\sum_{i=0}^N \mathbb{E}_{\theta_k} \{ \mathbf{1}_{\{\theta_k=i\}} \}^2} \|\tilde{\mathbf{W}}\|_{\text{F}}^2 \\ &\leq \sqrt{\sum_{i=0}^N \|\tilde{\mathbf{W}}\|_{\text{F}}^2} = \sqrt{N+1} \|\tilde{\mathbf{W}}\|_{\text{F}}, \end{aligned}$$

since  $\mathbb{E}\{\mathbb{1}_{\{\theta_k=i\}}\} = \mathbb{P}[\theta_k = i] \leq 1$ . Using that the Frobenius norm is submultiplicative and compatible with the Euclidean norm (cf. (A.1) and (A.3)), we then obtain

$$\begin{aligned} \sum_{j=0}^{k-1} \|\mathbf{A}'_{k-1} \mathbf{A}'_{k-2} \cdots \mathbf{A}'_{j+1} \mathbf{G}'_j \underline{w}'_j\|_2 &\leq \sum_{j=0}^{k-1} \|\mathbf{A}'_{k-1} \mathbf{A}'_{k-2} \cdots \mathbf{A}'_{j+1} \mathbf{G}'_j\|_F \|\underline{w}'_j\|_2 \\ &\leq \sum_{j=0}^{k-1} \|\mathbf{A}'_{k-1} \mathbf{A}'_{k-2} \cdots \mathbf{A}'_{j+1}\|_F \|\mathbf{G}'_j\|_F \|\underline{w}'_j\|_2 \\ &\leq c\sqrt{N+1} \|\tilde{\mathbf{W}}\|_F \sum_{j=0}^{k-1} \|\mathbf{A}'_{k-1} \mathbf{A}'_{k-2} \cdots \mathbf{A}'_{j+1}\|_F. \end{aligned}$$

The convergence of the sum on the right side is ensured by item (iii) of Lemma 6.1, which in turn establishes the convergence of the sum in (D.2).  $\square$

## D.4 Proof of Theorem 6.6

### Theorem 6.6:

Consider the polytopic MJLS (6.14) with first moment  $\hat{\underline{\psi}}_k = \mathbb{E}\{\underline{\psi}_k\}$  and let  $\mathcal{A}_R$  be given by (6.26). Then (6.12) holds with  $\underline{\psi} = \underline{0}$  whenever  $\hat{\rho}(\mathcal{A}_R) < 1$  holds, i.e., the first moment  $\hat{\underline{\psi}}_k$  converges to  $\underline{0}$  if the JSR of  $\mathcal{A}_R$  is less than one.

*Proof.* We need to show that

$$\hat{\rho}(\mathcal{A}_R) < 1 \Rightarrow \lim_{k \rightarrow \infty} \hat{\underline{\psi}}_k = \underline{0}, \quad (\text{D.4})$$

holds for any  $\underline{\psi}_0$  and  $\theta_0$ . To prove that this is indeed the case, we begin with the decomposition of the first moment  $\hat{\underline{\psi}}_k$  into  $N+1$  vectors  $\hat{\underline{\psi}}_k^{(i)}$  according to

$$\hat{\underline{\psi}}_k^{(i)} = \mathbb{E}_{\underline{\psi}_k, \theta_k} \left\{ \underline{\psi}_k \mathbb{1}_{\{\theta_k=i\}} \right\},$$

for  $i = 0, 1, \dots, N$ , so that it holds [127, p. 31]

$$\hat{\underline{\psi}}_k = \sum_{i=0}^N \hat{\underline{\psi}}_k^{(i)}. \quad (\text{D.5})$$

Computations similar to those used in the proof of Lemma 5.2 (cf. Appendix C.2) reveal that the dynamics of  $\hat{\underline{\psi}}_k^{(i)}$  is given by

$$\hat{\underline{\psi}}_{k+1}^{(j)} = \sum_{i=0}^N t_{k,ij} \tilde{\mathbf{A}}^{(i)} \hat{\underline{\psi}}_k^{(i)},$$

for  $j = 0, 1, \dots, N$ , yielding

$$\hat{\underline{\psi}}_{k+1}^{(j)} = \sum_{i=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)} \tilde{\mathbf{A}}^{(i)} \hat{\underline{\psi}}_k^{(i)},$$

when we use (6.15). The resulting  $N + 1$  equations

$$\begin{aligned}\underline{\hat{\psi}}_{k+1}^{(0)} &= \sum_{r=1}^R \alpha_k^{(r)} \left( \lambda_{00}^{(r)} \tilde{\mathbf{A}}^{(0)} \underline{\hat{\psi}}_k^{(0)} + \lambda_{10}^{(r)} \tilde{\mathbf{A}}^{(1)} \underline{\hat{\psi}}_k^{(1)} + \cdots + \lambda_{N0}^{(r)} \tilde{\mathbf{A}}^{(N)} \underline{\hat{\psi}}_k^{(N)} \right), \\ \underline{\hat{\psi}}_{k+1}^{(1)} &= \sum_{r=1}^R \alpha_k^{(r)} \left( \lambda_{01}^{(r)} \tilde{\mathbf{A}}^{(0)} \underline{\hat{\psi}}_k^{(0)} + \lambda_{11}^{(r)} \tilde{\mathbf{A}}^{(1)} \underline{\hat{\psi}}_k^{(1)} + \cdots + \lambda_{N1}^{(r)} \tilde{\mathbf{A}}^{(N)} \underline{\hat{\psi}}_k^{(N)} \right), \\ &\vdots \\ \underline{\hat{\psi}}_{k+1}^{(N)} &= \sum_{r=1}^R \alpha_k^{(r)} \left( \lambda_{0N}^{(r)} \tilde{\mathbf{A}}^{(0)} \underline{\hat{\psi}}_k^{(0)} + \lambda_{1N}^{(r)} \tilde{\mathbf{A}}^{(1)} \underline{\hat{\psi}}_k^{(1)} + \cdots + \lambda_{NN}^{(r)} \tilde{\mathbf{A}}^{(N)} \underline{\hat{\psi}}_k^{(N)} \right),\end{aligned}$$

are of the form (A.6) and can thus be written compactly as

$$\underline{s}_{k+1} = \mathbf{M}'_k \underline{s}_k, \quad (\text{D.6})$$

with

$$\underline{s}_k = \begin{bmatrix} \underline{\hat{\psi}}_k^{(0)} \\ \underline{\hat{\psi}}_k^{(1)} \\ \vdots \\ \underline{\hat{\psi}}_k^{(N)} \end{bmatrix} \in \mathbb{R}^{(N+1)n_\psi}, \quad \mathbf{M}'_k = \sum_{r=1}^R \alpha_k^{(r)} \left( \left( \boldsymbol{\Lambda}^{(r)} \right)^T \otimes \mathbf{I}_{n_\psi} \right) \tilde{\mathbf{A}} \in \mathbb{R}^{(N+1)n_\psi \times (N+1)n_\psi},$$

and where

$$\tilde{\mathbf{A}} = \begin{bmatrix} \tilde{\mathbf{A}}^{(0)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{A}}^{(1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \tilde{\mathbf{A}}^{(N)} \end{bmatrix} \in \mathbb{R}^{(N+1)n_\psi \times (N+1)n_\psi}.$$

In view of (D.6), denote by  $\mathcal{B}_R$  the set

$$\mathcal{B}_R = \left\{ \left( \left( \boldsymbol{\Lambda}^{(1)} \right)^T \otimes \mathbf{I}_{n_\psi} \right) \tilde{\mathbf{A}}, \left( \left( \boldsymbol{\Lambda}^{(2)} \right)^T \otimes \mathbf{I}_{n_\psi} \right) \tilde{\mathbf{A}}, \dots, \left( \left( \boldsymbol{\Lambda}^{(R)} \right)^T \otimes \mathbf{I}_{n_\psi} \right) \tilde{\mathbf{A}} \right\}.$$

Hence, we can write  $\mathbf{M}'_k \in \text{conv}(\mathcal{B}_R)$  for all  $k$ .

Now suppose that  $\hat{\rho}(\mathcal{B}_R) < 1$  holds. For any initial condition  $\underline{s}_0$ , the repeated application of (D.6) gives

$$\underline{s}_k = \mathbf{M}'_{k-1} \mathbf{M}'_{k-2} \cdots \mathbf{M}'_0 \underline{s}_0. \quad (\text{D.7})$$

Then, Theorems 6.1 and 6.2 allow us to conclude that the matrix product on the right of (D.7) must vanish as  $k \rightarrow \infty$ . Consequently, it holds  $\lim_{k \rightarrow \infty} \underline{s}_k = \underline{0}$ , and, in particular,  $\lim_{k \rightarrow \infty} \underline{\hat{\psi}}_k^{(i)} = \underline{0}$  for  $i = 0, 1, \dots, N$ . Convergence of the first moment  $\underline{\hat{\psi}}_k$  then follows from (D.5), that is, the implication

$$\hat{\rho}(\mathcal{B}_R) < 1 \Rightarrow \lim_{k \rightarrow \infty} \underline{\hat{\psi}}_k = \underline{0},$$

is true for any  $\underline{\psi}_0$  and  $\theta_0$ .

To verify (D.4), it remains to show that  $\hat{\rho}(\mathcal{A}_R) < 1 \Rightarrow \hat{\rho}(\mathcal{B}_R) < 1$ . To that end, assume that  $\hat{\rho}(\mathcal{A}_R) < 1$  and consider the dynamics (6.14) with  $\underline{\tilde{w}}_k \equiv \underline{0}$

$$\begin{aligned}\underline{\psi}_{k+1} &= \tilde{\mathbf{A}}^{(\theta_k)} \underline{\psi}_k, \\ \mathbf{T}_k &\in \text{conv}(\mathcal{L}).\end{aligned}$$

Accordingly, the dynamics of  $\underline{z}_k = \hat{\varphi}(\mathcal{Z}_k)$  becomes

$$\underline{z}_{k+1} = \mathbf{A}'_k \underline{z}_k,$$

as per to Lemma 6.2 with  $\mathbf{A}'_k$  given by (6.24), so that for any initial condition  $\underline{z}_0$  we can write

$$\underline{z}_k = \mathbf{A}'_{k-1} \mathbf{A}'_{k-2} \cdots \mathbf{A}'_0 \underline{z}_0.$$

Theorems 6.1 and 6.2 imply that  $\lim_{k \rightarrow \infty} \mathbf{A}'_{k-1} \mathbf{A}'_{k-2} \cdots \mathbf{A}'_0 = \mathbf{0}$ . Thus,  $\lim_{k \rightarrow \infty} \underline{z}_k = \underline{0}$  for any  $\underline{z}_0$ , and, consequently

$$\begin{aligned} \lim_{k \rightarrow \infty} \mathcal{Z}_k &= (\mathbf{0}, \mathbf{0}, \dots, \mathbf{0}), \\ \lim_{k \rightarrow \infty} \|\mathcal{Z}_k\|_{\mathbb{H}} &= 0, \end{aligned}$$

by Theorem A.1, where  $\|\cdot\|_{\mathbb{H}}$  denotes the norm on  $\mathbb{H}^{n_\psi}$  as defined by (6.19).

On the other hand, Jensen's inequality states that for any random variable  $\underline{a}$  we have that  $\|\mathbb{E}\{\underline{a}\}\|^2 \leq \mathbb{E}\{\|\underline{a}\|^2\}$  [256, p. 77]. Hence, using (D.7) and the definitions of  $\underline{s}_k$  and  $\hat{\psi}_k^{(i)}$ , we get

$$\begin{aligned} \|\mathbf{M}'_{k-1} \mathbf{M}'_{k-2} \cdots \mathbf{M}'_0 \underline{s}_0\|_2^2 &= \|\underline{s}_k\|_2^2 = \sum_{i=0}^N \|\hat{\psi}_k^{(i)}\|_2^2 = \sum_{i=0}^N \left\| \mathbb{E}\left\{ \underline{\psi}_k \mathbf{1}_{\{\theta_k=i\}} \right\} \right\|_2^2 \\ &\leq \sum_{i=0}^N \mathbb{E}\left\{ \left\| \underline{\psi}_k \mathbf{1}_{\{\theta_k=i\}} \right\|_2^2 \right\} = \sum_{i=0}^N \mathbb{E}\left\{ \underline{\psi}_k^T \underline{\psi}_k \mathbf{1}_{\{\theta_k=i\}} \right\} \\ &= \sum_{i=0}^N \text{tr} \left[ \mathbb{E}\left\{ \underline{\psi}_k \underline{\psi}_k^T \mathbf{1}_{\{\theta_k=i\}} \right\} \right] = \sum_{i=0}^N \text{tr} \left[ \Psi_k^{(i)} \right] \\ &\leq n_\psi \sum_{i=0}^N \|\Psi_k^{(i)}\| = n_\psi \|\mathcal{Z}_k\|_{\mathbb{H}} \xrightarrow{k \rightarrow \infty} 0, \end{aligned}$$

where the second inequality is due to Lemma A.1. Thus,  $\lim_{k \rightarrow \infty} \|\mathbf{M}'_{k-1} \mathbf{M}'_{k-2} \cdots \mathbf{M}'_0 \underline{s}_0\|_2^2 = 0$  for any  $\underline{s}_0$ , implying that  $\lim_{k \rightarrow \infty} \mathbf{M}'_{k-1} \mathbf{M}'_{k-2} \cdots \mathbf{M}'_0 = \mathbf{0}$ . Theorems 6.1 and 6.2 then confirm that  $\hat{\rho}(\mathcal{B}_R) < 1$  holds.  $\square$

## D.5 Proof of Theorem 6.7

### Theorem 6.7:

If there exist positive definite matrices  $\tilde{\mathbf{D}}^{(0)}, \tilde{\mathbf{D}}^{(1)}, \dots, \tilde{\mathbf{D}}^{(N)} \in \mathbb{R}^{n_\psi \times n_\psi}$  such that it holds

$$\tilde{\mathbf{D}}^{(i)} - \sum_{j=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)} \left( \tilde{\mathbf{A}}^{(i)} \right)^T \tilde{\mathbf{D}}^{(j)} \tilde{\mathbf{A}}^{(i)} \succ 0, \quad (6.27)$$

for  $i = 0, 1, \dots, N$  and any set of coefficients  $\alpha_k^{(1)}, \alpha_k^{(2)}, \dots, \alpha_k^{(R)} \geq 0$  satisfying  $\sum_{r=1}^R \alpha_k^{(r)} = 1$ , then the polytopic MJLS (6.14) is MSS.

*Proof.* For notational convenience, let us first denote by  $\mathbb{H}_+^{n_\psi}$  the restriction of  $\mathbb{H}^{n_\psi}$  to tuples of positive semidefinite matrices, i.e.,

$$\mathbb{H}_+^{n_\psi} \triangleq \left\{ \left( \mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N)} \right) \mid \mathbf{X}^{(i)} \in \mathbb{R}^{n_\psi \times n_\psi}, \mathbf{X}^{(i)} \succcurlyeq 0 \right\}.$$

We proceed similar to the proof of Proposition 7 in [258] and consider the dynamics (6.14) with  $\tilde{\underline{w}}_k \equiv \underline{0}$

$$\begin{aligned}\underline{\psi}_{k+1} &= \tilde{\mathbf{A}}^{(\theta_k)} \underline{\psi}_k, \\ \mathbf{T}_k &\in \text{conv}(\mathcal{L}).\end{aligned}$$

According to (D.1), it then holds for the evolution of the second moment

$$\Psi_{k+1}^{(j)} = \sum_{i=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)} \left( \tilde{\mathbf{A}}^{(i)} \Psi_k^{(i)} \left( \tilde{\mathbf{A}}^{(i)} \right)^{\text{T}} \right), \quad (\text{D.8})$$

for  $j = 0, 1, \dots, N$ , where  $\lambda_{ij}^{(r)}$  are the entries of  $\Lambda^{(r)}$ . For any initial condition  $\underline{\psi}_0, \theta_0$ , we have

$$\Psi_0^{(i)} = \mathbb{E} \left\{ \underline{\psi}_0 \underline{\psi}_0^{\text{T}} \mathbf{1}_{\{\theta_0=i\}} \right\} \succcurlyeq 0,$$

for  $i = 0, 1, \dots, N$ . Hence, we have

$$\mathcal{Z}_k = \left( \Psi_k^{(0)}, \Psi_k^{(1)}, \dots, \Psi_k^{(N)} \right) \in \mathbb{H}_+^{n_\psi},$$

for all  $k$  and the only equilibrium of (D.8) is

$$\mathcal{Z}_e = (\mathbf{0}, \mathbf{0}, \dots, \mathbf{0}) \in \mathbb{H}_+^{n_\psi}.$$

For any  $\mathcal{Y} = (\mathbf{Y}^{(0)}, \mathbf{Y}^{(1)}, \dots, \mathbf{Y}^{(N)}) \in \mathbb{H}_+^{n_\psi}$  consider the function  $V: \mathbb{H}_+^{n_\psi} \mapsto \mathbb{R}$  defined as

$$V(\mathcal{Y}) = \sum_{i=0}^N \text{tr}[\tilde{\mathbf{D}}^{(i)} \mathbf{Y}^{(i)}], \quad (\text{D.9})$$

We claim that (D.9) is a radially unbounded Lyapunov function for the dynamics (D.8), rendering its equilibrium  $\mathcal{Z}_e$  globally asymptotically stable. Consequently, the dynamics of  $\underline{z}_k = \hat{\varphi}(\mathcal{Z}_k)$  given by

$$\underline{z}_{k+1} = \mathbf{A}'_k \underline{z}_k, \quad \underline{z}_0 = \hat{\varphi}(\mathcal{Z}_0),$$

according to Lemma 6.2, with  $\mathbf{A}'_k$  as per (6.24), must possess a globally asymptotically stable equilibrium at the origin  $\underline{z}_e = \underline{0}$ . Hence, the product  $\mathbf{A}'_{k-1} \mathbf{A}'_{k-2} \cdot \dots \cdot \mathbf{A}'_0 \in \Pi_k(\text{conv}(\mathcal{A}_R))$ , with  $\Pi_k(\cdot)$  given by (6.5), must converge to  $\mathbf{0}$  for any initial condition  $\underline{z}_0$ . The application of Theorems 6.1 and 6.2 then yields that  $\hat{\rho}(\mathcal{A}_R) < 1$  must hold, from which we can conclude that (6.14) is MSS.

To establish that (D.9) is indeed a Lyapunov function and, moreover, radially unbounded, we need to show that [127, Theorem 2.13]

- (i)  $V$  is continuous,
- (ii)  $V(\mathcal{Z}_e) = 0$ ,
- (iii)  $V(\mathcal{Y}) > 0$  for all  $\mathcal{Y} \neq \mathcal{Z}_e$ ,
- (iv)  $V(\mathcal{Y}) \xrightarrow{\|\mathcal{Y}\|_{\mathbb{H}} \rightarrow \infty} \infty$ ,
- (v)  $\Delta V = V(\mathcal{Z}_{k+1}) - V(\mathcal{Z}_k) < 0$  for all  $\mathcal{Z}_k \neq \mathcal{Z}_e$ .

Continuity (i) is ensured by Theorem A.2 in Appendix A.2, since  $V$  is linear and  $\mathbb{H}_+^{n_\psi}$  is finite-dimensional. Application of Lemma A.2 shows that

$$\begin{aligned} V(\mathcal{Y}) &= \sum_{i=0}^N \text{tr}[\tilde{\mathbf{D}}^{(i)} \mathbf{Y}^{(i)}] \\ &\geq \sum_{i=0}^N \lambda_{\min}(\tilde{\mathbf{D}}^{(i)}) \text{tr}[\mathbf{Y}^{(i)}] \\ &\geq 0, \end{aligned} \tag{D.10}$$

with  $\lambda_{\min}(\tilde{\mathbf{D}}^{(i)}) > 0$  the minimal eigenvalue of  $\tilde{\mathbf{D}}^{(i)}$ . Since  $\text{tr}[\mathbf{Y}^{(i)}] = 0$  subject to  $\mathbf{Y}^{(i)} \succcurlyeq 0$  implies that  $\mathbf{Y}^{(i)} = \mathbf{0}$ , it follows that  $V(\mathcal{Y}) = 0 \Leftrightarrow \mathcal{Y} = \mathcal{Z}_e$ , verifying properties (ii) and (iii). Similarly, (D.10) implies (iv) because for  $\mathbf{Y}^{(i)} \succcurlyeq 0$  we have  $\text{tr}[\mathbf{Y}^{(i)}] \rightarrow \infty$  as  $\|\mathbf{Y}^{(i)}\| \rightarrow \infty$ . Finally, to establish (v) we evaluate the difference  $V(\mathcal{Z}_{k+1}) - V(\mathcal{Z}_k)$ , which yields using the dynamics (D.8)

$$\begin{aligned} \Delta V &= \sum_{j=0}^N \text{tr}[\tilde{\mathbf{D}}^{(j)} \Psi_{k+1}^{(j)}] - \sum_{j=0}^N \text{tr}[\tilde{\mathbf{D}}^{(j)} \Psi_k^{(j)}] \\ &= \sum_{j=0}^N \text{tr} \left[ \tilde{\mathbf{D}}^{(j)} \left( \sum_{i=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)} \left( \tilde{\mathbf{A}}^{(i)} \Psi_k^{(i)} \left( \tilde{\mathbf{A}}^{(i)} \right)^T \right) \right) \right] - \sum_{j=0}^N \text{tr}[\tilde{\mathbf{D}}^{(j)} \Psi_k^{(j)}] \\ &= \sum_{i=0}^N \text{tr} \left[ \sum_{j=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)} \left( \tilde{\mathbf{A}}^{(i)} \right)^T \tilde{\mathbf{D}}^{(j)} \tilde{\mathbf{A}}^{(i)} \Psi_k^{(i)} \right] - \sum_{i=0}^N \text{tr}[\tilde{\mathbf{D}}^{(i)} \Psi_k^{(i)}] \\ &= \sum_{i=0}^N \text{tr} \left[ \left( \sum_{j=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)} \left( \tilde{\mathbf{A}}^{(i)} \right)^T \tilde{\mathbf{D}}^{(j)} \tilde{\mathbf{A}}^{(i)} - \tilde{\mathbf{D}}^{(i)} \right) \Psi_k^{(i)} \right] \\ &= \sum_{i=0}^N \text{tr}[-\mathbf{S}^{(i)} \Psi_k^{(i)}], \end{aligned}$$

with  $\mathbf{S}^{(i)} \succ 0$  given by

$$\mathbf{S}^{(i)} = \tilde{\mathbf{D}}^{(i)} - \sum_{j=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)} \left( \tilde{\mathbf{A}}^{(i)} \right)^T \tilde{\mathbf{D}}^{(j)} \tilde{\mathbf{A}}^{(i)}.$$

Thus, by Lemma A.2

$$\begin{aligned} \Delta V &= - \sum_{i=0}^N \text{tr}[\mathbf{S}^{(i)} \Psi_k^{(i)}] \\ &\leq - \sum_{i=0}^N \lambda_{\min}(\mathbf{S}^{(i)}) \text{tr}[\Psi_k^{(i)}] \\ &\leq 0, \end{aligned}$$

with  $\lambda_{\min}(\mathbf{S}^{(i)}) > 0$  the minimal eigenvalue of  $\mathbf{S}^{(i)}$ . Since  $\text{tr}[\Psi_k^{(i)}] = 0$  subject to  $\Psi_k^{(i)} \succcurlyeq 0$  implies that  $\Psi_k^{(i)} = \mathbf{0}$ , it follows that  $\Delta V = 0 \Leftrightarrow \mathcal{Z}_k = \mathcal{Z}_e$ , and property (v) is established.  $\square$

## D.6 Proof of Theorem 6.8

### Theorem 6.8:

The following two assertions are equivalent:

- (i) There exist positive definite matrices  $\tilde{\mathbf{D}}^{(0)}, \tilde{\mathbf{D}}^{(1)}, \dots, \tilde{\mathbf{D}}^{(N)} \in \mathbb{R}^{n_\psi \times n_\psi}$  such that (6.27) is satisfied for  $i = 0, 1, \dots, N$ .
- (ii) There exist positive definite matrices  $\mathbf{D}^{(0)}, \mathbf{D}^{(1)}, \dots, \mathbf{D}^{(N)} \in \mathbb{R}^{n_\psi \times n_\psi}$  and positive definite matrices  $\mathbf{E}^{(0)}, \mathbf{E}^{(1)}, \dots, \mathbf{E}^{(N)} \in \mathbb{R}^{n_\psi \times n_\psi}$  satisfying

$$\begin{bmatrix} 2\mathbf{E}^{(i)} - \mathbf{D}^{(i)} & \mathbf{E}^{(i)} \left( \tilde{\mathbf{A}}^{(i)} \right)^\top \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right) \\ \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right)^\top \tilde{\mathbf{A}}^{(i)} \mathbf{E}^{(i)} & \bar{\mathbf{D}} \end{bmatrix} \succ 0, \quad (6.29)$$

for  $i = 0, 1, \dots, N$ , and  $r = 1, 2, \dots, R$ , where

$$\underline{\lambda}^{(i,r)} = \left[ \sqrt{\lambda_{i0}^{(r)}} \quad \sqrt{\lambda_{i1}^{(r)}} \quad \dots \quad \sqrt{\lambda_{iN}^{(r)}} \right] \in \mathbb{R}^{1 \times N+1}, \quad (6.30)$$

$$\bar{\mathbf{D}} = \begin{bmatrix} \mathbf{D}^{(0)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^{(1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{D}^{(N)} \end{bmatrix} \in \mathbb{R}^{(N+1)n_\psi \times (N+1)n_\psi}. \quad (6.31)$$

*Proof.* To prove the equivalence of assertions (i) and (ii), we use arguments similar to those employed in the proof of Proposition 2 in [245]. Assume first that (i) holds, i.e., we have

$$\tilde{\mathbf{D}}^{(i)} - \sum_{j=0}^N \sum_{r=1}^R \alpha_k^{(r)} \lambda_{ij}^{(r)} \left( \tilde{\mathbf{A}}^{(i)} \right)^\top \tilde{\mathbf{D}}^{(j)} \tilde{\mathbf{A}}^{(i)} \succ 0, \quad (D.11)$$

for  $i = 0, 1, \dots, N$  and any coefficients  $\alpha_k^{(r)}$  that satisfy

$$\sum_{r=1}^R \alpha_k^{(r)} = 1, \quad \alpha_k^{(1)}, \alpha_k^{(2)}, \dots, \alpha_k^{(R)} \geq 0.$$

Set

$$\alpha_k^{(r)} = \begin{cases} 1 & r = r' \\ 0 & r \neq r' \end{cases},$$

for some  $r' \in \{1, 2, \dots, R\}$ , so that (D.11) becomes

$$\left( \mathbf{D}^{(i)} \right)^{-1} - \left( \tilde{\mathbf{A}}^{(i)} \right)^\top \left[ \sum_{j=0}^N \lambda_{ij}^{(r')} \left( \mathbf{D}^{(j)} \right)^{-1} \right] \tilde{\mathbf{A}}^{(i)} \succ 0, \quad (D.12)$$

where we let  $\mathbf{D}^{(i)} = \left( \tilde{\mathbf{D}}^{(i)} \right)^{-1}$ . The sum in (D.12) is of the form (A.8), namely a weighted sum of square matrices. Rewriting it as indicated by (A.9) results in

$$\left( \mathbf{D}^{(i)} \right)^{-1} - \left( \tilde{\mathbf{A}}^{(i)} \right)^\top \left( \underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi} \right) \bar{\mathbf{D}}^{-1} \left( \underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi} \right)^\top \tilde{\mathbf{A}}^{(i)} \succ 0,$$

with  $\underline{\lambda}^{(i,r')}$  and  $\bar{\mathbf{D}}$  as per (6.30) and (6.31). We then apply Theorem A.7 to obtain that

$$\begin{bmatrix} (\mathbf{D}^{(i)})^{-1} & (\tilde{\mathbf{A}}^{(i)})^T (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi}) \\ (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi})^T \tilde{\mathbf{A}}^{(i)} & \bar{\mathbf{D}} \end{bmatrix} \succ 0.$$

Another application of Theorem A.7 reveals that also

$$\bar{\mathbf{D}} - (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi})^T \tilde{\mathbf{A}}^{(i)} \mathbf{D}^{(i)} (\tilde{\mathbf{A}}^{(i)})^T (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi}) \succ 0,$$

holds. Denote the matrix on the left by  $\mathbf{\Pi}^{(i,r')}$ . Since  $\mathbf{\Pi}^{(i,r')} \succ 0$ , we have  $(\mathbf{\Pi}^{(i,r')})^{-1} \succ 0$  and hence

$$(\tilde{\mathbf{A}}^{(i)})^T (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi}) (\mathbf{\Pi}^{(i,r')})^{-1} (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi})^T \tilde{\mathbf{A}}^{(i)} \succcurlyeq 0.$$

Now fix  $\beta^{(i)} > 0$  such that

$$\frac{\mathbf{D}^{(i)} + 2\beta^{(i)}\mathbf{I}_{n_\psi}}{(\beta^{(i)})^2} \succ (\tilde{\mathbf{A}}^{(i)})^T (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi}) (\mathbf{\Pi}^{(i,r')})^{-1} (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi})^T \tilde{\mathbf{A}}^{(i)},$$

holds. Note that by Theorem A.5 such  $\beta^{(i)}$  always exists. Thus, it holds

$$\mathbf{D}^{(i)} + 2\beta^{(i)}\mathbf{I}_{n_\psi} - (\beta^{(i)})^2 (\tilde{\mathbf{A}}^{(i)})^T (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi}) (\mathbf{\Pi}^{(i,r')})^{-1} (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi})^T \tilde{\mathbf{A}}^{(i)} \succ 0,$$

which, again by virtue of Theorem A.7, is equivalent to

$$\mathbf{\Xi}^{(i,r')} = \begin{bmatrix} \mathbf{D}^{(i)} + 2\beta^{(i)}\mathbf{I}_{n_\psi} & -\beta^{(i)} (\tilde{\mathbf{A}}^{(i)})^T (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi}) \\ -\beta^{(i)} (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi})^T \tilde{\mathbf{A}}^{(i)} & \mathbf{\Pi}^{(i,r')} \end{bmatrix} \succ 0.$$

With the definition  $\mathbf{E}^{(i)} = \mathbf{D}^{(i)} + \beta^{(i)}\mathbf{I}_{n_\psi} \succ 0$ , we rewrite  $\mathbf{\Xi}^{(i,r')}$  as

$$\mathbf{\Xi}^{(i,r')} = \begin{bmatrix} 2\mathbf{E}^{(i)} - \mathbf{D}^{(i)} & (\mathbf{D}^{(i)} - \mathbf{E}^{(i)}) (\tilde{\mathbf{A}}^{(i)})^T (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi}) \\ (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi})^T \tilde{\mathbf{A}}^{(i)} (\mathbf{D}^{(i)} - \mathbf{E}^{(i)}) & \mathbf{\Pi}^{(i,r')} \end{bmatrix},$$

which in turn can be written as the product

$$\mathbf{\Xi}^{(i,r')} = \mathbf{\Omega}^{(i,r')} \begin{bmatrix} 2\mathbf{E}^{(i)} - \mathbf{D}^{(i)} & \mathbf{E}^{(i)} (\tilde{\mathbf{A}}^{(i)})^T (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi}) \\ (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi})^T \tilde{\mathbf{A}}^{(i)} \mathbf{E}^{(i)} & \bar{\mathbf{D}} \end{bmatrix} (\mathbf{\Omega}^{(i,r')})^T,$$

with

$$\mathbf{\Omega}^{(i,r')} = \begin{bmatrix} \mathbf{I}_{n_\psi} & \mathbf{0} \\ (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi})^T \tilde{\mathbf{A}}^{(i)} & \mathbf{I}_{(N+1)n_\psi} \end{bmatrix},$$

and where we used the definition of  $\mathbf{\Pi}^{(i,r')}$ . Since  $\mathbf{\Xi}^{(i,r')} \succ 0$  and  $\mathbf{\Omega}^{(i,r')}$  invertible, we finally conclude that

$$\begin{bmatrix} 2\mathbf{E}^{(i)} - \mathbf{D}^{(i)} & \mathbf{E}^{(i)} (\tilde{\mathbf{A}}^{(i)})^T (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi}) \\ (\underline{\lambda}^{(i,r')} \otimes \mathbf{I}_{n_\psi})^T \tilde{\mathbf{A}}^{(i)} \mathbf{E}^{(i)} & \bar{\mathbf{D}} \end{bmatrix} \succ 0,$$

by Sylvester's law of inertia (cf. Theorem A.3). The assertion (ii) then follows because we chose  $r'$  arbitrarily.

For the opposite direction, assume now that (ii) holds. Then, from Theorem A.7, we have that

$$2\mathbf{E}^{(i)} - \mathbf{D}^{(i)} \succ 0,$$

must be true, and hence also (cf. Lemma A.3)

$$\mathbf{E}^{(i)} \left( \mathbf{D}^{(i)} \right)^{-1} \mathbf{E}^{(i)} \succcurlyeq 2\mathbf{E}^{(i)} - \mathbf{D}^{(i)} \succ 0.$$

Thus, we may write (6.29) as

$$\begin{bmatrix} \mathbf{E}^{(i)} \left( \mathbf{D}^{(i)} \right)^{-1} \mathbf{E}^{(i)} & \mathbf{E}^{(i)} \left( \tilde{\mathbf{A}}^{(i)} \right)^{\text{T}} \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right) \\ \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right)^{\text{T}} \tilde{\mathbf{A}}^{(i)} \mathbf{E}^{(i)} & \bar{\mathbf{D}} \end{bmatrix} \succ 0, \quad (\text{D.13})$$

for  $i = 0, 1, \dots, N$  and  $r = 1, 2, \dots, R$ . Since (D.13) can be written as

$$\begin{bmatrix} \mathbf{E}^{(i)} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{D}} \end{bmatrix} \begin{bmatrix} \left( \mathbf{D}^{(i)} \right)^{-1} & \left( \tilde{\mathbf{A}}^{(i)} \right)^{\text{T}} \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right) \bar{\mathbf{D}}^{-1} \\ \bar{\mathbf{D}}^{-1} \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right)^{\text{T}} \tilde{\mathbf{A}}^{(i)} & \bar{\mathbf{D}}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{E}^{(i)} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{D}} \end{bmatrix} \succ 0,$$

Sylvester's law of inertia (cf. Theorem A.3) allows us to state that

$$\begin{bmatrix} \left( \mathbf{D}^{(i)} \right)^{-1} & \left( \tilde{\mathbf{A}}^{(i)} \right)^{\text{T}} \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right) \bar{\mathbf{D}}^{-1} \\ \bar{\mathbf{D}}^{-1} \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right)^{\text{T}} \tilde{\mathbf{A}}^{(i)} & \bar{\mathbf{D}}^{-1} \end{bmatrix} \succ 0.$$

By Theorem A.7, the Schur complement of  $\left( \mathbf{D}^{(i)} \right)^{-1}$  is

$$\begin{aligned} & \left( \mathbf{D}^{(i)} \right)^{-1} - \left( \tilde{\mathbf{A}}^{(i)} \right)^{\text{T}} \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right) \bar{\mathbf{D}}^{-1} \bar{\mathbf{D}} \bar{\mathbf{D}}^{-1} \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right)^{\text{T}} \tilde{\mathbf{A}}^{(i)} \\ &= \left( \mathbf{D}^{(i)} \right)^{-1} - \left( \tilde{\mathbf{A}}^{(i)} \right)^{\text{T}} \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right) \bar{\mathbf{D}}^{-1} \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right)^{\text{T}} \tilde{\mathbf{A}}^{(i)} \\ &= \left( \mathbf{D}^{(i)} \right)^{-1} - \left( \tilde{\mathbf{A}}^{(i)} \right)^{\text{T}} \left[ \sum_{j=0}^N \lambda_{ij}^{(r)} \left( \mathbf{D}^{(j)} \right)^{-1} \right] \tilde{\mathbf{A}}^{(i)} \\ &\succ 0, \end{aligned}$$

where for the last equality we used that  $\left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right) \bar{\mathbf{D}}^{-1} \left( \underline{\lambda}^{(i,r)} \otimes \mathbf{I}_{n_\psi} \right)^{\text{T}}$  is of the form (A.9).

Multiplying by  $\alpha_k^{(r)} \geq 0$  yields

$$\alpha_k^{(r)} \left( \mathbf{D}^{(i)} \right)^{-1} - \alpha_k^{(r)} \left( \tilde{\mathbf{A}}^{(i)} \right)^{\text{T}} \left[ \sum_{j=0}^N \lambda_{ij}^{(r)} \left( \mathbf{D}^{(j)} \right)^{-1} \right] \tilde{\mathbf{A}}^{(i)} \succ 0. \quad (\text{D.14})$$

Since (D.14) is valid for all  $r = 1, 2, \dots, R$ , we have for the sum

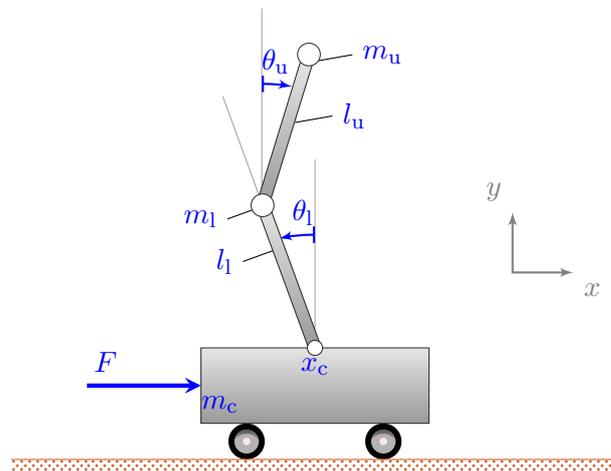
$$\sum_{r=1}^R \alpha_k^{(r)} \left( \mathbf{D}^{(i)} \right)^{-1} - \sum_{j=0}^N \sum_{r=1}^R \alpha_k^{(r)} \left( \tilde{\mathbf{A}}^{(i)} \right)^{\text{T}} \lambda_{ij}^{(r)} \left( \mathbf{D}^{(j)} \right)^{-1} \tilde{\mathbf{A}}^{(i)} \succ 0.$$

The assertion (i) then follows from the constraint  $\sum_{r=1}^R \alpha_k^{(r)} = 1$  when we let  $\tilde{\mathbf{D}}^{(i)} = \left( \mathbf{D}^{(i)} \right)^{-1}$ .  $\square$



## Equations of Motion of the Double Inverted Pendulum

The double inverted pendulum is sketched in Fig. E.1. It consists of two pendulums that are mounted on top of each other on a cart. The two pendulums can swing freely in the plane, and the cart has to move horizontally to keep them upright. To that end, an input force (in newtons) is applied to drive the cart. Both pendulums have massless rods, and the masses of the pendulum bobs and the cart are



**Figure E.1:** Illustration of the double inverted pendulum.

assumed concentrated in single points. Furthermore, we assume that the cart does not move vertically.

Using the parameters listed in Table E.1, the configuration of the system is fully described by the generalized coordinates  $\underline{q} \in \mathbb{R}^3$

$$\underline{q} = \begin{bmatrix} x_c \\ \theta_l \\ \theta_u \end{bmatrix},$$

**Table E.1:** Parameters of the double inverted pendulum.

Parameter	Unit	Symbol
Mass of cart	kg	$m_c$
Mass of lower pendulum bob	kg	$m_l$
Mass of upper pendulum bob	kg	$m_u$
Length of lower pendulum rod	m	$l_l$
Length of upper pendulum rod	m	$l_u$
Gravitational acceleration	m/s <sup>2</sup>	$g$
Input force applied to cart	N	$F$
Damping coefficient of cart	N s/m	$b_c$
Damping coefficient of lower pendulum joint	N s m	$b_l$
Damping coefficient of upper pendulum joint	N s m	$b_u$
Disturbance acting on cart	N	$w_c$
Disturbance acting on tip of lower pendulum	Nm	$w_l$
Disturbance acting on tip of upper pendulum	Nm	$w_u$

where  $x_c$  denotes the horizontal position of the cart (in meters) and where  $\theta_l$  and  $\theta_u$  denote the deviations (in radians) of the lower and the upper pendulum rod from the upright position, chosen such that positive values correspond to the clockwise direction.

The positions of the masses  $m_c$ ,  $m_l$ , and  $m_u$  are given by

$$\underline{x}_c = \begin{bmatrix} x_c \\ 0 \end{bmatrix}, \quad \underline{x}_l = \begin{bmatrix} x_c + \sin(\theta_l) l_l \\ \cos(\theta_l) l_l \end{bmatrix}, \quad \underline{x}_u = \begin{bmatrix} x_c + \sin(\theta_l) l_l + \sin(\theta_u) l_u \\ \cos(\theta_l) l_l + \cos(\theta_u) l_u \end{bmatrix}. \quad (\text{E.1})$$

Hence, the potential energy  $V$  of the system is

$$V = m_l g \cos(\theta_l) l_l + m_u g (\cos(\theta_l) l_l + \cos(\theta_u) l_u). \quad (\text{E.2})$$

From (E.1), we get that the velocities of the masses are

$$\dot{\underline{x}}_c = \begin{bmatrix} \dot{x}_c \\ 0 \end{bmatrix}, \quad \dot{\underline{x}}_l = \begin{bmatrix} \dot{x}_c + \cos(\theta_l) \dot{\theta}_l l_l \\ -\sin(\theta_l) \dot{\theta}_l l_l \end{bmatrix}, \quad \dot{\underline{x}}_u = \begin{bmatrix} \dot{x}_c + \cos(\theta_l) \dot{\theta}_l l_l + \cos(\theta_u) \dot{\theta}_u l_u \\ -\sin(\theta_l) \dot{\theta}_l l_l - \sin(\theta_u) \dot{\theta}_u l_u \end{bmatrix},$$

so that the kinetic energy  $T$  of the system is

$$T = \frac{1}{2} m_c \dot{x}_c^2 + \frac{1}{2} m_l \left[ (\dot{x}_c + \cos(\theta_l) \dot{\theta}_l l_l)^2 + (-\sin(\theta_l) \dot{\theta}_l l_l)^2 \right] + \frac{1}{2} m_u \left[ (\dot{x}_c + \cos(\theta_l) \dot{\theta}_l l_l + \cos(\theta_u) \dot{\theta}_u l_u)^2 + (-\sin(\theta_l) \dot{\theta}_l l_l - \sin(\theta_u) \dot{\theta}_u l_u)^2 \right]. \quad (\text{E.3})$$

The equations of motions can generally be derived from the Euler-Lagrange equations given by

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = \underline{Q},$$

with  $L = T - V$  the Lagrangian,  $\dot{q} = dq/dt$  the generalized velocities, and  $\underline{Q}$  the generalized forces.

The generalized forces are given by

$$\underline{Q} = \begin{bmatrix} F - b_c \dot{x}_c + w_c \\ -b_l \dot{\theta}_l + w_l \\ -b_u \dot{\theta}_u + w_u \end{bmatrix},$$

with  $F$  the input force applied to the cart, damping terms  $b_c \dot{x}_c$ ,  $b_l \dot{\theta}_1$ , and  $b_u \dot{\theta}_u$  to account for friction, and where  $w_c$ ,  $w_l$ , and  $w_u$  are external disturbances acting in the direction of each generalized coordinate.

Using (E.2) and (E.3), the Lagrangian is

$$\begin{aligned} L = & \frac{1}{2} m_c \dot{x}_c^2 + \frac{1}{2} m_l \left[ \left( \dot{x}_c + \cos(\theta_1) \dot{\theta}_1 l_1 \right)^2 + \left( -\sin(\theta_1) \dot{\theta}_1 l_1 \right)^2 \right] \\ & + \frac{1}{2} m_u \left[ \left( \dot{x}_c + \cos(\theta_1) \dot{\theta}_1 l_1 + \cos(\theta_u) \dot{\theta}_u l_u \right)^2 + \left( -\sin(\theta_1) \dot{\theta}_1 l_1 - \sin(\theta_u) \dot{\theta}_u l_u \right)^2 \right] \\ & - m_l g \cos(\theta_1) l_1 - m_u g (\cos(\theta_1) l_1 + \cos(\theta_u) l_u) . \end{aligned}$$

Taking the desired derivatives

$$\begin{aligned} \frac{\partial L}{\partial x_c} &= 0, \\ \frac{\partial L}{\partial \theta_1} &= l_1 \sin(\theta_1) (m_l + m_u) (g - \dot{\theta}_1 \dot{x}_c) - m_u l_1 l_u \dot{\theta}_1 \dot{\theta}_u \sin(\theta_1 - \theta_u), \\ \frac{\partial L}{\partial \theta_u} &= \theta_u \sin(\theta_u) m_u (g - \dot{\theta}_u \dot{x}_c) + m_u l_1 l_u \dot{\theta}_1 \dot{\theta}_u \sin(\theta_1 - \theta_u), \\ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{x}_c} \right) &= \ddot{x}_c (m_c + m_l + m_u) + (m_l + m_u) l_1 (\ddot{\theta}_1 \cos(\theta_1) - \dot{\theta}_1^2 \sin(\theta_1)) \\ &\quad + m_u l_u \ddot{\theta}_u \cos(\theta_u) - \dot{\theta}_u^2 \sin(\theta_u), \\ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) &= (m_l + m_u) l_1 (\ddot{x}_c \cos(\theta_1) + l_1 \ddot{\theta}_1 - \dot{x}_c \dot{\theta}_1 \sin(\theta_1)) \\ &\quad + m_u l_1 l_u (\ddot{\theta}_u \cos(\theta_1 - \theta_u) - \dot{\theta}_u (\dot{\theta}_1 - \dot{\theta}_u) \sin(\theta_1 - \theta_u)), \\ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_u} \right) &= m_u l_u (\ddot{x}_c \cos(\theta_u) + l_u \ddot{\theta}_u - \dot{x}_c \dot{\theta}_u \sin(\theta_u)) \\ &\quad + m_u l_1 l_u (\ddot{\theta}_1 \cos(\theta_1 - \theta_u) - \dot{\theta}_1 (\dot{\theta}_1 - \dot{\theta}_u) \sin(\theta_1 - \theta_u)), \end{aligned}$$

then yields the equations of motion

$$\mathbf{M}(\underline{q}) \ddot{\underline{q}} + \mathbf{C}(\underline{q}, \dot{\underline{q}}) \dot{\underline{q}} + \underline{g}(\underline{q}) = \underline{\tau}, \quad (\text{E.4})$$

where

$$\begin{aligned} \mathbf{M}(\underline{q}) &= \begin{bmatrix} m_c + m_l + m_u & l_1 (m_l + m_u) \cos(\theta_1) & m_u l_u \cos(\theta_u) \\ l_1 (m_l + m_u) \cos(\theta_1) & l_1^2 (m_l + m_u) & l_1 l_u m_u \cos(\theta_1 - \theta_u) \\ m_u l_u \cos(\theta_u) & l_1 l_u m_u \cos(\theta_1 - \theta_u) & l_u^2 m_u \end{bmatrix}, \\ \mathbf{C}(\underline{q}, \dot{\underline{q}}) &= \begin{bmatrix} b_c & -l_1 (m_l + m_u) \sin(\theta_1) \dot{\theta}_1 & -m_u l_u \sin(\theta_u) \dot{\theta}_1 \\ 0 & b_l & l_1 l_u m_u \sin(\theta_1 - \theta_u) \dot{\theta}_u \\ 0 & -l_1 l_u m_u \sin(\theta_1 - \theta_u) \dot{\theta}_1 & b_u \end{bmatrix}, \\ \underline{g}(\underline{q}) &= \begin{bmatrix} 0 \\ -g l_1 (m_l + m_u) \sin(\theta_1) \\ -g l_u m_u \sin(\theta_u) \end{bmatrix}, \quad \underline{\tau} = \begin{bmatrix} F \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} w_c \\ w_l \\ w_u \end{bmatrix}. \end{aligned}$$

The mass matrix  $\mathbf{M}(\underline{q})$  is positive definite for all  $\underline{q} \in \mathbb{R}^3$ , so that a state space representation of (E.4) is

$$\underbrace{\begin{bmatrix} \dot{\underline{q}} \\ \ddot{\underline{q}} \end{bmatrix}}_{=\dot{\underline{x}}(t)} = \underbrace{\left[ \mathbf{M}(\underline{q})^{-1} (\underline{\tau} - \mathbf{C}(\underline{q}, \dot{\underline{q}}) \dot{\underline{q}} - \underline{g}(\underline{q})) \right]}_{=f(\underline{x}(t), u(t), \underline{w}(t))}, \quad (\text{E.5})$$

with  $u(t) = F(t)$  and  $\underline{w}(t) = [w_c(t) \ w_1(t) \ w_u(t)]^T$ .

Linearizing (E.5) around the unstable upward equilibrium  $\underline{x}_e(t) = \underline{0}$  results in a linear dynamics of the form (3.2)

$$\dot{\underline{x}}(t) = \mathbf{A}_c \underline{x}(t) + \mathbf{B}_c u(t) + \mathbf{G}_c \underline{w}(t),$$

with

$$\mathbf{A}_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -\frac{g(m_1+m_u)}{m_c} & 0 & -\frac{b_c}{m_c} & \frac{b_1}{m_c l_1} & 0 \\ 0 & \frac{g(m_1+m_u)(m_c+m_1)}{m_c m_1 l_1} & -\frac{g m_u}{m_1 l_1} & \frac{b_c}{m_c l_1} & -\frac{b_1(m_c+m_1)}{m_c m_1 l_1^2} & \frac{b_u}{m_1 l_1 l_u} \\ 0 & -\frac{g(m_1+m_u)}{m_1 l_u} & \frac{g(m_1+m_u)}{m_1 l_u} & 0 & \frac{b_1}{m_1 l_1 l_u} & -\frac{b_u(m_1+m_u)}{m_1 m_u l_u^2} \end{bmatrix}, \quad (\text{E.6})$$

$$\mathbf{B}_c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m_c} \\ -\frac{1}{m_c l_1} \\ 0 \end{bmatrix}, \quad \mathbf{G}_c = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m_c} & -\frac{1}{m_c l_1} & 0 \\ -\frac{1}{m_c l_1} & \frac{m_c+m_1}{m_c m_1 l_1^2} & -\frac{1}{m_1 l_1 l_u} \\ 0 & -\frac{1}{m_1 l_1 l_u} & \frac{(m_1+m_u)}{m_1 m_u l_u} \end{bmatrix}.$$

## Parameters used in the Simulations and Numerical Examples

### F.1 Parameters used in the Simulations in Section 5.5

The probability distribution used in the simulation runs with network N1 to draw the packet delays is given by the stochastic row vector

$$\underline{p} = [0 \ 0.1 \ 0.2 \ 0.2 \ 0.15 \ 0.1 \ 0.02 \ 0.01 \ 0.01 \ 0.01 \ 0.2] . \quad (\text{F.1})$$

Its last entry indicates the packet loss probability and its first entry, denoting the probability that a packet is to be delivered without delay, is zero to reflect the behavior of real networks.

The transition matrix of the Markov chain governing the packet delays and losses in network N2 is

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0914 & 0.9086 & 0 & 0 & 0 & 0 \\ 0 & 0.4540 & 0.1788 & 0.3672 & 0 & 0 & 0 \\ 0 & 0 & 0.3654 & 0.3759 & 0.2586 & 0 & 0 \\ 0 & 0 & 0 & 0.4214 & 0.3229 & 0.2557 & 0 \\ 0 & 0 & 0 & 0 & 0.2918 & 0.4302 & 0.2780 \\ 0 & 0 & 0 & 0 & 0 & 0.1122 & 0.8878 \end{bmatrix} , \quad (\text{F.2})$$

with the last state indicating a packet loss. The transition probabilities  $p_{i0}^{\text{ca}} = \text{P}[\tau_k^{\text{ca}} = 0 | \tau_{k-1}^{\text{ca}} = i]$ ,  $i = 0, 1, \dots, 6$ , are set to zero to reflect that the instantaneous delivery of packets without delay is hardly realizable.

#### F.1.1 Parameters used in Section 5.5.1

With  $\mathbf{Q}$  and  $\mathbf{R}$  given by

$$\mathbf{Q} = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix} , \quad \mathbf{R} = 1 ,$$

according to (5.85), the unique stabilizing solution  $\mathbf{X}$  of the associated discrete-time algebraic Riccati equation

$$\mathbf{X} = \mathbf{A}^T \mathbf{X} \mathbf{A} - \mathbf{A}^T \mathbf{X} \mathbf{B} (\mathbf{B}^T \mathbf{X} \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^T \mathbf{X} \mathbf{A} + \mathbf{Q},$$

is given by

$$\mathbf{X} = \begin{bmatrix} 21.2361 & 11.2916 \\ 11.2916 & 17.9248 \end{bmatrix}. \quad (\text{F.3})$$

### F.1.2 Parameters used in Section 5.5.2

Matrices  $\mathbf{A}_c$ ,  $\mathbf{B}_c$ , and  $\mathbf{G}_c$  are given by

$$\mathbf{A}_c = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 19.62 & -0.6 & 0 \\ 0 & -58.86 & 1.2 & -0.05 \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -2 \end{bmatrix}, \quad \mathbf{G}_c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (\text{F.4})$$

With  $\mathbf{Q}$  and  $\mathbf{R}$  given by

$$\mathbf{Q} = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix}, \quad \mathbf{R} = 1,$$

according to (5.88), the unique stabilizing solution  $\mathbf{X}$  of the associated discrete-time algebraic Riccati equation

$$\mathbf{X} = \mathbf{A}^T \mathbf{X} \mathbf{A} - \mathbf{A}^T \mathbf{X} \mathbf{B} (\mathbf{B}^T \mathbf{X} \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^T \mathbf{X} \mathbf{A} + \mathbf{Q},$$

is given by

$$\mathbf{X} = 1 \cdot 10^3 \begin{bmatrix} 0.1099 & -0.1259 & 0.0886 & 0.0303 \\ -0.1259 & 2.7603 & -0.3022 & -0.0590 \\ 0.0886 & -0.3022 & 0.1620 & 0.0529 \\ 0.0303 & -0.0590 & 0.0529 & 0.0872 \end{bmatrix}. \quad (\text{F.5})$$

### F.2 Parameters used in the Numerical Example in Section 6.5

The matrix  $\mathbf{S}$  used in the simulation runs of scenario S1 is given below in (F.6). In each row of the matrix, the last entry subsumes the probability that a packet is delayed by more than three time steps or gets lost (infinite delay). The first entry of each row, indicating the probability that a packet is to be delivered without delay, is chosen very small to reflect the behavior of real networks.

$$\mathbf{S} = \begin{bmatrix} 0.0001 & 0.4071 & 0.2097 & 0.3034 & 0.0797 \\ 0.0001 & 0.2992 & 0.1598 & 0.4666 & 0.0743 \\ 0.0001 & 0.4310 & 0.1458 & 0.3886 & 0.0345 \\ 0.0001 & 0.1887 & 0.2486 & 0.5392 & 0.0234 \\ 0.0001 & 0.3083 & 0.4518 & 0.2018 & 0.0380 \\ 0.0001 & 0.3428 & 0.5174 & 0.0759 & 0.0637 \\ 0.0001 & 0.1693 & 0.7089 & 0.0604 & 0.0612 \\ 0.0001 & 0.1647 & 0.3259 & 0.4158 & 0.0936 \\ 0.0001 & 0.2824 & 0.1842 & 0.4901 & 0.0431 \\ 0.0001 & 0.1299 & 0.5976 & 0.2591 & 0.0133 \end{bmatrix}. \quad (\text{F.6})$$

The ten different transition matrices used in the simulation runs of scenario S2 are listed below in (F.7), where  $\epsilon = 10^{-15}$ . In each matrix, the transition probabilities  $p_{i0}^{ca} = P[\tau_k^{ca} = 0 | \tau_{k-1}^{ca} = i]$ ,  $i = 0, 1, \dots, 5$ , are chosen very small to reflect that the instantaneous delivery of packets without delay is hardly realizable.

$$\begin{aligned}
 & \begin{bmatrix} \epsilon & 1 - \epsilon & 0 & 0 & 0 & 0 \\ \epsilon & 0.1384 & 0.3004 & 0.2759 & 0.2363 & 0.0490 \\ \epsilon & 0.0027 & 0.2568 & 0.3609 & 0.3378 & 0.0418 \\ \epsilon & 0.3517 & 0.0355 & 0.2330 & 0.2929 & 0.0870 \\ \epsilon & 0.0436 & 0.2347 & 0.6819 & 0.0127 & 0.0271 \\ \epsilon & 0.1478 & 0.3260 & 0.3306 & 0.1258 & 0.0698 \end{bmatrix}, & \begin{bmatrix} \epsilon & 1 - \epsilon & 0 & 0 & 0 & 0 \\ \epsilon & 0.2696 & 0.0185 & 0.2499 & 0.1109 & 0.3510 \\ \epsilon & 0.2321 & 0.1963 & 0.1122 & 0.2477 & 0.2116 \\ \epsilon & 0.1704 & 0.1428 & 0.2150 & 0.1603 & 0.3115 \\ \epsilon & 0.3068 & 0.4381 & 0.1879 & 0.0311 & 0.0361 \\ \epsilon & 0.0585 & 0.0208 & 0.8813 & 0.0120 & 0.0274 \end{bmatrix}, \\
 & \begin{bmatrix} \epsilon & 1 - \epsilon & 0 & 0 & 0 & 0 \\ \epsilon & 0.2030 & 0.2254 & 0.2832 & 0.2860 & 0.0025 \\ \epsilon & 0.0489 & 0.2818 & 0.3520 & 0.0190 & 0.2983 \\ \epsilon & 0.2369 & 0.1441 & 0.2917 & 0.0165 & 0.3108 \\ \epsilon & 0.2795 & 0.3499 & 0.2486 & 0.0521 & 0.0699 \\ \epsilon & 0.1432 & 0.4386 & 0.3730 & 0.0436 & 0.0017 \end{bmatrix}, & \begin{bmatrix} \epsilon & 1 - \epsilon & 0 & 0 & 0 & 0 \\ \epsilon & 0.0451 & 0.3956 & 0.1521 & 0.2794 & 0.1277 \\ \epsilon & 0.2723 & 0.3174 & 0.0739 & 0.1177 & 0.2187 \\ \epsilon & 0.1980 & 0.2057 & 0.2939 & 0.2687 & 0.0338 \\ \epsilon & 0.0561 & 0.5274 & 0.3773 & 0.0132 & 0.0260 \\ \epsilon & 0.2913 & 0.3576 & 0.2497 & 0.0040 & 0.0975 \end{bmatrix}, \\
 & \begin{bmatrix} \epsilon & 1 - \epsilon & 0 & 0 & 0 & 0 \\ \epsilon & 0.1392 & 0.2889 & 0.2684 & 0.1055 & 0.1980 \\ \epsilon & 0.3283 & 0.2031 & 0.2826 & 0.1174 & 0.0688 \\ \epsilon & 0.3061 & 0.0103 & 0.2557 & 0.2925 & 0.1354 \\ \epsilon & 0.0643 & 0.4702 & 0.3864 & 0.0375 & 0.0415 \\ \epsilon & 0.0916 & 0.5845 & 0.2172 & 0.0882 & 0.0185 \end{bmatrix}, & \begin{bmatrix} \epsilon & 1 - \epsilon & 0 & 0 & 0 & 0 \\ \epsilon & 0.0230 & 0.3306 & 0.2057 & 0.0119 & 0.4288 \\ \epsilon & 0.0405 & 0.2541 & 0.3417 & 0.2471 & 0.1166 \\ \epsilon & 0.4266 & 0.0692 & 0.2884 & 0.1342 & 0.0815 \\ \epsilon & 0.5732 & 0.3179 & 0.0500 & 0.0282 & 0.0307 \\ \epsilon & 0.3011 & 0.1323 & 0.2779 & 0.2792 & 0.0096 \end{bmatrix}, & (F.7) \\
 & \begin{bmatrix} \epsilon & 1 - \epsilon & 0 & 0 & 0 & 0 \\ \epsilon & 0.2449 & 0.3006 & 0.0626 & 0.1867 & 0.2052 \\ \epsilon & 0.2824 & 0.1536 & 0.2127 & 0.2345 & 0.1168 \\ \epsilon & 0.2071 & 0.3403 & 0.3357 & 0.0557 & 0.0612 \\ \epsilon & 0.2350 & 0.3019 & 0.3001 & 0.0727 & 0.0902 \\ \epsilon & 0.3760 & 0.4211 & 0.1545 & 0.0003 & 0.0482 \end{bmatrix}, & \begin{bmatrix} \epsilon & 1 - \epsilon & 0 & 0 & 0 & 0 \\ \epsilon & 0.1279 & 0.3392 & 0.0600 & 0.2256 & 0.2474 \\ \epsilon & 0.1926 & 0.3001 & 0.3826 & 0.0151 & 0.1096 \\ \epsilon & 0.3053 & 0.3024 & 0.0047 & 0.2333 & 0.1543 \\ \epsilon & 0.3597 & 0.0949 & 0.4595 & 0.0835 & 0.0023 \\ \epsilon & 0.2799 & 0.3396 & 0.3048 & 0.0076 & 0.0680 \end{bmatrix}, \\
 & \begin{bmatrix} \epsilon & 1 - \epsilon & 0 & 0 & 0 & 0 \\ \epsilon & 0.1484 & 0.2721 & 0.0496 & 0.4048 & 0.1249 \\ \epsilon & 0.1256 & 0.2787 & 0.1914 & 0.0048 & 0.3996 \\ \epsilon & 0.1535 & 0.2839 & 0.2814 & 0.1878 & 0.0933 \\ \epsilon & 0.3569 & 0.2294 & 0.3456 & 0.0487 & 0.0194 \\ \epsilon & 0.0935 & 0.0454 & 0.7353 & 0.0565 & 0.0693 \end{bmatrix}, & \begin{bmatrix} \epsilon & 1 - \epsilon & 0 & 0 & 0 & 0 \\ \epsilon & 0.3047 & 0.4001 & 0.1581 & 0.0680 & 0.0691 \\ \epsilon & 0.0832 & 0.3729 & 0.1191 & 0.1122 & 0.3126 \\ \epsilon & 0.5795 & 0.1526 & 0.0233 & 0.1928 & 0.0518 \\ \epsilon & 0.3080 & 0.0188 & 0.5874 & 0.0076 & 0.0782 \\ \epsilon & 0.3114 & 0.3662 & 0.2492 & 0.0531 & 0.0201 \end{bmatrix}
 \end{aligned}$$

## F.3 Parameters used in in Chapter 7

### F.3.1 Parameters of the Double Inverted Pendulum

The parameters of the double inverted pendulum and their values used in the simulations are given in Table F.1.

### F.3.2 Parameters of the Network

Table F.2 lists the most relevant parameters of the Ethernet-based network configured in OMNeT++/INET for all simulation runs. Furthermore, unless otherwise noted, all links are perfect, that is, no bit errors occur and the propagation delay is zero.

Table F.3 lists the TCP configuration used for some of the cross traffic in the simulation runs in Section 7.3.2. Note that the TCP data flows are not affected by the congestion control CoCC.

**Table F.1:** Parameter values of the double inverted pendulum used in the simulations.

Parameter	Symbol	Value
Mass of cart	$m_c$	0.5 kg
Mass of lower pendulum bob	$m_l$	0.2 kg
Mass of upper pendulum bob	$m_u$	0.2 kg
Length of lower pendulum rod	$l_l$	0.3 m
Length of upper pendulum rod	$l_u$	0.3 m
Gravitational acceleration	$g$	9.81 m/s <sup>2</sup>
Damping coefficient of cart	$b_c$	0.1 N s/m
Damping coefficient of lower pendulum joint	$b_l$	0.002 N s m
Damping coefficient of upper pendulum joint	$b_u$	0.002 N s m

**Table F.2:** Parameters of the communication system configured in OMNeT++/INET for all simulation runs.

Maximum Transmission Unit (MTU)	1500 B
Link Layer Protocol	PPP with 7 B headers
Network Layer Protocol	IPv6 with 40 B headers
Transport Layer Protocol	UDP with 8 B datagram headers
CoCC Round Duration	50 ms

**Table F.3:** Configuration of TCP used for the cross traffic in Section 7.3.2.

Header Size	20 B
Maximum Segment Size	1440 B
Receive Window Size	2 <sup>10</sup> MTU
Window Scaling	Enabled
Limited Transmit	Enabled
Selective Acknowledgments	Disabled

### F.3.3 Parameters of the Controllers

#### MPC IMM

The length of the optimization horizon in all simulation runs is set to  $K = 100$ . The weighting matrices in the cost function (5.2) are set to

$$\mathbf{Q}_t = \mathbf{Q} = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{R}_t = \mathbf{R} = 100,$$

for  $t = 0, 1, \dots, K - 1$ , and  $\mathbf{Q}_K$  is chosen as the unique stabilizing solution  $\mathbf{X}$  of the associated discrete-time algebraic Riccati equation

$$\mathbf{X} = \mathbf{A}^T \mathbf{X} \mathbf{A} - \mathbf{A}^T \mathbf{X} \mathbf{B} (\mathbf{B}^T \mathbf{X} \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^T \mathbf{X} \mathbf{A} + \mathbf{Q},$$

which results in the matrix

$$\mathbf{X} = 1 \cdot 10^6 \begin{bmatrix} 0.0213 & -0.0491 & 0.1360 & 0.0226 & 0.0053 & 0.0197 \\ -0.0491 & 3.3608 & -4.5282 & -0.1097 & 0.1144 & -0.5573 \\ 0.1360 & -4.5282 & 7.1588 & 0.2694 & -0.0867 & 0.9005 \\ 0.0226 & -0.1097 & 0.2694 & 0.0383 & 0.0084 & 0.0384 \\ 0.0053 & 0.1144 & -0.0867 & 0.0084 & 0.0096 & -0.0084 \\ 0.0197 & -0.5573 & 0.9005 & 0.0384 & -0.0084 & 0.1147 \end{bmatrix}.$$

In the first simulation scenario (Section 7.3.1), we set  $N = 2$  and  $L = 1$ . In the second simulation scenario (Section 7.3.2), we set  $N = 8$  and  $L = 1$ .

The mapping  $f$  is

$$f_a = f(\text{QoC}) = 58 \exp[1.307 \cdot \text{QoC}] + 6.095 \cdot 10^{-8} \exp[-517.6 \cdot \text{QoC}], \quad (\text{F.8})$$

where  $f_a$  is expressed in Hz. The corresponding data rate in bit/s is obtained by multiplication with the packet size, which depends on the employed control sequence length  $N$ .

The mapping  $q$  is

$$\text{QoC} = q(\bar{e}_{\text{avg}}) = \frac{0.0044332}{\bar{e}_{\text{avg}}} - 0.5297. \quad (\text{F.9})$$

### MPC Bound

The length of the optimization horizon in all simulation runs is set to  $K = 20$ . The weighting matrices  $\mathbf{Q}_t$ ,  $\mathbf{Q}_K$ , and  $\mathbf{R}_t$  are the same as for the *MPC IMM*. In the first simulation scenario (Section 7.3.1), we set  $N = 2$  and  $L = 1$ . In the second simulation scenario (Section 7.3.2), we set  $N = 8$  and  $L = 1$ .

The mapping  $f$  is

$$f_a = f(\text{QoC}) = -1.4205 \cdot 10^5 \exp[2.3462 \cdot \text{QoC}] + 1.4210 \cdot 10^5 \exp[2.3460 \cdot \text{QoC}], \quad (\text{F.10})$$

where  $f_a$  is expressed in Hz. The corresponding data rate in bit/s is obtained by multiplication with the packet size, which depends on the employed control sequence length  $N$ .

The mapping  $q$  is

$$\text{QoC} = q(\bar{e}_{\text{avg}}) = \frac{0.004264}{\bar{e}_{\text{avg}}} - 0.3775. \quad (\text{F.11})$$

### Robust Controller

In all simulation runs, the initial controller gain  $\mathbf{L}$  is computed with  $\delta = 0.05$ . Each time the network model  $\tau_k^{\text{ca}}$  changes, we calculate  $t_{k,NN}$  by virtue of Theorem 4.1. Then, we set  $\delta = t_{k,NN}$  and recompute  $\mathbf{L}$ . In the simulations, the augmented state  $\underline{\psi}_k$  is not directly available. Hence, we use the IMM filter presented in Section 5.3.2 to provide the required state estimate and compute the control sequences according to<sup>1</sup>

$$\underline{U}_k = \mathbf{L} \hat{\underline{\psi}}_k.$$

In both simulation scenarios, we use  $N = 2$  and  $L = 1$ .

The mapping  $f$  is a cubic spline that minimizes

$$\mathcal{J}(f') = p \sum_{j=1}^{31} |f_{a,j} - f'(\text{QoC}_j)|^2 + (1-p) \int_0^1 \left| \frac{d}{ds} f'(s) \right|^2 ds, \quad (\text{F.12})$$

<sup>1</sup>Replacing  $\underline{\psi}_k$  by its estimate  $\hat{\underline{\psi}}_k$  is an application of the so-called *certainty equivalence* principle [158].

with  $p = 0.997803382803584$  and where  $(\text{QoC}_j, f_{a_j})$  is the  $j$ -th data pair obtained in Section 7.2. The corresponding data rate in bit/s is obtained by multiplication with the packet size, which depends on the employed control sequence length  $N$ .

The mapping  $q$  is

$$\text{QoC} = q(\bar{e}_{\text{avg}}) = \frac{0.007763}{\bar{e}_{\text{avg}}} - 0.49. \quad (\text{F.13})$$

# List of Algorithms

- 2.1 Conceptual Implementation of the  Interface: Controller → Communication System 10
- 2.2 Conceptual Implementation of the  Interface: Communication System → Controller 11
- 2.3 Conceptual Implementation of `ncs_doLoopStep` . . . . . 13
- 5.1 Conceptual Algorithm for Sequence-Based SMPC . . . . . 40
- 5.2 One Cycle of the Proposed IMM Filter . . . . . 53
- 5.3 Conceptual Algorithm for the Proposed Multiple Model Based SMPC . . . . . 54
- 5.4 Iterative Procedure for the Computation of the Control Laws . . . . . 69
- 5.5 Conceptual Algorithm for the Proposed SMPC . . . . . 70
- 7.1 Implementation of the  Interface: Controller → Communication System . . . . . 106
- 7.2 Implementation of the  Interface: Communication System → Controller . . . . . 106



# List of Figures

1.1	High level view of a cyber-physical system. . . . .	1
1.2	Data exchange provided by the CoCPN translator  . . . . .	2
1.3	Structure of the thesis. . . . .	3
2.1	Implicit coupling of two networked control systems sharing a link. . . . .	8
2.2	Possible mappings rate = $r(\text{QoC})$ provided by the CoCPN translator  . . . . .	10
2.3	Refined data exchange provided by the CoCPN translator  . . . . .	11
2.4	Data flow between Matlab and OMNeT++/INET within CoCPN-Sim. . . . .	12
3.1	Networked control system considered in this thesis. . . . .	15
4.1	Possible inputs at time $k$ for $N = 5$ . . . . .	24
4.2	Relationship between $\underline{U}_k$ , $\underline{\eta}_k$ , and $\underline{\eta}_{k+1}$ . . . . .	26
4.3	Dependencies between the entries of $\mathbf{T}_k$ . . . . .	28
5.1	Non-convex cost-to-go $\mathcal{V}_0$ for different values of $\mathbf{L}_0$ and $\mathbf{K}_0$ . . . . .	63
5.2	Convexified cost-to-go $\bar{\mathcal{V}}_0$ for different values of $\mathbf{L}_0$ and $\mathbf{K}_0$ . . . . .	68
5.3	Networks N1 and N2 considered in the evaluation. . . . .	71
5.4	Double integrator: Medians of the average incurred cost $\mathcal{J}_{\text{avg}}$ . . . . .	73
5.5	Double integrator: Evolution of the plant states $q_k$ and $\dot{q}_k$ with network N1. . . . .	74
5.6	Double integrator: Evolution of the plant states $q_k$ and $\dot{q}_k$ with network N2. . . . .	75
5.7	Sketch of an overhead crane that maneuvers a load. . . . .	76
5.8	Overhead crane: Medians of the average incurred cost $\mathcal{J}_{\text{avg}}$ . . . . .	77
5.9	Overhead crane: Evolution of the plant states $\theta_k$ and $\dot{\theta}_k$ with network N1. . . . .	78
5.10	Overhead crane: Evolution of the plant states $\theta_k$ and $\dot{\theta}_k$ with network N2. . . . .	79
6.1	Vertices $\mathbf{\Lambda}^{(1)}, \mathbf{\Lambda}^{(2)}, \dots, \mathbf{\Lambda}^{(6)}$ resulting from Theorem 6.3 with $N = 3$ . . . . .	86
6.2	Simplexes $\mathcal{S}^1, \mathcal{S}^2$ , and frustum $\mathcal{F}$ . . . . .	88
6.3	Norms of the sample mean $\bar{x}_k$ and the sample second moment $\bar{\mathbf{X}}_k$ over time. . . . .	97
6.4	Evolution of the plant states $q_k$ and $\dot{q}_k$ in five exemplary runs. . . . .	98
7.1	Sketch of the double inverted pendulum. . . . .	100
7.2	Basic network topology used to obtain the mappings rate = $r(\text{QoC})$ . . . . .	101
7.3	Excerpt of the recorded end-to-end delays with the basic network topology. . . . .	102
7.4	Derivation of the mapping $f_a = f(\text{QoC})$ for the <i>MPC IMM</i> . . . . .	103
7.5	Derivation of the mapping $f_a = f(\text{QoC})$ for the <i>MPC Bound</i> . . . . .	104
7.6	Derivation of the mapping $f_a = f(\text{QoC})$ for the <i>Robust Controller</i> . . . . .	105
7.7	Mappings $\text{QoC} = q(\bar{e}_{\text{avg}})$ . . . . .	107
7.8	Network topology considered in the first simulation scenario. . . . .	108
7.9	Times of operation of the NCS in the first simulation scenario. . . . .	109
7.10	Evolution of $\text{QoC}_{\text{target}}$ in the first simulation scenario. . . . .	110
7.11	Corresponding sampling rates $f_{\text{atarget}}$ for some of the NCS. . . . .	110
7.12	Excerpt of the recorded end-to-end delays in the first simulation scenario. . . . .	110
7.13	Sample mean and 1-sigma interval of the norm of the plant state for all NCS. . . . .	111
7.14	Desired target QoC and achieved QoC for all NCS. . . . .	112

7.15	Network topology considered in the second simulation scenario. . . . .	113
7.16	Excerpt of the recorded end-to-end delays in the second simulation scenario. . . . .	114
7.17	Exemplary evolutions of $QoC_{target}$ in the second simulation scenario. . . . .	114
7.18	Results of the second simulation scenario. . . . .	115
E.1	Illustration of the double inverted pendulum. . . . .	159

# List of Tables

4.1	Partitioning of the state space of $\mathcal{T}_k^{ca}$ induced by $\theta_k$ . . . . .	29
5.1	Parameters of the overhead crane used in the simulation. . . . .	76
7.1	Configuration of the eight NCS in the first simulation scenario. . . . .	108
E.1	Parameters of the double inverted pendulum. . . . .	160
F.1	Parameter values of the double inverted pendulum used in the simulations. . . . .	166
F.2	Parameters of the communication system configured in OMNeT++/INET. . . . .	166
F.3	Configuration of TCP used for the cross traffic in Section 7.3.2. . . . .	166



# Bibliography

- [1] S. K. Khaitan and J. D. McCalley, "Design Techniques and Applications of Cyberphysical Systems: A Survey," *IEEE Systems Journal*, vol. 9, no. 2, pp. 350–365, 2015.
- [2] A. Ahlen, J. Akerberg, M. Eriksson, A. J. Isaksson, T. Iwaki, K. H. Johansson, S. Knorn, T. Lindh, and H. Sandberg, "Toward Wireless Control in Industrial Process Automation: A Case Study at a Paper Mill," *IEEE Control Systems Magazine*, vol. 39, no. 5, pp. 36–57, 2019.
- [3] D. Baumann, F. Mager, U. Wetzker, L. Thiele, M. Zimmerling, and S. Trimpe, "Wireless Control for Smart Manufacturing: Recent Approaches and Open Challenges," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 441–467, 2021.
- [4] Z. Guo, Y. Zhang, X. Zhao, and X. Song, "CPS-Based Self-Adaptive Collaborative Control for Smart Production-Logistics Systems," *IEEE Transactions on Cybernetics*, vol. 51, no. 1, pp. 188–198, 2021.
- [5] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, "Cyber-physical systems in manufacturing," *CIRP Annals*, vol. 65, no. 2, pp. 621–641, 2016.
- [6] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent Manufacturing in the Context of Industry 4.0: A Review," *Engineering*, vol. 3, no. 5, pp. 616–630, 2017.
- [7] F. Tao, Q. Qi, L. Wang, and A. Nee, "Digital Twins and Cyber-Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison," *Engineering*, vol. 5, no. 4, pp. 653 – 661, 2019.
- [8] A. W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, and S. Yin, "Industrial Cyberphysical Systems: A Backbone of the Fourth Industrial Revolution," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 6–16, March 2017.
- [9] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): An analysis framework," *Computers in Industry*, vol. 101, pp. 1–12, 2018.
- [10] P. Leitão, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart Agents in Industrial Cyber-Physical Systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1086–1101, May 2016.
- [11] F. Allgöwer, J. Borges de Sousa, J. Kapinski, P. Mosterman, J. Oehlerking, P. Panciatici, M. Prandini, A. Rajhans, P. Tabuada, and P. Wenzelburger, "Position paper on the challenges posed by modern applications to cyber-physical systems theory," *Nonlinear Analysis: Hybrid Systems*, vol. 34, pp. 147–165, 2019.
- [12] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A Survey of Recent Results in Networked Control Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

- [13] X. Zhang, Q. Han, and X. Yu, "Survey on Recent Advances in Networked Control Systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1740–1752, Oct. 2016.
- [14] PROFIBUS & PROFINET International, "More than 20 million PROFINET devices on the market," <https://www.profibus.com/newsroom/press-news/more-than-20-million-profinet-devices-on-the-market/>, accessed: 2021-09-07.
- [15] R. Shaw and B. Jackman, "An introduction to FlexRay as an industrial network," in *Proceedings of the 2008 IEEE International Symposium on Industrial Electronics*, 2008, pp. 1849–1854.
- [16] EtherCAT Technology Group, "EtherCAT - the Ethernet Fieldbus," <https://www.ethercat.org/en/technology.html>, accessed: 2021-09-07.
- [17] CAN in Automation (CiA), "CAN in Automation (CiA)," <https://www.can-cia.org/>, accessed: 2021-09-07.
- [18] F.-L. Lian, J. R. Moyne, and D. M. Tilbury, "Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 66–83, 2001.
- [19] K. Liu, A. Selivanov, and E. Fridman, "Survey on time-delay approach to networked control," *Annual Reviews in Control*, vol. 48, pp. 57–79, 2019.
- [20] D. Chen, M. Nixon, and A. K. Mok, *WirelessHART : real-time mesh network for industrial automation*. New York: Springer, 2010.
- [21] F. Tramarin, A. K. Mok, and S. Han, "Real-Time and Reliable Industrial Control Over Wireless LANs: Algorithms, Protocols, and Future Directions," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1027–1052, 2019.
- [22] P. A. M. Devan, F. A. Hussin, R. Ibrahim, K. Bingi, and F. A. Khanday, "A Survey on the Application of WirelessHART for Industrial Process Monitoring and Control," *Sensors*, vol. 21, no. 15, 2021.
- [23] J. R. Moyne and D. M. Tilbury, "The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 29–47, 2007.
- [24] S. Vitturi, F. Tramarin, and L. Seno, "Industrial Wireless Networks: The Significance of Timeliness in Communication Systems," *IEEE Industrial Electronics Magazine*, vol. 7, no. 2, pp. 40–51, 2013.
- [25] F. Branz, R. Antonello, L. Schenato, F. Tramarin, and S. Vitturi, "Time-Critical Wireless Networked Embedded Systems: Feasibility and Experimental Assessment," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7732–7742, 2020.
- [26] F. Branz, R. Antonello, M. Pezzutto, S. Vitturi, F. Tramarin, and L. Schenato, "Drive-by-Wi-Fi: Model-Based Control Over Wireless at 1 kHz," *IEEE Transactions on Control Systems Technology*, pp. 1–12, 2021.
- [27] S. Lucia, M. Kögel, P. Zometa, D. E. Quevedo, and R. Findeisen, "Predictive Control, Embedded Cyberphysical Systems and Systems of Systems – A Perspective," *Annual Reviews in Control*, vol. 41, pp. 193 – 207, 2016.
- [28] F. Mager, D. Baumann, R. Jacob, L. Thiele, S. Trimpe, and M. Zimmerling, "Feedback Control Goes Wireless: Guaranteed Stability over Low-Power Multi-Hop Networks," in *Proceedings*

- of the 10th ACM/IEEE International Conference on Cyber-Physical Systems, ser. ICCPS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 97–108.
- [29] D. Baumann, F. Mager, R. Jacob, L. Thiele, M. Zimmerling, and S. Trimpe, “Fast Feedback Control over Multi-Hop Wireless Networks with Mode Changes and Stability Guarantees,” *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 2, Nov. 2019.
- [30] D. Baumann, F. Mager, M. Zimmerling, and S. Trimpe, “Control-Guided Communication: Efficient Resource Arbitration and Allocation in Multi-Hop Wireless Control Systems,” *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 127–132, 2020.
- [31] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele, “Low-Power Wireless Bus,” in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 1–14.
- [32] R. Jacob, L. Zhang, M. Zimmerling, J. Beutel, S. Chakraborty, and L. Thiele, “TTW: A Time-Triggered Wireless Design for CPS,” in *Proceedings of the 2018 Design, Automation & Test in Europe Conference & Exhibition*, 2018, pp. 865–868.
- [33] G. D. Di Girolamo and A. D’Innocenzo, “Codesign of controller, routing and scheduling in WirelessHART networked control systems,” *International Journal of Robust and Nonlinear Control*, vol. 29, no. 7, pp. 2171–2187, 2019.
- [34] Y. Z. Lun and A. D’Innocenzo, “Stabilizability of Markov jump linear systems modeling wireless networked control scenarios,” in *Proceedings of the 58th IEEE Conference on Decision and Control*, Dec. 2019, pp. 5766–5772.
- [35] Y. Z. Lun, C. Rinaldi, A. Alrish, A. D’Innocenzo, and F. Santucci, “On the impact of accurate radio link modeling on the performance of WirelessHART control networks,” in *Proceedings of the 2020 IEEE Conference on Computer Communications*, 2020, pp. 2430–2439.
- [36] W. Wang, D. Nešić, and R. Postoyan, “Emulation-based stabilization of networked control systems implemented on FlexRay,” *Automatica*, vol. 59, pp. 73–83, 2015.
- [37] ———, “Observer design for networked control systems with FlexRay,” *Automatica*, vol. 82, pp. 42–48, 2017.
- [38] F. Sygulla, R. Wittmann, P. Seiwald, T. Berninger, A. Hildebrandt, D. Wahrmann, and D. Rixen, “An EtherCAT-Based Real-Time Control System Architecture for Humanoid Robots,” in *Proceedings of the IEEE 14th International Conference on Automation Science and Engineering*, 2018, pp. 483–490.
- [39] Y. Tang, D. Zhang, D. W. C. Ho, and F. Qian, “Tracking Control of a Class of Cyber-Physical Systems via a FlexRay Communication Network,” *IEEE Transactions on Cybernetics*, vol. 49, no. 4, pp. 1186–1199, 2019.
- [40] P. Moreira, J. Serrano, T. Wlostowski, P. Loschmidt, and G. Gaderer, “White rabbit: Subnanosecond timing distribution over ethernet,” in *Proceedings of the 2009 International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, 2009, pp. 1–5.
- [41] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, “Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 1, pp. 88–145, 2019.

- [42] N. G. Nayak, F. Dürr, and K. Rothermel, “Incremental Flow Scheduling and Routing in Time-Sensitive Software-Defined Networks,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2066–2075, 2018.
- [43] S. Linsenmayer, B. W. Carabelli, F. Dürr, J. Falk, F. Allgöwer, and K. Rothermel, “Integration of Communication Networks and Control Systems Using a Slotted Transmission Classification Model,” in *Proceedings of 16th IEEE Annual Consumer Communications & Networking Conference*, 2019, pp. 1–6.
- [44] M. Barzegaran, B. Zarrin, and P. Pop, “Quality-Of-Control-Aware Scheduling of Communication in TSN-Based Fog Computing Platforms Using Constraint Programming,” in *Proceedings of the 2nd Workshop on Fog Computing and the IoT*, 2020, pp. 3:1–3:9.
- [45] P. Pop, M. L. Raagaard, M. Gutierrez, and W. Steiner, “Enabling Fog Computing for Industrial Automation Through Time-Sensitive Networking (TSN),” *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 55–61, 2018.
- [46] O. Seijo, I. Val, J. A. López-Fernández, J. Montalban, and E. Iradier, “On the use of White Rabbit for Precise Time Transfer in 5G URLLC Networks for Factory Automation Applications,” in *Proceedings of the 2019 IEEE International Conference on Industrial Cyber-Physical Systems*, 2019, pp. 385–390.
- [47] O. Seijo, J. A. López-Fernández, H. Bernhard, and I. Val, “Enhanced Timestamping Method for Subnanosecond Time Synchronization in IEEE 802.11 Over WLAN Standard Conditions,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5792–5805, 2020.
- [48] M. Jiménez-López, F. Girela-López, J. López-Jiménez, E. Marín-López, R. Rodríguez, and J. Díaz, “10 Gigabit White Rabbit: Sub-Nanosecond Timing and Data Distribution,” *IEEE Access*, vol. 8, pp. 92 999–93 010, 2020.
- [49] M. Vilgelm, O. Ayan, S. Zoppi, and W. Kellerer, “Control-aware Uplink Resource Allocation for Cyber-Physical Systems in Wireless Networks,” in *Proceedings of the 23th European Wireless Conference*, 2017, pp. 1–7.
- [50] J. Hahn, R. Schoeffauer, G. Wunder, and O. Stursberg, “Distributed MPC with Prediction of Time-Varying Communication Delay,” *IFAC-PapersOnLine*, vol. 51, no. 23, pp. 224–229, 2018, 7th IFAC Workshop on Distributed Estimation and Control in Networked Systems.
- [51] M. H. Mamduhi and S. Hirche, “Try-once-discard scheduling for stochastic networked control systems,” *International Journal of Control*, vol. 92, no. 11, pp. 2532–2546, 2019.
- [52] M. Kögel and R. Findeisen, “Combined online communication scheduling and output feedback MPC of cyber-physical systems,” in *Proceedings of the 2019 IEEE Annual Consumer Communications & Networking Conference*, 2019, pp. 1–6.
- [53] O. Ayan, M. Vilgelm, M. Klügel, S. Hirche, and W. Kellerer, “Age-of-Information vs. Value-of-Information Scheduling for Cellular Networked Control Systems,” in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, ser. ICCPS ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 109–117.
- [54] O. Ayan, M. Vilgelm, and W. Kellerer, “Optimal Scheduling for Discounted Age Penalty Minimization in Multi-Loop Networked Control,” in *Proceedings of the 2020 IEEE Annual Consumer Communications & Networking Conference*, 2020, pp. 1–7.

- 
- [55] M. Klügel, M. Mamduhi, O. Ayan, M. Vilgelm, K. H. Johansson, S. Hirche, and W. Kellerer, "Joint Cross-Layer Optimization in Real-Time Networked Control Systems," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1903–1915, 2020.
- [56] M. H. Mamduhi, D. Maity, J. S. Baras, and K. H. Johansson, "A Cross-Layer Optimal Co-Design of Control and Networking in Time-Sensitive Cyber-Physical Systems," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 917–922, 2021.
- [57] S. H. Hong, "Scheduling Algorithm of Data Sampling Times in the Integrated Communication and Control Systems," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 2, pp. 225–230, 1995.
- [58] J. Bai, E. P. Eyisi, F. Qiu, Y. Xue, and X. D. Koutsoukos, "Optimal Cross-Layer Design of Sampling Rate Adaptation and Network Scheduling for Wireless Networked Control Systems," in *Proceedings of the Third IEEE/ACM International Conference on Cyber-Physical Systems*, 2012, pp. 107–116.
- [59] A. Saifullah, C. Wu, P. B. Tiwari, Y. Xu, Y. Fu, C. Lu, and Y. Chen, "Near Optimal Rate Selection for Wireless Control Systems," *ACM Transactions on Embedded Computing Systems*, vol. 13, no. 4s, apr 2014.
- [60] D. Kim, Y. Won, S. Kim, Y. Eun, K.-J. Park, and K. H. Johansson, "Sampling Rate Optimization for IEEE 802.11 Wireless Control Systems," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, ser. ICCPS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 87–96.
- [61] K. Mirdha and M. Jung, "SERUM: A Lightweight Approach for Secure Network Route Monitoring in Industrial Networks," in *Proceedings of the 2019 IEEE Symposium on Computers and Communications*, 2019, pp. 1–6.
- [62] A. Bemporad, "Predictive Control of Teleoperated Constrained Systems with Unbounded Communication Delays," in *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 2, Dec. 1998, pp. 2133–2138.
- [63] V. Gupta, B. Sinopoli, S. Adlakha, A. Goldsmith, and R. Murray, "Receding Horizon Networked Control," in *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing*, 2006.
- [64] D. E. Quevedo and D. Nesic, "Input-to-State Stability of Packetized Predictive Control over Unreliable Networks Affected by Packet-Dropouts," *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 370–375, 2011.
- [65] D. E. Quevedo and I. Jurado, "Stability of Sequence-Based Control With Random Delays and Dropouts," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1296–1302, May 2014.
- [66] R. Yang, G. Liu, P. Shi, C. Thomas, and M. V. Basin, "Predictive Output Feedback Control for Networked Control Systems," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 512–520, 2014.
- [67] J. Østergaard and D. Quevedo, "Multiple descriptions for packetized predictive control," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 45, Apr 2016.
- [68] M. Jung and M. Zitterbart, "Cooperative Congestion Control for Cyber-Physical Systems," in *Proceedings of the 10th Mediterranean Conference on Embedded Computing*. IEEE, 2021, pp. 1–4.

- [69] ———, “Hop-By-Hop: Advancing Cooperative Congestion Control for Cyber-Physical Systems,” in *Proceedings of the 2021 IEEE Conference on Local Computer Networks*, 2021, pp. 511–518.
- [70] A. H. Levis, R. A. Schlueter, and M. Athans, “On the behaviour of optimal linear sampled-data regulators,” *International Journal of Control*, vol. 13, no. 2, pp. 343–361, 1971.
- [71] P. Khosla, “Choosing sampling rates for robot control,” in *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, vol. 4, 1987, pp. 169–174.
- [72] D. Seto, J. P. Lehoczky, L. Sha, and K. G. Shin, “On task schedulability in real-time control systems,” in *Proceedings of the 17th IEEE Real-Time Systems Symposium*, 1996, pp. 13–21.
- [73] R. Blind and F. Allgöwer, “Is it worth to retransmit lost packets in networked control systems?” in *Proceedings of the 51st IEEE Conference on Decision and Control*, 2012, pp. 1368–1373.
- [74] C. Du, A. Kong, and Y. Zhang, “Time delay and sampling rate effect on dual-stage servo control performance,” *Microsystem Technologies*, vol. 22, no. 6, pp. 1213–1219, Jun 2016.
- [75] W. Schultz and V. Rideout, “Control system performance measures: Past, present, and future,” *IRE transactions on automatic control*, no. 1, pp. 22–35, 1961.
- [76] A. P. Swanda and D. E. Seborg, “Controller performance assessment based on setpoint response data,” in *Proceedings of the 1999 American Control Conference*, vol. 6. IEEE, 1999, pp. 3863–3867.
- [77] P. D. Domański, “Performance assessment of predictive control—A survey,” *Algorithms*, vol. 13, no. 4, p. 97, 2020.
- [78] S. L. Shah, R. Patwardhan, and B. Huang, “Multivariate controller performance analysis: methods, applications and challenges,” in *AIChE Symposium Series*. New York; American Institute of Chemical Engineers; 1998, 2002, pp. 190–207.
- [79] T. J. Harris, “Assessment of control loop performance,” *The Canadian Journal of Chemical Engineering*, vol. 67, no. 5, pp. 856–861, 1989.
- [80] OpenSim Ltd., “OMNeT++ Discrete Event Simulator – Home,” <https://omnetpp.org/>, accessed: 2021-11-10.
- [81] Z. Bojthe, L. Meszaros, G. Szászok, R. Hornig, A. Varga, and A. Török, “INET Framework,” <https://inet.omnetpp.org/>, accessed: 2021-11-10.
- [82] The MathWorks, Inc, “MATLAB Compiler SDK,” <https://www.mathworks.com/products/matlab-compiler-sdk.html>, accessed: 2021-12-27.
- [83] S. Johannessen, “Time synchronization in a local area network,” *IEEE Control Systems Magazine*, vol. 24, no. 2, pp. 61–69, 2004.
- [84] D. Mills, J. Martin, J. Burbank, and W. Kasch, “Network Time Protocol Version 4: Protocol and Algorithms Specification,” Internet Requests for Comments, RFC Editor, RFC 5905, June 2010, <http://www.rfc-editor.org/rfc/rfc5905.txt>.
- [85] “IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,” *IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008)*, pp. 1–499, 2020.
- [86] R. N. Gore, E. Lisova, J. Åkerberg, and M. Björkman, “In Sync with Today’s Industrial System Clocks,” in *Proceedings of 2020 International Conference on Communication Systems & Networks*, 2020, pp. 785–790.

- [87] J. Postel, "User Datagram Protocol," Internet Requests for Comments, RFC Editor, STD 6, August 1980, <http://www.rfc-editor.org/rfc/rfc768.txt>.
- [88] ———, "Transmission Control Protocol," Internet Requests for Comments, RFC Editor, STD 7, September 1981, <http://www.rfc-editor.org/rfc/rfc793.txt>.
- [89] J. Baillieul and P. J. Antsaklis, "Control and Communication Challenges in Networked Real-Time Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 9–28, 2007.
- [90] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of Control and Estimation Over Lossy Networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.
- [91] J. Fischer, A. Hekler, M. Dolgov, and U. D. Hanebeck, "Optimal Sequence-Based LQG Control over TCP-like Networks Subject to Random Transmission Delays and Packet Losses," in *Proceedings of the 2013 American Control Conference*, Jun. 2013, pp. 1543–1549.
- [92] B. Davie, A. Charny, J. Bennet, K. Benson, J. L. Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)," Internet Requests for Comments, RFC Editor, RFC 3246, March 2002.
- [93] K. Nichols, S. Blake, F. Baker, and D. L. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," Internet Requests for Comments, RFC Editor, RFC 2474, December 1998, <http://www.rfc-editor.org/rfc/rfc2474.txt>.
- [94] S. Blake, D. L. Black, M. A. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," Internet Requests for Comments, RFC Editor, RFC 2475, December 1998, <http://www.rfc-editor.org/rfc/rfc2475.txt>.
- [95] L. Zhang, H. Gao, and O. Kaynak, "Network-Induced Constraints in Networked Control Systems—A Survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 403–416, 2013.
- [96] E. Garone, B. Sinopoli, and A. Casavola, "LQG Control over Lossy TCP-Like Networks with Probabilistic Packet Acknowledgements," *International Journal of Systems, Control and Communications*, vol. 2, no. 1-3, pp. 55–81, 2010.
- [97] M. Moayedi, Y. Foo, and Y. Soh, "Networked LQG control over unreliable channels," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 2, pp. 167–189, 2013.
- [98] H. Lin, H. Su, P. Shi, R. Lu, and Z.-G. Wu, "Estimation and LQG Control Over Unreliable Network With Acknowledgment Randomly Lost," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4074–4085, 2017.
- [99] S. Trimpe, "Predictive and self triggering for event-based state estimation," in *Proceedings of the IEEE 55th Conference on Decision and Control*, 2016, pp. 3098–3105.
- [100] J. Sijs, B. Noack, M. Lazar, and U. D. Hanebeck, "Time-periodic state estimation with event-based measurement updates," *Event-Based Control and Signal Processing*, 2016.
- [101] D. Shi, T. Chen, and L. Shi, "Event-triggered maximum likelihood state estimation," *Automatica*, vol. 50, no. 1, pp. 247–254, 2014.
- [102] L. Xu, Y. Mo, and L. Xie, "Remote State Estimation With Stochastic Event-Triggered Sensor Schedule and Packet Drops," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4981–4988, 2020.

- [103] “IEEE Standard for Ethernet,” *IEEE Std 802.3-2018 (Revision of IEEE Std 802.3-2015)*, pp. 1–5600, 2018.
- [104] “IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pp. 1–4379, 2021.
- [105] L. Schenato, “To Zero or to Hold Control Inputs With Lossy Links?” *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1093–1099, 2009.
- [106] D. A. Vivanco and A. P. Jayasumana, “A Measurement-Based Modeling Approach for Network-Induced Packet Delay,” in *Proceedings of the 2007 IEEE Conference on Local Computer Networks*, 2007, pp. 175–182.
- [107] P. Park, S. Coleri Ergen, C. Fischione, C. Lu, and K. H. Johansson, “Wireless Network Design for Control Systems: A Survey,” *IEEE Communications Surveys and Tutorials*, vol. 20, no. 2, pp. 978–1013, 2018.
- [108] P. Sadeghi, R. A. Kennedy, P. B. Rapajic, and R. Shams, “Finite-state Markov modeling of fading channels - a survey of principles and applications,” *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 57–80, 2008.
- [109] Y. Mo, E. Garone, and B. Sinopoli, “LQG control with Markovian packet loss,” in *Proceedings of the 2013 European Control Conference*, Jul. 2013, pp. 2380–2385.
- [110] C. Han, H. Zhang, and M. Fu, “Optimal filtering for networked systems with Markovian communication delays,” *Automatica*, vol. 49, no. 10, pp. 3097 – 3104, 2013.
- [111] D. Dolz, D. E. Quevedo, I. Peñarrocha, and R. Sanchis, “Performance vs complexity trade-offs for Markovian networked jump estimators,” *IFAC-PapersOnLine*, vol. 47, no. 3, pp. 7412 – 7417, 2014, 19th IFAC World Congress.
- [112] D. E. Quevedo, P. K. Mishra, R. Findeisen, and D. Chatterjee, “A Stochastic Model Predictive Controller for Systems with Unreliable Communications,” *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 57–64, 2015, 5th IFAC Conference on Nonlinear Model Predictive Control.
- [113] E. G. Peters, D. Marelli, D. E. Quevedo, and M. Fu, “Controller Design for Networked Control Systems Affected by Correlated Packet Losses,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2555 – 2560, 2017, 20th IFAC World Congress.
- [114] H. Song, S. Chen, and Y. Yam, “Sliding Mode Control for Discrete-Time Systems With Markovian Packet Dropouts,” *IEEE Transactions on Cybernetics*, vol. 47, no. 11, pp. 3669–3679, 2017.
- [115] J. Wu, L. Shi, L. Xie, and K. H. Johansson, “An Improved Stability Condition for Kalman Filtering with Bounded Markovian Packet Losses,” *Automatica*, vol. 62, pp. 32–38, 2015.
- [116] L. Xiao, A. Hassibi, and J. P. How, “Control with Random Communication Delays via a Discrete-Time Jump System Approach,” in *Proceedings of the 2000 American Control Conference*, vol. 3, Jun. 2000, pp. 2199–2204.
- [117] X. R. Li and Y. Zhang, “Numerically Robust Implementation of Multiple-Model Algorithms,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 1, pp. 266–278, 2000.

- [118] P. J. Antsaklis, “Special issue on hybrid systems: theory and applications a brief introduction to the theory and applications of hybrid systems,” *Proceedings of the IEEE*, vol. 88, no. 7, pp. 879–887, 2000.
- [119] J. Lygeros and M. Prandini, “Stochastic Hybrid Systems: A Powerful Framework for Complex, Large Scale Applications,” *European Journal of Control*, vol. 16, no. 6, pp. 583–594, 2010.
- [120] F. Zhu and P. J. Antsaklis, “Optimal control of hybrid switched systems: A brief survey,” *Discrete Event Dynamic Systems*, vol. 25, no. 3, pp. 345–364, Sep 2015.
- [121] R. Krtolica, Ü. Özgüner, H. Chan, H. Göktas, J. Winkelman, and M. Liubakka, “Stability of linear feedback systems with random communication delays,” *International Journal of Control*, vol. 59, no. 4, pp. 925–953, 1994.
- [122] A. R. Fioravanti, A. P. Gonçalves, and J. C. Geromel, “Filtering of Discrete-Time Markov Jump Linear Systems with Cluster Observation: An Approach to Gilbert-Elliot’s Network Channel,” in *Proceedings of the 2009 European Control Conference*. IEEE, 2009, pp. 2283–2288.
- [123] J. Fischer, “Optimal Sequence-Based Control of Networked Linear Systems,” Ph.D. dissertation, Karlsruher Institut für Technologie (KIT), 2014.
- [124] M. Dolgov, J. Fischer, and U. D. Hanebeck, “Infinite-Horizon Sequence-based Networked Control without Acknowledgments,” in *Proceedings of the 2015 American Control Conference*, Jul. 2015, pp. 402–408.
- [125] H. Lin, H. Su, M. Z. Chen, Z. Shu, R. Lu, and Z. Wu, “On stability and convergence of optimal estimation for networked control systems with dual packet losses without acknowledgment,” *Automatica*, vol. 90, pp. 81 – 90, 2018.
- [126] H. Lin, J. Lam, M. Z. Q. Chen, Z. Shu, and Z. Wu, “Interacting Multiple Model Estimator for Networked Control Systems: Stability, Convergence, and Performance,” *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 928–943, 2019.
- [127] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-Time Markov Jump Linear Systems*. Springer Science & Business Media, 2006.
- [128] C. J. Burke and M. Rosenblatt, “A Markovian Function of a Markov Chain,” *The Annals of Mathematical Statistics*, vol. 29, no. 4, pp. 1112–1122, 1958.
- [129] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. Springer, 1976.
- [130] N.-F. Peng, “On weak lumpability of a finite Markov chain,” *Statistics & Probability Letters*, vol. 27, no. 4, pp. 313 – 318, 1996.
- [131] P. Buchholz, “Exact and ordinary lumpability in finite Markov chains,” *Journal of Applied Probability*, vol. 31, no. 1, p. 59–75, 1994.
- [132] W. C. Swope, J. W. Pitera, and F. Suits, “Describing Protein Folding Kinetics by Molecular Dynamics Simulations. 1. Theory,” *The Journal of Physical Chemistry B*, vol. 108, no. 21, pp. 6571–6581, 2004.
- [133] Y. Guédon, Y. d’Aubenton Carafa, and C. Thermes, “Analysing grouping of nucleotides in DNA sequences using lumped processes constructed from Markov chains,” *Journal of Mathematical Biology*, vol. 52, pp. 343–72, 04 2006.
- [134] K. H. Hoffmann and P. Salamon, “Accuracy of coarse grained Markovian dynamics,” *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 18, pp. 3086 – 3094, 2011.

- [135] M. I. Loizides and A. N. Yannacopoulos, "Lumpable Markov chains in risk management," *Optimization Letters*, vol. 6, no. 3, pp. 489–501, Mar 2012.
- [136] K. Georgiou, G. Domazakis, D. Pappas, and A. Yannacopoulos, "Markov chain lumpability and applications to credit risk modelling in compliance with the International Financial Reporting Standard 9 framework," *European Journal of Operational Research*, vol. 292, no. 3, pp. 1146–1164, 2021.
- [137] P. Buchholz, "Lumpability and nearly-lumpability in hierarchical queueing networks," in *Proceedings of 1995 IEEE International Computer Performance and Dependability Symposium*, 1995, pp. 82–91.
- [138] S. Kazemian and I. Nikolaidis, "Lumped Markovian Estimation for Wi-Fi Channel Utilization Prediction," in *Proceedings of the 2019 International Conference on Network and Service Management*, 2019, pp. 1–5.
- [139] A. P. Gonçalves, A. R. Fioravanti, and J. C. Geromel, "Markov jump linear systems and filtering through network transmitted measurements," *Signal Processing*, vol. 90, no. 10, pp. 2842 – 2850, 2010.
- [140] Z. Du, N. Ozay, and L. Balzano, "Mode Clustering for Markov Jump Systems," in *Proceedings of the 2019 IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2019, pp. 126–130.
- [141] K. H. Hoffmann and P. Salamon, "Bounding the lumping error in Markov chain dynamics," *Applied Mathematics Letters*, vol. 22, no. 9, pp. 1471 – 1475, 2009.
- [142] A. M. S. Barreto and M. D. Fragoso, "Computing the Stationary Distribution of a Finite Markov Chain Through Stochastic Factorization," *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 4, pp. 1513–1523, 2011.
- [143] P. Zhao, Y. Kang, and Y. Zhao, "A Brief Tutorial and Survey on Markovian Jump Systems: Stability and Control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 5, no. 2, pp. 37–45, 2019.
- [144] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103–123, 1998.
- [145] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part V. Multiple-Model Methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, 2005.
- [146] C. J. Lee, J. M. Pak, C. K. Ahn, K. M. Min, P. Shi, and M. T. Lim, "Multi-target FIR tracking algorithm for Markov jump linear systems based on true-target decision-making," *Neurocomputing*, vol. 168, pp. 298–307, 2015.
- [147] L. Blackmore, A. Bektassov, M. Ono, and B. C. Williams, "Robust, optimal predictive control of jump markov linear systems using particles," in *Proceedings of the 2007 International Workshop on Hybrid Systems: Computation and Control*. Springer, 2007, pp. 104–117.
- [148] A. N. Vargas, E. F. Costa, and J. B. do Val, "On the control of Markov jump linear systems with no mode observation: application to a DC Motor device," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 10, pp. 1136–1150, 2013.

- [149] A. N. Vargas, D. C. Bortolin, E. F. Costa, and J. B. Val, "Gradient-Based Optimization Techniques for the Design of Static Controllers for Markov Jump Linear Systems with Unobservable Modes," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 28, no. 3, pp. 239–253, 2014.
- [150] J. Tonne and O. Stursberg, *Constrained Model Predictive Control of Processes with Uncertain Structure Modeled by Jump Markov Linear Systems*. Cham: Springer International Publishing, 2016, pp. 335–361.
- [151] A. de Oliveira and O. Costa, "Mixed Control of Hidden Markov Jump Systems," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 4, pp. 1261–1280, 2018.
- [152] A. Hekler, J. Fischer, and U. D. Hanebeck, "Sequence-Based Control for Networked Control Systems Based on Virtual Control Inputs," in *Proceedings of the 51st IEEE Conference on Decision and Control*, Dec. 2012, pp. 7–13.
- [153] D. P. Bertsekas, *Dynamic programming and optimal control*, 2nd ed. Belmont, Mass.: Athena Scientific, 2000, vol. 1.
- [154] ———, "Dynamic Programming and Suboptimal Control: A Survey from ADP to MPC," *European Journal of Control*, vol. 11, no. 4, pp. 310 – 334, 2005.
- [155] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proceedings of the 51st IEEE Conference on Decision and Control*, Dec. 2012, pp. 3270–3285.
- [156] C. Peng and F. Li, "A survey on recent advances in event-triggered communication and control," *Information Sciences*, vol. 457-458, pp. 113–125, 2018.
- [157] K. Zhang, B. Zhou, and G. Duan, "Event-Triggered and Self-Triggered Control of Discrete-Time Systems With Input Constraints," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–10, 2020.
- [158] Y. Bar-Shalom and E. Tse, "Dual Effect, Certainty Equivalence, and Separation in Stochastic Control," *IEEE Transactions on Automatic Control*, vol. 19, no. 5, pp. 494–500, Oct. 1974.
- [159] Y. Bar-Shalom, "Stochastic dynamic programming: Caution and probing," *IEEE Transactions on Automatic Control*, vol. 26, no. 5, pp. 1184–1195, 1981.
- [160] S. V. Raković and W. S. Levine, Eds., *Handbook of Model Predictive Control*. Cham: Springer International Publishing, 2019.
- [161] D. Lenz, T. Kessler, and A. Knoll, "Stochastic Model Predictive Controller with Chance Constraints for Comfortable and Safe Driving Behavior of Autonomous Vehicles," in *Proceedings of the 2015 IEEE Intelligent Vehicles Symposium*, 2015, pp. 292–297.
- [162] M. Farina, L. Giulioni, and R. Scattolini, "Stochastic Linear Model Predictive Control with Chance Constraints – A review," *Journal of Process Control*, vol. 44, pp. 53 – 67, 2016.
- [163] T. A. N. Heirung, J. A. Paulson, J. O’Leary, and A. Mesbah, "Stochastic model predictive control — how does it work?" *Computers & Chemical Engineering*, vol. 114, pp. 158 – 170, 2018.
- [164] A. Mesbah, "Stochastic Model Predictive Control: An Overview and Perspectives for Future Research," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016.

- [165] M. A. Sehr and R. R. Bitmead, “Stochastic output-feedback model predictive control,” *Automatica*, vol. 94, pp. 315 – 323, 2018.
- [166] P. K. Mishra, D. E. Quevedo, and D. Chatterjee, “Dropout feedback parametrized policies for stochastic predictive controller,” *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 59 – 64, 2016, 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016.
- [167] P. K. Mishra, D. Chatterjee, and D. E. Quevedo, “Sparse and Constrained Stochastic Predictive control for Networked Systems,” *Automatica*, vol. 87, pp. 40 – 51, 2018.
- [168] ———, “Stochastic Predictive Control under Intermittent Observations and Unreliable actions,” *Automatica*, vol. 118, p. 109012, 2020.
- [169] I. Jurado, P. Millán, D. Quevedo, and F. Rubio, “Stochastic MPC with applications to process control,” *International Journal of Control*, vol. 88, no. 4, pp. 792–800, 2015.
- [170] B. Li, Y. Ma, T. Westenbroek, C. Wu, H. Gonzalez, and C. Lu, “Wireless Routing and Control: A Cyber-Physical Case Study,” in *Proceedings of the 7th ACM/IEEE International Conference on Cyber-Physical Systems*, 2016, pp. 1–10.
- [171] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [172] D. Q. Mayne, “Competing methods for robust and stochastic MPC,” *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 169–174, 2018, 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018.
- [173] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. Sastry, “Optimal control with unreliable communication: the TCP case,” in *Proceedings of the 2005 American Control Conference*, 2005, pp. 3354–3359.
- [174] B. E. Griffiths and K. A. Loparo, “Optimal Control of Jump-Linear Gaussian Systems,” *International Journal of Control*, vol. 42, no. 4, pp. 791–819, 1985.
- [175] M. Dolgov, “Approximate Stochastic Optimal Control of Smooth Nonlinear Systems and Piecewise Linear Systems,” Ph.D. dissertation, Karlsruher Institut für Technologie (KIT), 2018.
- [176] G. Pan and Y. Bar-Shalom, “Stabilization of Jump Linear Gaussian Systems without Mode Observations,” *International Journal of Control*, vol. 64, no. 4, pp. 631–661, 1996.
- [177] A. N. Vargas, J. B. R. do Val, and E. F. Costa, “Receding Horizon Control of Markov Jump Linear Systems Subject to Noise and Unobserved State Chain,” in *Proceedings of the 43rd IEEE Conference on Decision and Control*, vol. 4, Dec. 2004, pp. 4381–4386.
- [178] A. N. Vargas, W. Furloni, and J. B. do Val, “Constrained model predictive control of jump linear systems with noise and non-observed Markov state,” in *Proceedings of the 2006 American Control Conference*, 2006, pp. 929–934.
- [179] A. Sala, M. Hernández-Mejías, and C. Ariño, “Stable receding-horizon scenario predictive control for Markov-jump linear systems,” *Automatica*, vol. 86, pp. 121–128, 2017.
- [180] A. N. Vargas, L. Acho, G. Pujol, E. F. Costa, J. Y. Ishihara, and J. B. do Val, “Output feedback of Markov jump linear systems with no mode observation: An automotive throttle application,” *International Journal of Robust and Nonlinear Control*, vol. 26, no. 9, pp. 1980–1993, 2016.

- 
- [181] M. Dolgov and U. D. Hanebeck, "Static Output-Feedback Control of Markov Jump Linear Systems Without Mode Observation," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5401–5406, 2017.
- [182] Y. A. Zabala and O. L. Costa, "Static output constrained control for discrete-time hidden Markov jump linear systems," *IEEE Access*, vol. 8, pp. 62 969–62 979, 2020.
- [183] G. Ackerson and K. Fu, "On state estimation in switching environments," *IEEE Transactions on Automatic Control*, vol. 15, no. 1, pp. 10–17, 1970.
- [184] H. Blom and Y. Bar-Shalom, "The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.
- [185] O. Costa, "Linear Minimum Mean Square Error Estimation for Discrete-Time Markovian Jump Linear Systems," *IEEE Transactions on Automatic Control*, vol. 39, no. 8, pp. 1685–1689, 1994.
- [186] O. L. V. Costa and S. Guerra, "Stationary Filter for Linear Minimum Mean Square Error Estimator of Discrete-Time Markovian Jump Systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1351–1356, 2002.
- [187] M. H. Terra, J. Y. Ishihara, and G. Jesus, "Information Filtering and Array Algorithms for Discrete-Time Markovian Jump Linear Systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 158–162, Jan. 2009.
- [188] C. E. Seah and I. Hwang, "Performance Analysis of Kalman Filter Based Hybrid Estimation Algorithms," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 13 498–13 503, 2008, 17th IFAC World Congress.
- [189] L. Campo and Y. Bar-Shalom, "Control of Discrete-Time Hybrid Stochastic Systems," *IEEE Transactions on Automatic Control*, vol. 37, no. 10, pp. 1522–1527, Oct. 1992.
- [190] Y. Bar-Shalom, S. Challa, and H. Blom, "IMM Estimator Versus Optimal Estimator for Hybrid Systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 986–991, 2005.
- [191] Y. Bar-Shalom and H. Chen, "IMM Estimator with Out-of-Sequence Measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 90–98, 2005.
- [192] J. P. Georges, D. Theilliol, J. C. Ponsart, and C. Aubrun, "Networked Control system with intermittent observations: FDI/FTC design based on Interacting Multiple Model approach," in *Proceedings of the 17th Mediterranean Conference on Control and Automation*, 2009, pp. 528–533.
- [193] J. Fischer, A. Hekler, and U. D. Hanebeck, "State Estimation in Networked Control Systems," in *Proceedings of the 15th International Conference on Information Fusion*, Singapore, Jul. 2012, pp. 1947–1954.
- [194] L. Schenato, "Optimal estimation in networked control systems subject to random delay and packet drop," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1311–1317, 2008.
- [195] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.
- [196] M. Huang and S. Dey, "Stability of Kalman filtering with Markovian packet losses," *Automatica*, vol. 43, no. 4, pp. 598–607, 2007.

- [197] E. I. Silva and M. A. Solis, "An alternative look at the constant-gain Kalman filter for state estimation over erasure channels," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3259–3265, 2013.
- [198] M. Dolgov, G. Kurz, and U. D. Hanebeck, "Finite-horizon Dynamic Compensation of Markov Jump Linear Systems without Mode Observation," in *Proceedings of the 55th IEEE Conference on Decision and Control*, Dec. 2016, pp. 2757–2762.
- [199] M. Athans, "The Matrix Minimum Principle," *Information and Control*, vol. 11, no. 5, pp. 592 – 606, 1967.
- [200] J. B. do Val and T. Başar, "Receding Horizon Control of Jump Linear Systems and a Macroeconomic Policy Problem," *Journal of Economic Dynamics and Control*, vol. 23, no. 8, pp. 1099 – 1131, 1999.
- [201] P. A. Bekker, "The Positive Semidefiniteness of Partitioned Matrices," *Linear Algebra and its Applications*, vol. 111, pp. 261–278, 1988.
- [202] D. S. Bernstein, L. D. Davis, and D. C. Hyland, "The optimal projection equations for reduced-order, discrete-time modeling, estimation, and control," *Journal of Guidance, Control, and Dynamics*, vol. 9, no. 3, pp. 288–293, 1986.
- [203] D. S. Bernstein and W. M. Haddad, "Optimal projection equations for discrete-time fixed-order dynamic compensation of linear systems with multiplicative white noise," *International Journal of Control*, vol. 46, no. 1, pp. 65–73, 1987.
- [204] L. Van Willigenburg and W. De Koning, "Numerical Algorithms and Issues Concerning the Discrete-Time Optimal Projection Equations," *European Journal of Control*, vol. 6, no. 1, pp. 93–110, 2000.
- [205] W. De Koning, "Compensatability and optimal compensation of systems with white parameters," *IEEE Transactions on Automatic Control*, vol. 37, no. 5, pp. 579–588, 1992.
- [206] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. Sastry, "Optimal Linear LQG Control Over Lossy Networks without Packet Acknowledgment," in *Proceedings of the 45th IEEE Conference on Decision and Control*, Dec 2006, pp. 392–397.
- [207] P. H. Schönemann, "On the Formal Differentiation of Traces and Determinants," *Multivariate Behavioral Research*, vol. 20, no. 2, pp. 113–139, 1985.
- [208] B. Kagstrom and L. Westin, "Generalized Schur methods with condition estimators for solving the generalized Sylvester equation," *IEEE Transactions on Automatic Control*, vol. 34, no. 7, pp. 745–751, 1989.
- [209] F. Ding and T. Chen, "Gradient Based Iterative Algorithms for Solving a class of Matrix Equations," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1216–1221, 2005.
- [210] F. Ding, P. X. Liu, and J. Ding, "Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle," *Applied Mathematics and Computation*, vol. 197, no. 1, pp. 41–50, 2008.
- [211] A. Kittisopaporn and P. Chansangiam, "Approximated least-squares solutions of a generalized Sylvester-transpose matrix equation via gradient-descent iterative algorithm," *Advances in Difference Equations*, vol. 2021, no. 1, pp. 1–18, 2021.

- 
- [212] A. Hekler, J. Fischer, and U. D. Hanebeck, "Control over unreliable networks based on control input densities," in *Proceedings of the 15th International Conference on Information Fusion*, 2012, pp. 1277–1283.
- [213] L. Hewing and M. N. Zeilinger, "Scenario-Based Probabilistic Reachable Sets for Recursively Feasible Stochastic Model Predictive Control," *IEEE Control Systems Letters*, vol. 4, no. 2, pp. 450–455, 2020.
- [214] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *The Bell System Technical Journal*, vol. 39, no. 5, pp. 1253–1265, 1960.
- [215] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *The Bell System Technical Journal*, vol. 42, no. 5, pp. 1977–1997, 1963.
- [216] T. Farjam, T. Charalambous, and H. Wymeersch, "Timer-Based Distributed Channel Access in Networked Control Systems over Known and Unknown Gilbert-Elliott Channels," in *Proceedings of 2019 European Control Conference*, 2019, pp. 2983–2989.
- [217] V. P. Jilkov and X. R. Li, "Online Bayesian Estimation of Transition Probabilities for Markovian Jump Systems," *IEEE Transactions on Signal Processing*, vol. 52, no. 6, pp. 1620–1630, 2004.
- [218] U. Orguner and M. Demirekler, "An online sequential algorithm for the estimation of transition probabilities for jump Markov linear systems," *Automatica*, vol. 42, no. 10, pp. 1735–1744, 2006.
- [219] ———, "Maximum likelihood estimation of transition probabilities of jump Markov linear systems," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5093–5108, 2008.
- [220] G. Wang, "ML estimation of transition probabilities in jump Markov systems via convex optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 3, pp. 1492–1502, 2010.
- [221] K. Gatsis and G. J. Pappas, "Sample Complexity of Networked Control Systems Over Unknown Channels," in *Proceedings of the 57th IEEE Conference on Decision and Control*, 2018, pp. 6067–6072.
- [222] ———, "Statistical learning for analysis of networked control systems over unknown channels," *Automatica*, vol. 125, p. 109386, 2021.
- [223] Y. Jiang, J. Fan, T. Chai, F. L. Lewis, and J. Li, "Tracking Control for Linear Discrete-Time Networked Control Systems With Unknown Dynamics and Dropout," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4607–4620, 2018.
- [224] J. Fan, Q. Wu, Y. Jiang, T. Chai, and F. L. Lewis, "Model-Free Optimal Output Regulation for Linear Discrete-Time Lossy Networked Control Systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4033–4042, 2020.
- [225] K. Gatsis and G. J. Pappas, "Learning to Control over Unknown Wireless Channels," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2600–2605, 2020, 21st IFAC World Congress.
- [226] M. Eisen, K. Gatsis, G. J. Pappas, and A. Ribeiro, "Learning in Non-Stationary Wireless Control Systems via Newton's Method," in *Proceedings of the 2018 Annual American Control Conference*, 2018, pp. 1410–1417.
- [227] ———, "Learning in Wireless Control Systems Over Nonstationary Channels," *IEEE Transactions on Signal Processing*, vol. 67, no. 5, pp. 1123–1137, 2019.

- [228] S. Schaal and C. G. Atkeson, "Learning Control in Robotics," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 20–29, 2010.
- [229] R. Sutton, *Reinforcement Learning : An Introduction*. Cambridge, Massachusetts London, England: The MIT Press, 2018.
- [230] T. Fujita and T. Ushio, "RL-based Optimal Networked Control Considering Network Delay of Discrete-Time Linear Systems," in *Proceedings of 2015 European Control Conference*, 2015, pp. 2476–2481.
- [231] Y. Bouteiller, S. Ramstedt, G. Beltrame, C. Pal, and J. Binas, "Reinforcement Learning with Random Delays," in *Proceedings of the 2021 International Conference on Learning Representations*, 2021.
- [232] J. P. Jansch-Porto, B. Hu, and G. Dullerud, "Policy Learning of MDPs with Mixed Continuous/Discrete Variables: A Case Study on Model-Free Control of Markovian Jump Systems," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, vol. 120, 2020, pp. 947–957.
- [233] O. L. Costa and J. C. Aya, "Monte Carlo TD( $\lambda$ )-methods for the optimal control of discrete-time Markovian jump linear systems," *Automatica*, vol. 38, no. 2, pp. 217–225, 2002.
- [234] R. L. Beirigo, M. G. Todorov, and A. M. S. Barreto, "Online TD( $\lambda$ ) for discrete-time Markov jump linear systems," in *Proceedings of the 57th IEEE Conference on Decision and Control*, 2018, pp. 2229–2234.
- [235] M. Schuurmans, P. Sopasakis, and P. Patrinos, "Safe Learning-Based Control of Stochastic Jump Linear Systems: a Distributionally Robust Approach," in *Proceedings of the 58th IEEE Conference on Decision and Control*, 2019, pp. 6498–6503.
- [236] K. Zhou and J. Doyle, *Essentials of Robust Control*. Prentice Hall, 1998.
- [237] I. Tzortzis, C. D. Charalambous, and C. N. Hadjicostis, "Jump LQR Systems With Unknown Transition Probabilities," *IEEE Transactions on Automatic Control*, vol. 66, no. 6, pp. 2693–2708, 2021.
- [238] L. Zhang, " $\mathcal{H}_\infty$  estimation for discrete-time piecewise homogeneous Markov jump linear systems," *Automatica*, vol. 45, no. 11, pp. 2570–2576, 2009.
- [239] S. B. Gibson and D. J. Stilwell, "Receding Horizon  $\mathcal{H}_\infty$  Control of Hidden Markov Jump Linear Systems," in *Proceedings of the 57th IEEE Conference on Decision and Control*, Dec 2018, pp. 5580–5585.
- [240] C. C. Lutz and D. J. Stilwell, "Stability and Disturbance Attenuation for Markov Jump Linear Systems with Time-Varying Transition Probabilities," *IEEE Transactions on Automatic Control*, vol. 61, no. 5, pp. 1413–1418, 2016.
- [241] A. R. C. Serafini, L. Delforno, J. M. Palma, F. H. Behrens, and C. F. Morais, "Robust Static Output-Feedback Control for MJLS with Non-Homogeneous Markov Chains: A Comparative Study Considering a Wireless Sensor Network with Time-Varying PER," *Sensors*, vol. 21, no. 19, 2021.
- [242] Y. Yin and Z. Lin, "Constrained control of uncertain nonhomogeneous Markovian jump systems," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 17, pp. 3937–3950, 2017.

- [243] Y. Z. Lun, A. D’Innocenzo, A. Abate, and M. D. Di Benedetto, “Optimal robust control and a separation principle for polytopic time-inhomogeneous Markov jump linear systems,” in *Proceedings of the 56th IEEE Conference on Decision and Control*, 2017, pp. 6525–6530.
- [244] Y. Z. Lun, A. D’Innocenzo, and M. D. Di Benedetto, “Robust Stability of Polytopic Time-Inhomogeneous Markov Jump Linear Systems,” *Automatica*, vol. 105, pp. 286–297, 2019.
- [245] S. Aberkane, “Stochastic Stabilization of a Class of Nonhomogeneous Markovian Jump Linear Systems,” *Systems & Control Letters*, vol. 60, no. 3, pp. 156–160, 2011.
- [246] O. Costa, E. Assumpção Filho, E. Boukas, and R. Marques, “Constrained quadratic state feedback control of discrete-time Markovian jump linear systems,” *Automatica*, vol. 35, no. 4, pp. 617–626, 1999.
- [247] C. F. Morais, M. F. Braga, R. C. L. F. Oliveira, and P. L. D. Peres, “Reduced-order dynamic output feedback control of uncertain discrete-time Markov jump linear systems,” *International Journal of Control*, vol. 90, no. 11, pp. 2368–2383, 2017.
- [248] L. Zhang and E.-K. Boukas, “Stability and Stabilization of Markovian Jump Linear Systems with Partly Unknown Transition Probabilities,” *Automatica*, vol. 45, no. 2, pp. 463–468, 2009.
- [249] C. F. Morais, M. F. Braga, R. C. Oliveira, and P. L. Peres, “ $\mathcal{H}_\infty$  state feedback control for MJLS with uncertain probabilities,” *Automatica*, vol. 52, pp. 317–321, 2015.
- [250] Y. Z. Lun, A. D’Innocenzo, and M. D. Di Benedetto, “On Stability of Time-Inhomogeneous Markov Jump Linear Systems,” in *Proceedings of the 55th IEEE Conference on Decision and Control*, Dec 2016, pp. 5527–5532.
- [251] ———, “Robust Stability of Time-Inhomogeneous Markov Jump Linear Systems,” in *Proceedings of the 20th IFAC World Congress*, 2017, pp. 3418–3423.
- [252] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge New York: Cambridge University Press, 2012.
- [253] G.-C. Rota and W. Strang, “A note on the joint spectral radius,” *Proceedings of the Netherlands Academy*, vol. 22, pp. 379–381, 1960.
- [254] R. Jungers, *The Joint Spectral Radius: Theory and Applications*, ser. Lecture Notes in Control and Information Sciences. Berlin: Springer, 2009.
- [255] M. A. Berger and Y. Wang, “Bounded Semigroups of Matrices,” *Linear Algebra and its Applications*, vol. 166, pp. 21–27, 1992.
- [256] S. Boyd, *Convex Optimization*. Cambridge, UK New York: Cambridge University Press, 2004.
- [257] G. Varsis, “The multidimensional content of the frustum of the simplex,” *Pacific Journal of Mathematics*, vol. 46, no. 1, pp. 303–314, 1973.
- [258] O. Costa and M. Fragoso, “Stability Results for Discrete-Time Linear Systems with Markovian Jumping Parameters,” *Journal of Mathematical Analysis and Applications*, vol. 179, no. 1, pp. 154 – 178, 1993.
- [259] J. G. VanAntwerp and R. D. Braatz, “A tutorial on linear and bilinear matrix inequalities,” *Journal of Process Control*, vol. 10, no. 4, pp. 363–385, 2000.
- [260] M. Wright, “The interior-point revolution in optimization: history, recent developments, and lasting consequences,” *Bulletin of the American mathematical society*, vol. 42, no. 1, pp. 39–56, 2005.

- [261] G. Vankeerberghen, J. Hendrickx, and R. M. Jungers, “JSR: A Toolbox to Compute the Joint Spectral Radius,” in *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '14. New York, NY, USA: ACM, 2014, pp. 151–156.
- [262] P. Patrinos, P. Sopasakis, H. Sarimveis, and A. Bemporad, “Stochastic model predictive control for constrained discrete-time Markovian switching systems,” *Automatica*, vol. 50, no. 10, pp. 2504–2514, 2014.
- [263] P. Sopasakis, D. Herceg, P. Patrinos, and A. Bemporad, “Stochastic economic model predictive control for Markovian switching systems,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 524–530, 2017, 20th IFAC World Congress.
- [264] I. Hwang, C. E. Seah, and S. Lee, “A study on stability of the interacting multiple model algorithm,” *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 901–906, 2016.
- [265] M. Dolgov, G. Kurz, and U. D. Hanebeck, “State estimation for Stochastic Hybrid Systems based on deterministic Dirac mixture approximation,” in *Proceedings of the 2014 American Control Conference*, 2014, pp. 1408–1413.
- [266] M. Maiworm, T. Bätthge, and R. Findeisen, “Scenario-based Model Predictive Control: Recursive Feasibility and Stability,” *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 50–56, 2015, 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.
- [267] M. Prandini, S. Garatti, and J. Lygeros, “A Randomized Approach to Stochastic Model Predictive Control,” in *Proceedings of the 51st IEEE Conference on Decision and Control*, 2012, pp. 7315–7320.
- [268] F. Kozin, “A Survey of Stability of Stochastic Systems,” *Automatica*, vol. 5, no. 1, pp. 95–112, 1969.
- [269] M. Schinkel, W.-H. Chen, and A. Rantzer, “Optimal control for systems with varying sampling rate,” in *Proceedings of the 2002 American Control Conference*, 2002, pp. 2979–2984.
- [270] H. Haimovich and E. N. Osella, “On controller-driven varying-sampling-rate stabilization via Lie-algebraic solvability,” *Nonlinear Analysis: Hybrid Systems*, vol. 7, no. 1, pp. 28–38, 2013.
- [271] S. Linsenmayer and F. Allgöwer, “Performance oriented triggering mechanisms with guaranteed traffic characterization for linear discrete-time systems,” in *Proceedings of the 2018 European Control Conference*, 2018, pp. 1474–1479.
- [272] S. Wildhagen, M. A. Müller, and F. Allgöwer, “Predictive Control Over a Dynamical Token Bucket Network,” *IEEE Control System Letters*, vol. 3, no. 4, pp. 859–864, 2019.
- [273] S. Wildhagen and F. Allgöwer, “Scheduling and control over networks using MPC with time-varying terminal ingredients,” in *Proceedings of the 2020 American Control Conference*, 2020, pp. 1913–1918.
- [274] J. Yu and S. J. Qin, “Statistical MIMO controller performance monitoring. Part I: Data-driven covariance benchmark,” *Journal of Process Control*, vol. 18, no. 3, pp. 277–296, 2008.
- [275] M. Bauer, A. Horch, L. Xie, M. Jelali, and N. Thornhill, “The current state of control loop performance monitoring – A survey of application in industry,” *Journal of Process Control*, vol. 38, pp. 1–10, 2016.
- [276] S. P. Chatzis and Y. Demiris, “Nonparametric Mixtures of Gaussian Processes With Power-Law Behavior,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 12, pp. 1862–1871, 2012.

- 
- [277] A. Naylor and G. Sell, *Linear Operator Theory in Engineering and Science*, ser. Applied Mathematical Sciences. Springer New York, 1982.
- [278] J. Brewer, "Kronecker Products and Matrix Calculus in System Theory," *IEEE Transactions on Circuits and Systems*, vol. 25, no. 9, pp. 772–781, 1978.
- [279] H. V. Henderson and S. R. Searle, "The vec-permutation matrix, the vec operator and Kronecker products: a review," *Linear and Multilinear Algebra*, vol. 9, no. 4, pp. 271–288, 1981.
- [280] S.-D. Wang, T.-S. Kuo, and C.-F. Hsu, "Trace bounds on the solution of the algebraic matrix Riccati and Lyapunov equation," *IEEE Transactions on Automatic Control*, vol. 31, no. 7, pp. 654–656, 1986.
- [281] J. Geromel, A. Gonçalves, and A. Fioravanti, "Dynamic Output Feedback Control of Discrete-Time Markov Jump Linear Systems through Linear Matrix Inequalities," *SIAM Journal on Control and Optimization*, vol. 48, no. 2, pp. 573–593, 2009.
- [282] N. J. Higham, "Analysis of the Cholesky decomposition of a semi-definite matrix," in *Reliable Numerical Computation*. Oxford University Press, 1990, pp. 161–185.
- [283] E. Kreindler and A. Jameson, "Conditions for Nonnegativeness of Partitioned Matrices," *IEEE Transactions on Automatic Control*, vol. 17, no. 1, pp. 147–148, 1972.
- [284] A. Albert, "Conditions for Positive and Nonnegative Definiteness in Terms of Pseudoinverses," *SIAM Journal on Applied Mathematics*, vol. 17, no. 2, pp. 434–440, 1969.
- [285] E. Behrends, *Introduction to Markov Chains*. Braunschweig/Wiesbaden: Vieweg, 2000.



## Own Publications

- [286] F. Rosenthal and U. D. Hanebeck, “Stability Analysis of Polytopic Markov Jump Linear Systems with Applications to Sequence-Based Control over Networks,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3104–3111, 2020, 21st IFAC World Congress.
- [287] ———, “Sequence-Based Stochastic Receding Horizon Control Using IMM Filtering and Value Function Approximation,” in *Proceedings of the 58th IEEE Conference on Decision and Control*, Nice, France, Dec. 2019.
- [288] F. Rosenthal, M. Dolgov, and U. D. Hanebeck, “Sequence-Based Receding Horizon Control Over Networks with Delays and Data Losses,” in *Proceedings of the 2019 American Control Conference*, Philadelphia, PA, USA, Jul. 2019.
- [289] F. Rosenthal and U. D. Hanebeck, “A Control Approach for Cooperative Sharing of Network Resources in Cyber-Physical Systems,” in *Proceedings of the 2019 IEEE International Conference on Industrial Cyber-Physical Systems*, Taipei, Republic of China, May 2019.
- [290] F. Rosenthal, M. Jung, M. Zitterbart, and U. D. Hanebeck, “CoCPN – Towards Flexible and Adaptive Cyber-Physical Systems Through Cooperation,” in *Proceedings of the 2019 IEEE Annual Consumer Communications & Networking Conference*, Las Vegas, USA, Jan. 2019.
- [291] S. Özgen, M. F. Huber, F. Rosenthal, J. Mayer, B. Noack, and U. D. Hanebeck, “Retrodiction of Data Association Probabilities via Convex Optimization,” in *Proceedings of the 21st International Conference on Information Fusion*, Cambridge, United Kingdom, Jul. 2018.
- [292] F. Rosenthal, B. Noack, and U. D. Hanebeck, “State Estimation in Networked Control Systems with Delayed and Lossy Acknowledgments,” in *Multisensor Fusion and Integration in the Wake of Big Data, Deep Learning and Cyber Physical System*, S. Lee, H. Ko, and S. Oh, Eds. Cham: Springer International Publishing, 2018, pp. 22–38.
- [293] M. Dolgov, G. Kurz, D. Grimm, F. Rosenthal, and U. D. Hanebeck, “Stochastic Optimal Control Using Local Sample-Based Value Function Approximation,” in *Proceedings of the 2018 American Control Conference*, Milwaukee, Wisconsin, USA, Jun. 2018.
- [294] F. Rosenthal, B. Noack, and U. D. Hanebeck, “Scheduling of Measurement Transmission in Networked Control Systems Subject to Communication Constraints,” in *Proceedings of the 2018 American Control Conference*, Milwaukee, Wisconsin, USA, Jun. 2018.
- [295] M. Jung, F. Rosenthal, and M. Zitterbart, “Poster Abstract: CoCPN-Sim: An Integrated Simulation Environment for Cyber-Physical Systems,” in *Proceedings of the IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation*, Orlando, FL, USA, Apr. 2018, Best Poster Runner-Up IoTDI 2018.
- [296] F. Rosenthal, B. Noack, and U. D. Hanebeck, “State Estimation in Networked Control Systems With Delayed And Lossy Acknowledgments,” in *Proceedings of the 2017 IEEE International*

*Conference on Multisensor Fusion and Integration for Intelligent Systems*, Daegu, Republic of Korea, Nov. 2017.

## (Co-) Supervised Student Theses

For privacy reasons, the names of the students are omitted in the following list.

- [297] “Ereignisgesteuerte Folgeregelung über verlustbehaftete Netzwerke mit Anwendung auf holonome Roboter,” Master’s thesis, Karlsruhe Institute of Technology (KIT), 2021.
- [298] “Omnimover Low Speed Control,” Master’s thesis, Karlsruhe Institute of Technology (KIT), 2019.
- [299] “Development of a Pose Controller and Dead Reckoning for a Novel Omnidirectional Wheeled Mobile Robot,” Master’s thesis, Karlsruhe Institute of Technology (KIT), 2019.
- [300] “Including Update Mechanisms for Network Models into State Estimation in Networked Control Systems,” Master’s thesis, Karlsruhe Institute of Technology (KIT), 2019.
- [301] “Modeling End-to-End Delay in Networked Control Systems via Dirichlet Process Mixture Model,” Bachelor’s thesis, Karlsruhe Institute of Technology (KIT), 2019.
- [302] “Entwicklung einer Applikation zur Steuerung einer omnidirektionalen Roboterplattform über mobile Geräte,” Bachelor’s thesis, Karlsruhe Institute of Technology (KIT), 2019.
- [303] “Koppelnavigation mit Inertialsensorik für eine mobile Roboterplattform,” Bachelor’s thesis, Karlsruhe Institute of Technology (KIT), 2019.