SIMON HELLING

Fortschrittsberichte aus der Regelungstechnik
und Prozessautomatisierung

Band 01

NONLINEAR AND OPTIMAL CONTROL, COLLISION AVOIDANCE, AND ESTIMATION OF MARINE SURFACE VESSELS

Karlsruher Institut für Technologie

Simon Helling

Nonlinear and Optimal Control, Collision Avoidance, and Estimation of Marine Surface Vessels

Fortschrittsberichte aus der Regelungstechnik
und Prozessautomatisierung
Band 01

Herausgeber: Prof. Dr.-Ing. habil. Thomas Meurer

Nonlinear and Optimal Control, Collision Avoidance, and Estimation of Marine Surface Vessels

by
Simon Helling

Karlsruher Institut für Technologie

Karlsruher Institut für Technologie
Institut für Mechanische Verfahrenstechnik und Mechanik

Nonlinear and Optimal Control, Collision Avoidance, and Estimation of Marine Surface Vessels

Zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften von der KIT-Fakultät für Chemieingenieurwesen und Verfahrenstechnik des Karlsruher Instituts für Technologie (KIT) genehmigte Dissertation

von Simon Helling

Tag der mündlichen Prüfung: 8. Juli 2024
Erstgutachter: Prof. Dr.-Ing. Thomas Meurer
Zweitgutachter: Prof. Dr.-Ing. Dirk Abel

Impressum

Karlsruher Institut für Technologie

Karlsruher Institut für Technologie (KIT)
Kaiserstraße 12
76131 Karlsruhe

Institut für Mechanische Verfahrenstechnik und Mechanik
www.mvm.kit.edu

2024

ISSN 2943-0887
DOI 10.5445/IR/1000174026

Abstract

In this work the modeling, optimal and nonlinear control of marine surface vessels (MSVs) is addressed along with collision avoidance, state and disturbance estimation. The basis of the control and estimation schemes is the underlying mathematical model. Therefore, a thorough derivation of the nonlinear dynamical model of MSVs is presented, which takes into account a combination of rigid body dynamics and hydrodynamic effects. The former is derived using Newton's and Euler's laws of motion and the latter using a Taylor series expansion of the hydrodynamic forces acting on the vessel hull. Furthermore, the effect of the propulsion is taken into account, which includes actuation forces caused by propellers and rudders. A nonlinear control design is presented that is based on feedback linearization, which transforms the nonlinear system into a decoupled linear system by a suitable change of coordinates and nonlinear state feedback control. The resulting linear system is then controlled using a linear pole placement controller. Based on the propulsion model, a control allocation is presented, which is used for overactuated vessels to distribute the desired control forces to the actuators and realizes secondary goals such as minimum wear and tear of the actuators. The control allocation problem constitutes a constrained static optimization problem that is solved using numerical methods. In this context, an interior-point sequential quadratic programming (SQP) line search method is proposed that is able to cope with nonlinear static constrained optimization problems and serves as an example for state-of-the-art methods of numerical optimization and their complexity. Additionally, a predictive controller is proposed, which is able to handle constraints on the control inputs and states. The latter are especially relevant in the context of the control of MSVs in confined areas, where collision avoidance is a major concern. Usually, obstacles are handled using potential fields, which are added to the cost function, or by imposing implicit inequality constraints to the model predictive control (MPC) problem. Most popularly, ellipsoidal constraints are used that overestimate the obstacle shapes and fail to include the controlled vessel's shape. To this end, dual collision avoidance constraints in combination with culling techniques are introduced that are able to represent convex vessel shapes efficiently. Moreover, the dual approach is able to include the controlled vessel's shape making it especially useful for applications in confined environments. Both control approaches rely on the full state information. Therefore, state estimation is addressed by means of an adaptive iterated Extended Kalman Filter (AIEKF). This filter is able to additionally estimate time-varying noise covariance matrices and disturbances. The nonlinear controller in combination with the control allocation scheme is used to illustrate a dynamic positioning problem, where a high fidelity simulation with two different disturbance observer and measurement models is compared. The optimization-based control scheme is illustrated by means of a predictive path following collision avoidance problem in the Kiel Bay area, where Automatic Identification System (AIS) data is used to simulate the vessel traffic.

Kurzfassung

In dieser Arbeit wird die Modellierung, optimale und nichtlineare Regelung von Schiffen behandelt. Darüber hinaus wird die Zustands- und Störgrößenschätzung behandelt. Grundlage der Regelungs- und Schätzverfahren ist das mathematische Schiffsmodell. Daher erfolgt eine ausführliche Herleitung des dynamischen Schiffsmodells, das eine Kombination aus Starrkörperdynamik und hydrodynamischen Effekten darstellt. Erstere wird mithilfe der Newtonschen und Eulerschen Bewegungsgesetze abgeleitet, während letztere mithilfe der Taylor-Reihenentwicklung modelliert werden. Darüber hinaus wird die Auswirkung des Antriebs analysiert, der die Effekte der Aktuatoren mit einbezieht. Das nichtlineare Regelungskonzept basiert auf einer exakten Linearisierung, die das System durch eine geeignete Koordinatentransformation und ein nichtlineares Stellgesetz in ein lineares System überführt und entkoppelt. Das resultierende lineare System wird dann mithilfe einer Polplatzierung stabilisiert. Basierend auf dem Antriebsmodell wird die *Control-Allocation* vorgestellt, die bei überaktuierten Schiffen verwendet wird, um die geforderten Kräfte auf die Aktoren zu verteilen und sekundäre Ziele wie z.B. minimalen Verschleiß zu realisieren. Die *Control-Allocation* stellt ein beschränktes statisches Optimierungsproblem dar, das mit numerischen Methoden gelöst wird. In diesem Zusammenhang wird ein *Interior-Point*-SQP-Liniensuchverfahren vorgestellt, das in der Lage ist, Probleme dieser Art zu lösen und als Beispiel für *State-of-the-Art*-Methoden der numerischen Optimierung und zur Veranschaulichung der Komplexität derer dient. Zusätzlich wird eine modellprädiktive Regelung (MPC) erläutert, die in der Lage ist, Eingangs- und Zustandschranken zu berücksichtigen. Letztere sind besonders relevant im Zusammenhang mit der Schiffsführung in beengten Seegebieten, bei der die Kollisionsvermeidung von entscheidender Bedeutung ist. Üblicherweise werden Hindernisse mithilfe von Potentialfeldern berücksichtigt oder, indem dem MPC-Problem Ungleichheitsbeschränkungen auferlegt werden. Am häufigsten werden dabei Ellipsoide verwendet, die die Hindernisformen überschätzen und die Form des geregelten Schiffs nicht berücksichtigen. Zu diesem Zweck wird ein duales Konzept in Verbindung mit *Culling*-Methoden vorgestellt, das konvexe Schiffsformen darstellen und einbeziehen kann. Darüber hinaus berücksichtigt der duale Ansatz die Form des geregelten Schiffs, was besonders in beengten Seegebieten nützlich ist. Beide Regelungskonzepte benötigen die vollständige Zustandsinformation. Daher wird die Zustandsschätzung mithilfe eines adaptiven iterierten erweiterten Kalman Filters gelöst. Dieses Filter ist in der Lage, zusätzlich zeitlich veränderliche Rauschkovarianzmatrizen und Störgrößen zu schätzen. Der nichtlineare Regler in Kombination mit der *Control-Allocation* wird verwendet, um ein dynamisches Positionierungsproblem zu veranschaulichen, bei dem eine *High-Fidelity*-Simulation mit zwei verschiedenen Störgrößenmodellen verglichen wird. Die optimierungsbasierte Regelung wird anhand eines prädiktiven Pfadfolgeproblems mit

Kollisionsvermeidung in der Kieler Förde veranschaulicht, das *Automatic Identification System (AIS)*-Daten zur Simulation des Schiffsverkehrs verwendet.

Preface

Diese Arbeit begann im Jahr 2019 am Lehrstuhl für Regelungstechnik an der Christian-Albrechts-Universität zu Kiel und fand ihr Ende im Jahr 2024 bei der Gruppe für Digital Process Engineering am Karlsruher Institut für Technologie (KIT).

Mein ausgesprochener Dank wird Herrn Prof. Dr.-Ing. Thomas Meurer zuteil, der es mir ermöglicht hat, fokussiert und frei zu forschen; der außerdem stets ein offenes Ohr für fachliche Diskussionen hat und anhaltend sein Engagement für mich und andere unter Beweis stellt.

Ich möchte mich außerdem bei meinen Lehrstuhlkollegen und Freunden Pascal Jerono, Max Lutz und Sönke Bartels bedanken für die regen Diskussionen, den moralischen Beistand, die stabilisierende Wirkung, die unzähligen lustigen Abende und für eine andauernde Freundschaft, die ich nicht missen möchte.

Außerdem möchte ich Fabian Heuer meinen Dank aussprechen, der mich in meinem Studium tagtäglich begleitet, motiviert, unterstützt und mir stets neue und interessante Perspektiven auf Probleme jeglicher Art im Studium und im Leben aufgezeigt hat.

Mein Dank gilt ausserdem Gerhard Freudenthaler, der mich vor meiner Zeit als Doktorand auf meinem Werdegang begleitet und sich immer die nötige Zeit für mich genommen und mich ein Stück weit zu dem Wissenschaftler gemacht hat, der ich jetzt bin.

Nicht zuletzt bedanke ich mich bei allen Freunden und Freundinnen, die sich angesprochen fühlen, wenn sie manchmal nicht umhin kamen, mit Regelungstechnik-behafteten Monologen belastet zu werden und die für mich da waren, wenn ich sie gebraucht habe. Habt Dank.

Please note that part of this work, especially in Chapter 7, has been published in the following papers:

- Helling et al. [175]
- Helling and Meurer [172]
- Helling and Meurer [168]

Furthermore note that Sec. A.4 in the appendix is not contained in the examination copy of this thesis but was added as a supplement for this published version. That particular section enhances the Fig. 6.2 and provides context from different disciplines with regard to the numerical solution of optimal control problems.

February 12, 2024, Kiel *Simon Helling*

”Learning does not make one learned: there are those who have knowledge and those who have understanding. The first requires memory, the second philosophy.”’

–Alexandre Dumas, *The Count of Monte Cristo*

”A little learning is a dangerous thing;
Drink deep, or taste not the Pierian spring:
There shallow draughts intoxicate the brain,
And drinking largely sobers us again.
Fired at first sight with what the Muse imparts,
In fearless youth we tempt the heights of arts,
While from the bounded level of our mind,
Short views we take, nor see the lengths behind,
But more advanced behold with strange surprise,
New distant scenes of endless science rise!”

–Alexander Pope, *An Essay on Criticism*

Table of Contents

Abstract i

Kurzfassung iii

Preface v

Abbreviations and Symbols xiii

I. Introduction 1

1. Motivation 3

2. State of the Art 5

2.1. Mathematical Modeling 5

2.2. Control 6

2.3. Obstacle Constraints 7

2.4. Navigation 9

3. Objectives 11

4. Outline 13

II. Theoretical Background 15

5. Mathematical Modeling of MSVs 17

5.1. Rigid Body Dynamics 18

5.1.1. Translational Motion – Newton's Second Law 18

5.1.2. Rotational Motion – Euler's Second Law 21

5.1.3. Matrix-Vector Representation 24

5.2. External Forces and Moments 25

5.2.1. Acceleration-Dependent Forces 25

5.2.2. Velocity-Dependent Forces 27

5.2.3. Wind Forces 28

5.2.4. Control Forces 30

5.3. Nonlinear Input-Affine Ordinary Differential Equation 33

5.4. Autopilot (Nomoto) Model 34

6. Nonlinear and Optimization-Based Control Design 37
6.1. Feedback Linearization 38
6.2. Control Allocation 42
6.2.1. Problem Formulation 43
6.2.2. Numerical Solution 45
6.3. Optimal and Model Predictive Control 56
6.3.1. Problem Formulation 56
6.3.2. Numerical Solution 58
6.3.3. Model Predictive Control 62

7. Collision Avoidance 63
7.1. Geometric Ship Model 64
7.2. Indicator Constraints 66
7.2.1. Singleton Case 66
7.2.2. Full Body Case 69
7.3. Distance Constraints 69
7.3.1. Singleton Case 71
7.3.2. Full Body Case 72
7.4. Signed Distance Constraints 74
7.4.1. Singleton Case 75
7.4.2. Full Body Case 76
7.5. Culling Procedure 77
7.5.1. View Frustum Culling 79
7.5.2. Occlusion Culling 79
7.5.3. Backface Culling 80

8. Kalman Filter-Based Estimation 83
8.1. Extended Kalman Filter 83
8.2. Iterated Extended Kalman Filter 84
8.3. Extended Kalman Smoother 86
8.4. Adaptive Noise Covariance Estimation 88

III. Simulative Studies 91

9. Dynamic Positioning 93
9.1. Problem Formulation 93
9.1.1. Vessel Model 94
9.1.2. Nonlinear Feedback Controller 95
9.1.3. Control Allocation 95
9.1.4. State and Disturbance Estimation 98

9.2. Results and Discussion . . . 104
9.2.1. Lumped Disturbance Model . . . 104
9.2.2. Combined Lumped and Wind Disturbance Model . . . 109

10. Predictive Path Following . . . 113
10.1. Problem Formulation . . . 113
10.1.1. Vessel Model . . . 114
10.1.2. Model Predictive Controller . . . 115
10.1.3. State Estimation . . . 119
10.2. Results and Discussion . . . 119

IV. Conclusion and Perspectives for Future Research 125

11. Conclusions . . . 127

12. Perspectives for Future Research . . . 131

A. Appendix . . . 133
A.1. Example Problems for the Numerical Solution of NLPs . . . 133
A.1.1. Static Optimization: Hanging Chain . . . 133
A.1.2. Dynamic Optimization: Inverted Pendulum on a Cart . . . 135
A.2. Proof of (7.11) using Farkas' Lemma . . . 139
A.3. Extended Kalman Filter (EKF) Jacobian Matrices for the Dynamic Positioning Simulation . . . 140
A.3.1. Lumped Disturbance Estimation Model . . . 140
A.3.2. Combined Lumped and Wind Disturbance Estimation Model . . . 140
A.4. On Numerical Methods for Solving Optimal Control Problems . . . 141

List of Figures . . . 143

List of Tables . . . 147

References . . . 149

Supervised Theses . . . 163

Own Publications . . . 165
Journal Publications . . . 165
Conference Publications . . . 165

Abbreviations and Symbols

Abbreviations

AIEKF	adaptive iterated Extended Kalman Filter
AIS	Automatic Identification System
BFGS	Broyden-Fletcher-Goldfarb-Shanno
CG	center of gravity
COLREG	International Regulations for Preventing Collisions at Sea
DOF	degree of freedom
EKF	Extended Kalman Filter
EPA	expanding polytopes algorithm
GNC	guidance, navigation and control
GJK	Gilbert-Johnson-Keerthi
IEKF	iterated Extended Kalman Filter
KF	Kalman Filter
KKT	Karush-Kuhn-Tucker
MPC	model predictive control
MSV	marine surface vessel
NED	North-East-Down
NLP	nonlinear program
OCP	optimal control problem
ODE	ordinary differential equation
PDE	partial differential equation
PID	proportial integral derivative
RTS	Rauch-Tung-Striebel

SOC	second order correction
SQP	sequential quadratic programming
TPBVP	two point boundary value problem
UKF	Unscented Kalman Filter

Sets, Norms

$\mathbb{N}$	Natural numbers $\mathbb{N} = \{1, 2, \dots\}$
$\mathbb{N}_0$	Natural numbers including zero $\mathbb{N}_0 = \{0, 1, 2, \dots\}$
$\mathbb{Z}$	Integer numbers $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$
$\mathbb{R}$	Real numbers
$\mathbb{R}^+$	Positive real numbers not including zero $\mathbb{R}^+ = \{a \in \mathbb{R} \mid a > 0\}$
$\mathbb{R}_0^+$	Positive real numbers including zero $\mathbb{R}_0^+ = \{a \in \mathbb{R} \mid a \geq 0\}$
$\mathbb{R}^-$	Negative real numbers not including zero $\mathbb{R}^- = \{a \in \mathbb{R} \mid a < 0\}$
$\mathbb{R}_0^-$	Negative real numbers including zero $\mathbb{R}_0^- = \{a \in \mathbb{R} \mid a \leq 0\}$

Operators and mathematical Symbols

a	Scalar
$\boldsymbol{a}$	Vector
$\boldsymbol{a}_k$	Vector sampled at time step $t_k = k\Delta t$
$\dot{\boldsymbol{a}}$	Time derivative of vector $\boldsymbol{a}$
$\hat{\boldsymbol{a}}$	Estimate of vector $\boldsymbol{a}$
$\hat{\boldsymbol{a}}^-$	*A priori* estimate
$\hat{\boldsymbol{a}}^+$	*A posteriori* estimate
A	Matrix

Latin Symbols and Variables

x	North position in the North-East-Down coordinate system
y	East position in the North-East-Down coordinate system
z	Down position in the North-East-Down coordinate system
u	Surge speed in body-fixed coordinate system
v	Sway speed in body-fixed coordinate system
w	Heave speed in body-fixed coordinate system
p	Roll (angular) speed in body-fixed coordinate system
q	Pitch (angular) speed in body-fixed coordinate system
r	Yaw (angular) speed in body-fixed coordinate system
τ_X	Surge force
τ_Y	Sway force
τ_Z	Heave force
τ_K	Roll moment
τ_M	Pitch moment
τ_N	Yaw moment
$\boldsymbol{x}$	State vector
$\boldsymbol{y}$	Measurement
$\boldsymbol{u}$	Manipulated variables
n_x	Number of states
n_y	Number of measurements
n_u	Number of control inputs
n_w	Number of decision variables
I	Identity matrix of appropriate size
$I^{(n\times m)}$	Identity matrix $I \in \mathbb{R}^{n\times m},\ n \in \mathbb{N}, m \in \mathbb{N}$
0	Zero matrix of appropriate size
$0^{(n\times m)}$	Zero matrix $0 \in \mathbb{R}^{n\times m},\ n \in \mathbb{N}, m \in \mathbb{N}$
M	Mass matrix
C	Coriolis matrix

D	Damping matrix
$\mathcal{D}$	Coupling matrix
P	State error covariance matrix
Q	Process noise spectral density matrix
R	Measurement noise covariance matrix
B_τ	(Vessel) Input matrix
B_T	Thrust configuration matrix

Greek Symbols and Variables

ρ	Air density
$\rho_{\mathbf{w}}$	Water density
ω	Rotational speed typically associated with a propeller
ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
$\boldsymbol{\eta}$	Pose of the vessel
$\boldsymbol{\nu}$	Body-fixed velocity
$\boldsymbol{\tau}_{\mathbf{c}}^{*}$	Commanded control forces
$\boldsymbol{\tau}_{\mathbf{c}}$	Realized control forces

General Subscripts

am	Added Mass
ekf	Extended Kalman Filter
fb	Feedback
mpc	Model Predictive Control
rb	Rigid Body
nl	Nonlinear

Part I.

Introduction

1. Motivation

The field of autonomous navigation and control of marine surface vessels (MSVs) receives a growing interest in the research community. This is mainly due to the fact that over 80 % of goods are transported on the seaway [63]. Thus, economic aspects drive the research and development of autonomous navigation and control techniques of marine craft to reduce the cost of transportation in freight shipping. The latter includes the cost of human operators on the ship and the cost of fuel. Both these cost factors can be mitigated by automation of the navigation and control of the ship. Safety requirements constitute another driving force for the development of autonomous navigation and control techniques as most accidents can be traced back to human error. These requirements include the ability to detect and avoid static and dynamic obstacles taking into account the International Maritime Organization measures of increasing safety of navigation at sea, in particular, the International Regulations for Preventing Collisions at Seas (COLREGs) [75]. Other safety-critical factors include the compensation of environmental disturbance forces and the ability to perform safe and efficient docking maneuvers in confined areas. The latter is especially relevant in the context of autonomous passenger ferry transportation. Autonomous systems that cope with these challenges enable safer vessel operation in many applications such as autonomous passenger ferry operation in remote and isolated areas [90], autonomous cargo transportation [146], and autonomous offshore operations, e.g., for wind park installation, oil rig maintenance, drilling, and pipe laying [145, 156].

From a theoretical point of view, the field of autonomous navigation and control of marine craft is a challenging task due to the wide variety of requirements ranging from robust and reliable control schemes that need to take into account physical limitations as well as environmental uncertainties to observer design that needs to be able accurately reconstruct states, parameters, and disturbances acting on the ship based on noisy and possibly unreliable sensor measurements. The magnitude of these individual requirements is linked to their respective application, which range from trajectory tracking [122] and path following [25] to obstacle avoidance [37] and dynamic positioning [145]. Most if not all of these approaches rely on the use of a mathematical model of the vessel, which includes the rigid body characteristics of the ship and the effects of the ship displacing water during maneuvering. Furthermore, a variety of different forces and moments acting on the ship needs to be considered, such as the hydrodynamic forces and moments effecting from the vessel's propulsion system, the environmental forces and moments resulting from wind, waves, and ocean currents, Coriolis effects, and viscous damping forces and moments. The complexity of the model is, again, linked to the application it is being used in. It ranges from simple transfer functions for autopilots [120] to nonlinear maneuvering models for dynamically positioned vessels [1, 52]. The complexity and variety of the individual aspects of autonomous control of MSVs motivate the need of a concise and comprehensive derivation of the mathematical modeling of MSVs and, furthermore, necessitate the need for a systematic and structured approach to the design and synthesis of a wide variety of state-of-the-art control and observer designs techniques.

2. State of the Art

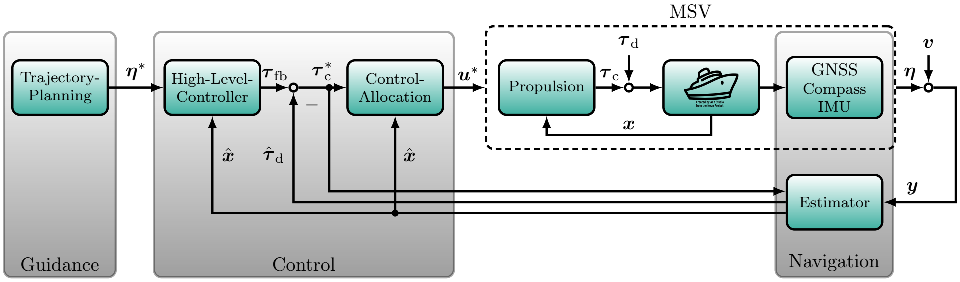


Figure 2.1.: Schematic control loop structure with GNC of an MSV (dashed) with reference pose $\boldsymbol{\eta}^*$, feedback, commanded, and realized control forces $\boldsymbol{\tau}_{\mathrm{fb}}$, $\boldsymbol{\tau}_{\mathrm{c}}^*$, and $\boldsymbol{\tau}_{\mathrm{c}}$, respectively, optimal control surfaces $\boldsymbol{u}^*$, measurements $\boldsymbol{y}$, measurement noise $\boldsymbol{v}$, real and estimated states $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, respectively, and real and estimated disturbances $\boldsymbol{\tau}_{\mathrm{d}}$ and $\hat{\boldsymbol{\tau}}_{\mathrm{d}}$, respectively.

In this chapter, an overview of the state of the art in the field of autonomous vessel control is provided w.r.t the different elements of the control loop that is usually categorized into guidance, navigation and control (GNC) as depicted in Fig. 2.1. To this end, focus is put on the navigation and control elements.

2.1. Mathematical Modeling

First attempts to obtain a mathematical model for the dynamic behavior of marine vessels date back to Davidson and Schiff [33], which has been extended by Nomoto et al. [120] and where sway-yaw interactions are modeled as a first-order system that is decoupled from the surge dynamics. These models are still used in many situations [4, 26] today and can be extended to include higher-order terms [47, Sec. 7.2.3]. A major development regarding the development of maneuvering models for marine craft constitutes the work of Abkowitz [1], who derived a nonlinear dynamical model based on the combination of rigid body dynamics and a Taylor series expansion of forces and moments acting on the vessel w.r.t. an operating point. The large number of resulting parameters in the model is greatly reduced by symmetry considerations of the vessel hull. Several variants arose based on this model, e.g., the so-called modulus model proposed in Fedayevsky and Sobolev [44], where some of the higher-order terms in Taylor series terms of the Abkowitz model are replaced with absolute value functions. A more recent development regarding the modeling of marine craft is the work of Fossen [48], who introduced the nonlinear maneuvering model in a structure resembling that of robotic manipulators. This latter model is very widely-used today. Modeling of marine craft can be distinguished between Newtonian, Lagrangian, and Kirchoff approaches, all of which are equivalent in the sense that they yield the same equations of motion [105, 27]. For a detailed overview of different modeling techniques of marine craft, see Sutulo and Guedes Soares [147].

2.2. Control

In general, the different control algorithms applied to MSVs can be classified into two categories, namely, classical and optimization-based approaches. Classical control approaches can be further distinguished into linear and nonlinear methods, where the former rely on a simplified, linear vessel model which include, e.g., the Nomoto model or linearized maneuvering models and methods from linear control theory can be utilized for controller design such as, e.g., Laplace transform, Bode plots, and pole placement [32]. The latter approaches make use of nonlinear maneuvering models and apply techniques of nonlinear control theory such as, e.g., feedback linearization, flatness-based feedforward control, passivity-based approaches, and backstepping [123, 174, 53, 49]. Optimization-based approaches utilize the theory of dynamic optimization and can handle both linear and nonlinear mathematical models of the vessel. Depending on the application, different control approaches are prevalent. Trajectory tracking controllers for MSVs can be designed using Lyapunov's direct method as shown, e.g., in Breivik and Fossen [22] or feedback linearization [122, 123] but recently increasingly rely on optimization-based approaches see, e.g., Kosch et al. [87], Kinjo et al. [83]. Path following controllers are handled very similarly to trajectory tracking controllers. However, in this context, proportial integral derivative (PID) controllers are also applied to solve the task see, e.g., [47, Sec. 10.3.2],[25, 112]. Furthermore, apart from classical feedback linearization approaches [123], an extension to the transverse normal form can be used to explicitly include the degrees of freedom (DOFs) resulting from the path following problem see, e.g., Banaszuk and Hauser [10], Nielsen and Maggiore [115], Nielsen et al. [116]. Apart from nonlinear control theory approaches, optimization-based methods can be applied to solve the path following problem [58, 39, 41, 42]. Moreover, hybrid approaches combine optimization-based methods with nonlinear approaches such as differential flatness as proposed in [17]. The task of dynamic positioning can be solved with PID controllers as proposed by [133]. The first optimization-based approach has been introduced by Balchen et al. [9] where, among other techniques, wave and wind filtering techniques were integrated into the disturbance estimator. Alternatively, nonlinear approaches such as feedback linearization [14], backstepping [49], and passivity [51] are also widely-used and allow for useful analytical insights into the control problem. Control allocation, which maps desired control forces and moments to the actor configuration on the vessel, can be categorized based on the assumptions of the underlying actor model, i.e., time-varying and time-invariant models. The former usually results in a dynamic optimization problem see, e.g., Brandner [21], whereas the latter results in a static optimization problem. This latter approach is much more commonly used and can be solved efficiently using a variety of methods as shown, e.g., in Johansen et al. [79], Bodson [18].

2.3. Obstacle Constraints

Most commonly, obstacles are represented using implicit functions and, more specifically, ellipsoidal representations [16, 175, 72]. See also Tab. 2.1 for an overview of obstacle representations and related methods. Alternatively, CSG functions combine geometric primitives (such as implicit quadric functions) with set operations [129], i.e., set union and set intersection and allow versatile obstacle shapes but are inherently difficult to produce and, therefore, are seldom used [174]. For convex polyhedra an extensive amount of research has been conducted and several very efficient algorithms have been developed that make extensive use of the convexity property. This is mainly due to the fact that convexity is very well-studied and understood [20]. Therefore, it is no surprise that the convex polyhedral representation is a very efficient and accurate obstacle representation technique that gains popularity in the context of collision avoidance in optimal control problems (OCPs) [175, 124, 160, 161]. Non-convex polyhedra and other non-convex shapes are usually decomposed into a set of convex polyhedra and, therefore, are handled with corresponding methods. Another way to deal with collision avoidance constraints in the context of polyhedral objects is to associate a binary variable with the polyhedral faces and formulate the collision avoidance as a mixed-integer linear problem [130]. However, the solution of these types of problems often relies on heuristics and, as the name suggests, is often restricted to linear costs and constraints [62]. These limitations restrict the applicability of the mixed-integer approach since the majority of autonomous vehicle control problems involves nonlinearities in one way or the other. More generally applicable are (non-integer) optimization-based techniques, where the notion of a distance function is often used to evaluate collision avoidance. Especially focused on robotics, this idea originated, e.g., in [59], where the distance constraint is included as part of an OCP to ensure collision avoidance. Therein, the distance is calculated in each iteration step by a subordinate algorithm. Not surprisingly, the authors of [59] also proposed the Gilbert-Johnson-Keerthi (GJK) algorithm, which can be used to calculate the distance between two objects efficiently [60]. This algorithm is more commonly known in the context of computer game physics and various improvements have been proposed to improve efficiency [111]. Since for many applications a measure of severity of collisions is of interest, the distance in the OCP (or physics engine) can be replaced with the signed distance, which also provides a measure of penetration depth in case of collision. To this end, the GJK algorithm is complemented by the expanding polytopes algorithm (EPA) that evaluates the penetration depth of overlapping obstacles [151]. This so-called primal approach of including the (signed) distance as part of the OCP can also be found in, e.g., [101, 136]. However, since the (signed) distance function is not continuously differentiable, problems may occur in the optimization algorithm using the primal method. Even though empirical

	Mathematical Representation	Associated Method(s)	Application(s) to collision avoidance
Non-polyhedral	Implicit functions [121]	Level Set Method, Fast Marching Method [139]	Quadric obstacle shapes [175, 16, 72]
	Parametric curves [91]	[135], B-splines, NURBS [125]	–
	Constructive Solid Geometry [129]	–	Versatile obstacle shapes [174]
Polyhedral	Convex [20]	Lagrange Duality [20, 175], GJK [60], EPA [151], Culling [57, 172]	Polyhedral obstacle shapes [175, 161, 124, 160]
	Non-convex	Convex decomposition [57]	–

Table 2.1.: Classification of obstacle representations, related methods and example applications for collision avoidance OCPs.

evidence suggests that this problem seems negligible see, e.g., [101, 136], various measures have been proposed to circumvent this issue such as in [134], where the (signed) distance constraint is approximated using a convex free region. Another popular approach is to reformulate the (signed) distance constraint by making use of the Lagrange dual problem of the (signed) distance, which is itself an optimization problem, see [20]. This leads to the dual method that provides an exact, continuously differentiable re-formulation of the primal (signed) distance OCP but comes at the cost of additional dual decision variables see, e.g. [124] for its (first) application to aircraft trajectory generation. Recently, this approach has been adapted and improved for other applications see, e.g., [45, 175, 144, 160], where the focus lies on multi-robot coordination, autonomous vessel operations, robust MPC, and autonomous parking, respectively. A further improvement to the dual method is demonstrated in [57], where a culling procedure is proposed that minimizes the number of dual decision variables as part of a proprietary active-set SQP algorithm. In [172], this concept is adapted to perform culling *a priori* to solving the OCP in order to allow the use of a variety of general purpose solvers. Along with the use of GJK and EPA in the primal approach, this technique demonstrates a further connection of optimization-based autonomous vehicle control to computer graphics principles since it is a fundamental concept in computer graphics rendering pipelines see, e.g., [36, 74]. In [45], a semi-dual (or distributed) approach is introduced, where the (signed) distance is not included to the OCP using dual variables, but is rather calculated as a subordinate optimization problem (similar to the primal approach) obtained using the Lagrange dual problem of the (signed) distance definition (different from primal approach). For a more general review of the dual method and focus on generating a feasible initial guess for the dual variables see [172, 161].

2.4. Navigation

At the core of the navigation system that collects sensor information is the navigation filter or observer that takes care of the estimation of the vessel's position and orientation in the global reference frame along with its translational and angular velocities, and possibly also disturbances and system parameters. De-facto standard for this task is the Kalman Filter (KF) in one of its variants [80], especially the EKF [140, 55] and the Unscented Kalman Filter (UKF) [34, 35, 118]. However, especially in the context of disturbance estimation, nonlinear observers become more prevalent [6, 7, 51, 96].

3. Objectives

This thesis aims to provide a concise overview of several aspects of the GNC loop depicted in Fig. 2.1. This work aims to analyze, explain, combine, and extend several state-of-the-art methods for all of the depicted components of the control loop with special focus on the mathematical model, predictive high-level controller design, obstacle modeling, collision avoidance, EKF-based state and disturbance estimation, and control allocation. The basis for most of the components of the control loop is the underlying mathematical vessel model. Therefore, a thorough derivation of the mathematical vessel model is given and, based on this, two widely-used high-level controller architectures are introduced that are able to cover two of the most important applications for marine craft, namely, path following and dynamic positioning. The control architectures consist of an MPC controller and a nonlinear controller based on feedback linearization that can be utilized in combination with a control allocation algorithm. The former constitutes a special type of OCP that needs to be solved numerically on a receding horizon, which results in an MPC controller, which is one of the most widely-used control schemes next to PID controllers. In this context, a typical numerical method to solve the MPC problem is discussed that is especially suited for collision avoidance applications by making use of a so-called culling procedure. The latter controller makes use of methods related to more classical approaches of nonlinear control design and a short but concise overview of these types of methods is given. In principle, the control allocation can be applied to both control schemes but is especially suited for the combination with the nonlinear controller in that it distributes the commanded generalized control forces $\boldsymbol{\tau}_{\mathrm{c}}^{*}$ to the actuators $\boldsymbol{u}^{*}$ taking into account actor constraints. In the context of control allocation, secondary goals apart from realizing the commanded control force can be introduced such as, e.g., minimum wear of the actuators, making it especially relevant for practical implementations. The solution to the control allocation problem relies on a numerical solution to a static constrained optimization problem. To this end, a novel interior-point SQP line search algorithm is proposed that captures the essence of these types of solution approaches. A fundamental aspect of the control loop is the estimation of the states and disturbances to close the control loop. In this context, the EKF is discussed and several extensions are introduced that elevate the EKF to a higher-order filter and make it suitable for the estimation of the process and/or measurement noise covariance, and parameter estimation.

Whenever feasible, this thesis aims to provide theoretical backgrounds of the above-mentioned approaches that are in itself contained. All in all, the key contributions of this thesis are summarized as follows:

- Concise derivation of the mathematical vessel model including the forces acting on it.
- A novel interior-point SQP line search algorithm suitable for, e.g., the solution of control allocation and OCPs.
- Derivation of a flat parameterization of fully-actuated surface vessels.
- Dual collision avoidance constraints for MPC problems.
- Size reduction of the underlying collision avoidance MPC problem with culling techniques.
- Combination of adaptive covariance estimation with an AIEKF.
- Wind disturbance and measurement model for state and disturbance estimation.

The proposed methods are illustrated by means of two high-fidelity simulations. The first illustrates the MPC control using a simple observer model with dual collision avoidance and culling techniques. The second simulation illustrates the control allocation using a nonlinear controller and an AIEKF with adaptive covariance estimation with two different observer models, which are used to simultaneously estimate the system states and disturbances acting on the system.

4. Outline

This thesis is separated into three parts. The second part of the thesis is the main part and is concerned with theoretical foundations. To this end, that particular part is separated into four chapters.

The first chapter of the first part gives a detailed derivation of the mathematical modeling of MSVs that is the foundation of most of the following concepts and ideas. Therefore, the mathematical model is derived using rigid body dynamics and, subsequently, introduces all relevant hydrodynamic forces and moments acting on the hull of the vessel. These may be separated into acceleration- and velocity-dependent forces and moments as well as environmental and controlled forces. The resulting mathematical model is then compactly represented as a first order ordinary differential equation (ODE) in input-affine form.

The second chapter discusses two controller designs. First, it introduces and recalls classical methods of nonlinear control theory, namely, feedback linearization and differential flatness, which are usually applied to MSVs in the context of dynamic positioning. This section then applies the aforementioned concepts to the general structure of the MSV model to obtain further insight in the model. In this context, optimization-based control allocation is discussed, which is especially suited to be used in combination with the nonlinear control design, since it solves the problem of distributing a given, desired generalized input force to the effectors and actuators of the vessel in an optimal way. This resulting static optimization problem is taken as a motivation to introduce fundamental concepts of constrained, static optimization and the numerical solution thereof. Therefore, a basic interior-point SQP line search method is introduced as an example of the complexity of finding an optimal solution to nonlinear optimization problems. Furthermore, this chapter reviews the method of optimal control that enables to compensate for the disadvantages of methods from classical nonlinear control theory by taking into account input and state constraints. Therefore, the fundamental concepts of constrained, dynamic optimization problems are discussed, and numerical solution techniques are analyzed, which, in general, rely on the solution of a large-scale nonlinear program (NLP). Furthermore, the proposed OCP is extended to a nonlinear MPC scheme that is able to deal with time-varying constraints and disturbances and, essentially, realizes a closed-loop control by solving the OCP on a finite, receding horizon.

The third chapter focuses on the obstacle modeling and implementation in confined environments in the context of MPC. To this end, a dual formulation is introduced that takes into account polyhedral obstacle shapes as well as the geometry of the controlled vessel. Several variations of these types of constraints are derived depending on whether the controlled vessel is assumed to be a point or its full geometry is considered. Moreover, a culling procedure is proposed to achieve an efficient solution to the resulting MPC problem.

The fourth chapter deals with the EKF-based state and disturbance (parameter) estimation. Several extensions are reviewed and combined to maximize the EKF performance, namely, an iterative measurement update realizes higher-order filter performance, and adaptive covariance estimation is proposed to compensate for possible missing knowledge w.r.t. the process and/or measurement noise covariance matrices.

The third part of the thesis is concerned with the application of the proposed theoretical concepts to the problem of predictive path-following and dynamic positioning of MSVs by means of high-fidelity simulations. The first chapter in this part deals with the dynamic positioning of a MSV and utilizes a nonlinear feedback linearization to achieve this task in conjunction with a control allocation and AIEKF-based state and disturbance estimation. Subsequently, the second chapter illustrates a model predictive controller that realizes a path-following problem in a confined environment using real AIS data and, therefore, also realizes collision avoidance by means of the dual collision avoidance constraints and culling techniques.

Finally, conclusions are summarized and directions for future work are outlined in the fourth part. The appendix provides example solutions of a static and dynamic optimization problem and compares the performance of the proposed interior-point SQP line search method to a state-of-the-art solver. Furthermore, analytical Jacobians for the proposed observer models are provided.

Part II.

Theoretical Background

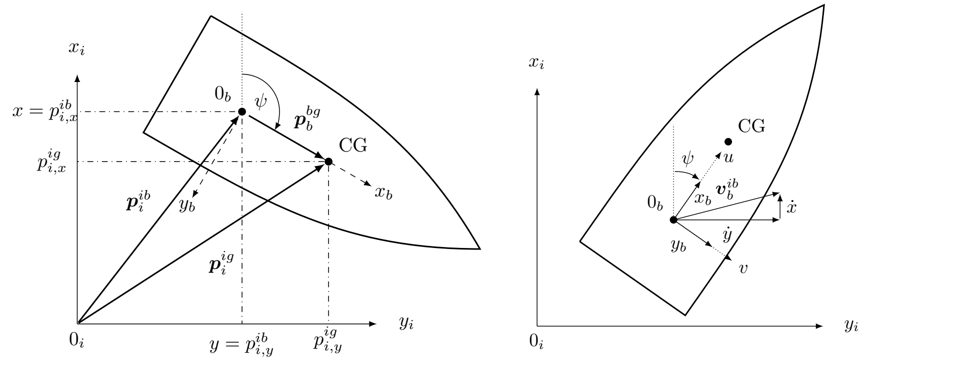
5. Mathematical Modeling of MSVs

In this chapter the three DOF mathematical model of MSVs is derived, which is modeled as a set of ODEs. These equations of motion can be separated into a kinematic part and a kinetic part [47]. The kinematic part relates the ship's velocities described in an inertial frame, namely, the North-East-Down (NED) frame, and a body-fixed frame. The kinetic part describes the dynamics of the vessel based on forces and moments acting on the hull. Here, focus is put on the derivation of the equations of motion using classical mechanics. This particular approach involves the description of translational motion using Newton's Second Law and the description of rotational motion using Euler's Second Law. Alternatively, the equations of motion can be derived using Kirchhoff's equations [84], which constitute a special form of the Lagrange formalism, see [132]. However, all three methods are equivalent see, e.g., [27]. In the following, a thorough derivation of the equations of motion is presented that first considers the inertial frame as the reference frame for the ship's motion and, subsequently, expresses the resulting equations in the body-fixed frame, which facilitates the formulation of the ship's actuation forces and moments.

Notation

In this chapter, a specific notation similar to Hahn [64] is used to clarify quantities expressed in different frames and to distinguish different reference points.

More specifically, position vectors are expressed using the notation $\boldsymbol{p}_{\text{expressed in}}^{\text{from to}}$, e.g., a vector pointing from the body-fixed origin to the center of gravity (CG) expressed in the inertial frame is denoted using $\boldsymbol{p}_i^{bg}$. Furthermore, angular velocity is expressed with $\boldsymbol{\omega}_{\text{expressed in}}^{\text{of}}$, and forces with $\boldsymbol{f}_{\text{expressed in}}^{\text{acting on}}$, e.g., a force acting on a vessel's CG expressed in the inertial frame is denoted using $\boldsymbol{f}_i^g$. For (turning) moments, the notation $\boldsymbol{m}_{\text{expressed in}}^{\text{with respect to}}$ is used, e.g., a moment with respect to a vessel's body-fixed origin expressed in the inertial frame is expressed using $\boldsymbol{m}_i^b$. Unless otherwise noted, a time derivative is always assumed to be w.r.t. the corresponding frame where the underlying quantity is expressed in, e.g., $\dot{\boldsymbol{a}}_i$, is the time derivative of $\boldsymbol{a}_i$ (expressed in the inertial frame) w.r.t. the inertial frame. Therefore, the more explicit but cumbersome notation ${}^i\dot{\boldsymbol{a}}_i$ is avoided when possible. Note that, accordingly, ${}^b\dot{\boldsymbol{a}}_i$ is the time derivative of the vector $\boldsymbol{a}_i$ expressed in the inertial frame w.r.t. the body-fixed frame and, in general, ${}^i\dot{\boldsymbol{a}}_i = {}^b\dot{\boldsymbol{a}}_i + \boldsymbol{\omega}_i^b \times \boldsymbol{a}_i = R_i^b({}^b\boldsymbol{a}_b + \boldsymbol{\omega}_b^b \times \boldsymbol{a}_b)$, and ${}^i\boldsymbol{a}_b = {}^b\boldsymbol{a}_b + \boldsymbol{\omega}_b^b \times \boldsymbol{a}_b$, which is abbreviated to $\dot{\boldsymbol{a}}_i = R_i^b(\dot{\boldsymbol{a}}_b + \boldsymbol{\omega}_b^b \times \boldsymbol{a}_b)$



(a) Geometric (kinematic) relations of the body-fixed frame $(0_b x_b y_b z_b)$, the inertial frame $(0_i x_i y_i z_i)$, the heading angle ψ, and the CG.

(b) Surge speed u, sway speed v defined in the body-fixed frame and $\dot{x}$ and $\dot{y}$ defined in the inertial frame such that $\boldsymbol{v}^{ib}_i = R^b_i \boldsymbol{v}^{ib}_b$

Figure 5.1.: Schematic relations of the vessel's body-fixed frame $(0_b x_b y_b z_b)$ to the inertial frame $(0_i x_i y_i z_i)$ in the 2D plane, i.e., for $z = \dot{z} = 0$, and with $y_g = z_g = 0$, i.e., the CG lies midships along the x_b-axis.

5.1. Rigid Body Dynamics

In this section the kinematic relations between the NED- and the body-frame are derived in 6DOF. Furthermore, Newton's Second Law is applied to obtain the translational rigid body dynamics of the MSV about the center of origin w.r.t. the NED frame. Analogously, Euler's Second Law is utilized to obtain the rotational rigid body dynamics. The resulting differential equations are then expressed in the body-fixed coordinates and velocities. Finally, the equations of motion are reduced to 3DOF and expressed in a matrix-vector form, which originates in Fossen et al. [47].

5.1.1. Translational Motion – Newton's Second Law

The equations of motion for MSVs typically rely on a body-fixed coordinate system $(0_b x_b y_b z_b)$ and an inertial coordinate system $(0_i x_i y_i z_i)$, i.e., the NED coordinate system, see Fig. 5.1. The MSV can be viewed as a rigid body and, therefore, the translational motion can be derived based on Newton's second law

$$\frac{\mathrm{d}}{\mathrm{d}t} m \boldsymbol{v}^{ig}_i = \boldsymbol{f}^g_i \tag{5.1}$$

with velocity $\boldsymbol{v}^{ig}_i = \frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{p}^{ig}_i$ and vector $\boldsymbol{p}^{ig}_i$ to the CG in inertial frame coordinates, i.e., relative to 0_i and $\boldsymbol{f}^g_i$ is the resultant external force acting on ship's CG expressed in the inertial frame. In view of Fig. 5.1, $\boldsymbol{p}^{ig}_i$ is given by

$$\boldsymbol{p}^{ig}_i = \boldsymbol{p}^{ib}_i + R^b_i \boldsymbol{p}^{bg}_b, \tag{5.2}$$

where

$$\boldsymbol{p}_i^{ib} = [p_{i,x}^{ib},\, p_{i,y}^{ib},\, p_{i,z}^{ib}]^\top := [x,\, y,\, z]^\top \tag{5.3}$$

is defined as the position of the MSV in inertial coordinates and for 3DOF models,

$$R_i^b = R_{z,\psi} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.4}$$

is the elemental rotation matrix about the z_b- or yaw-axis and $R_i^b = \{R_i^b \in \mathbb{R}^{3\times 3} \mid R_i^b R_b^i = I,\ \det R_i^b = 1\}$. From this it follows that $(R_i^b)^{-1} = (R_i^b)^\top = R_b^i$. Note that (5.2) is often abbreviated using

$$\boldsymbol{p}_i^{ig} = \boldsymbol{p}_i^{ib} + \boldsymbol{p}_i^{bg}, \tag{5.5}$$

where the time dependence of $\boldsymbol{p}_i^{bg}(t)$ is explicitly stated when expressed in the inertial frame. Note that, $\boldsymbol{p}_i^{bg} = R_i^b \boldsymbol{p}_b^{bg}$, where $\boldsymbol{p}_b^{bg} = [x_g,\, y_g,\, z_g]^\top$ is expressed in the body-fixed frame and, therefore, is not time-varying[1]. Differentiating (5.2) w.r.t. time results in

$$\boldsymbol{v}_i^{ig} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{p}_i^{ig} = \boldsymbol{v}_i^{ib} + \dot{R}_i^b \boldsymbol{p}_b^{bg} + R_i^b \boldsymbol{v}_b^{bg} = \boldsymbol{v}_i^{ib} + \dot{R}_i^b \boldsymbol{p}_b^{bg} \tag{5.6}$$

since $\boldsymbol{v}_b^{bg} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{p}_b^{bg} = \boldsymbol{0}$ for a rigid body. For the time derivative of R_i^b it is important to note that $R_i^b R_b^i = I$ and, therefore $\frac{\mathrm{d}}{\mathrm{d}t} I = \frac{\mathrm{d}}{\mathrm{d}t} R_i^b R_b^i = \dot{R}_i^b R_b^i + R_i^b \dot{R}_b^i = 0$ and, thus,

$$S(\boldsymbol{\omega}_i^b) = \dot{R}_i^b R_b^i = -S^\top(\boldsymbol{\omega}_i^b) = \begin{bmatrix} 0 & -\omega_{i,z}^b & \omega_{i,y}^b \\ \omega_{i,x}^b & 0 & -\omega_{i,x}^b \\ -\omega_{i,x}^b & \omega_{i,x}^b & 0 \end{bmatrix} \tag{5.7}$$

is a skew-symmetric matrix and $\boldsymbol{\omega}_i^b = [\omega_{i,x}^b,\, \omega_{i,y}^b,\, \omega_{i,z}^b]^\top$ is the vector of angular velocities. Additionally, to being skew-symmetric $S(\boldsymbol{a})$ has the properties $S(R_i^b \boldsymbol{a})\boldsymbol{b} = R_i^b S(\boldsymbol{a}) R_b^i \boldsymbol{b}$, and $S(\boldsymbol{a})\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$. Finally, $\dot{R}_i^b = \dot{R}_i^b I = \dot{R}_i^b (R_b^i R_i^b) = (\dot{R}_i^b R_b^i) R_i^b = S(\boldsymbol{\omega}_i^b) R_i^b$ and

$$\boldsymbol{v}_i^{ig} = \boldsymbol{v}_i^{ib} + S(\boldsymbol{\omega}_i^b) R_i^b \boldsymbol{p}_b^{bg} = \boldsymbol{v}_i^{ib} + \boldsymbol{\omega}_i^b \times \boldsymbol{p}_i^{bg}. \tag{5.8}$$

Inserting (5.8) in (5.1) yields

$$\frac{\mathrm{d}}{\mathrm{d}t} m[\boldsymbol{v}_i^{ib} + \boldsymbol{\omega}_i^b \times \boldsymbol{p}_i^{bg}] = \boldsymbol{f}_i^g \tag{5.9}$$

$$\tag{5.10}$$

and hence

$$m[\dot{\boldsymbol{v}}_i^{ib} + \dot{\boldsymbol{\omega}}_i^b \times \boldsymbol{p}_i^{bg} + \boldsymbol{\omega}_i^b \times (\boldsymbol{\omega}_i^b \times \boldsymbol{p}_i^{bg})] = \boldsymbol{f}_i^g. \tag{5.11}$$

[1] This explicit statement of time dependence is abandoned henceforth for the sake of simplicity unless emphasis is put on the time dependence.

Translational Motion Expressed with Body-Fixed Quantities

Note that all forces and velocities in (5.11) are expressed in the inertial frame. In the context of MSVs, it is more common to express the forces in the body-fixed frame where

$$\boldsymbol{f}_i^g = R_i^b \boldsymbol{f}_b^g, \tag{5.12}$$

and $\boldsymbol{f}_b^g = [\tau_X,\ \tau_Y,\ \tau_Z]^\top$ is the vector of external forces acting on the vessel expressed in the body-fixed coordinate frame. Furthermore, the velocity components in the inertial frame $\boldsymbol{v}_i^{ib} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{p}_i^{ib} = [\dot{x},\ \dot{y},\ \dot{z}]^\top$ are related to the velocity components in body-fixed frame with $u = \cos(\psi)\dot{x} + \sin(\psi)\dot{y}$ (surge speed), $v = -\sin(\psi)\dot{x} + \cos(\psi)\dot{y}$ (sway speed), and $w = \dot{z}$ (heave speed) in x_b-, y_b-, and z_b-direction, respectively, see Fig. 5.1b. Combining these body-fixed speeds yields $\boldsymbol{v}_b^{ib} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{p}_b^{ib} = [u,\ v,\ w]^\top$ and the relation to the inertial frame is expressed with $\boldsymbol{v}_b^{ib} = R_b^i \boldsymbol{v}_i^{ib}$ and, *vice versa*,

$$\boldsymbol{v}_i^{ib} = R_i^b \boldsymbol{v}_b^{ib}. \tag{5.13}$$

Time differentiation w.r.t. the inertial frame yields [64, Sec. 2.1.2.1]

$$\begin{aligned} {}^i\dot{\boldsymbol{v}}_i^{ib} &= R_i^b\,{}^b\dot{\boldsymbol{v}}_b^{ib} + S(\boldsymbol{\omega}_i^b)R_i^b\boldsymbol{v}_b^{ib} && (5.14\text{a}) \\ &= R_i^b({}^b\dot{\boldsymbol{v}}_b^{ib} + \boldsymbol{\omega}_b^b \times \boldsymbol{v}_b^{ib}) && (5.14\text{b}) \\ &= R_i^b\,{}^i\dot{\boldsymbol{v}}_b^{ib}, && (5.14\text{c}) \end{aligned}$$

where ${}^b\dot{\boldsymbol{v}}_b^{ib} = [\dot{u},\ \dot{w},\ \dot{w}]^\top$, i.e., the time derivative of the body-fixed velocity vector w.r.t. the body-fixed frame. Note that the frame distinction of the time derivative becomes necessary in this case as compared to (5.6), since, here, ${}^b\dot{\boldsymbol{v}}_b^{ib} \neq \boldsymbol{0}$ but ${}^b\boldsymbol{v}_b^{bg} = \boldsymbol{0}$ in (5.6). Analogously, the angular velocities in the respective frames relate with

$$\boldsymbol{\omega}_i^b = R_i^b \boldsymbol{\omega}_b^b, \tag{5.15}$$

where $\boldsymbol{\omega}_b^b = [p,\ q,\ r]^\top$ is expressed in the body-fixed frame. Differentiating w.r.t. time yields

$$\begin{aligned} {}^i\dot{\boldsymbol{\omega}}_i^b &= R_i^b\,{}^b\dot{\boldsymbol{\omega}}_b^b + S(\boldsymbol{\omega}_i^b)R_i^b\boldsymbol{\omega}_b^b && (5.16\text{a}) \\ &= R_i^b({}^b\dot{\boldsymbol{\omega}}_b^b + \boldsymbol{\omega}_b^b \times \boldsymbol{\omega}_b^b) && (5.16\text{b}) \\ &= R_i^b\,{}^b\dot{\boldsymbol{\omega}}_b^b, && (5.16\text{c}) \end{aligned}$$

since $\boldsymbol{\omega}_b^b \times \boldsymbol{\omega}_b^b = \boldsymbol{0}$, i.e., the angular acceleration is the same regardless of the frame. Finally, with (5.12), (5.14b), and (5.16c) Newton's second law (5.11) for the MSV expressed with body-fixed velocities and forces become

$$m[R_i^b({}^b\dot{\boldsymbol{v}}_b^{ib} + \boldsymbol{\omega}_b^b \times \boldsymbol{v}_b^{ib}) + R_i^b\dot{\boldsymbol{\omega}}_b^b \times \boldsymbol{p}_i^{bg} + \boldsymbol{\omega}_i^b \times (\boldsymbol{\omega}_i^b \times \boldsymbol{p}_i^{bg}))] = R_i^b \boldsymbol{f}_b^g, \tag{5.17}$$

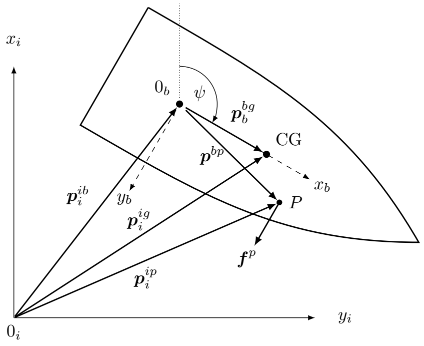


Figure 5.2.: A force $\boldsymbol{f}^p$ acting at point P induces a moment $\boldsymbol{m}_i^i = \boldsymbol{p}_i^{ip} \times \boldsymbol{f}_i^p = \boldsymbol{p}_i^{ib} \times \boldsymbol{f}_i^p + \boldsymbol{m}_i^b = \boldsymbol{p}_i^{ib} \times \boldsymbol{f}_i^p + \boldsymbol{p}_i^{bp} \times \boldsymbol{f}_i^p$ w.r.t. the inertial frame. Furthermore, it effects a translational motion of the MSV according to (5.11).

and finally

$$m[{}^b\dot{\boldsymbol{v}}_b^{ib} + \boldsymbol{\omega}_b^b \times \boldsymbol{v}_b^{ib} + \dot{\boldsymbol{\omega}}_b^b \times \boldsymbol{p}_b^{bg} + \boldsymbol{\omega}_b^b \times (\boldsymbol{\omega}_b^b \times \boldsymbol{p}_b^{bg})] = \boldsymbol{f}_b^g, \tag{5.18}$$

which is obtained by multiplying (5.17) with R_b^i from the left. Assuming that $y_g = z_g = 0$, (5.18) is expressed in component form as

$$m[\dot{u} - rv + qw - x_g(q^2 + r^2)] = \tau_X, \tag{5.19a}$$

$$m[\dot{v} - pw + ru + x_g(qp + \dot{r})] = \tau_Y, \tag{5.19b}$$

$$m[\dot{w} - qu + pv + x_g(pr - \dot{q})] = \tau_Z. \tag{5.19c}$$

5.1.2. Rotational Motion – Euler's Second Law

Any force $\boldsymbol{f}_i^p$ acting on the MSV at a point $\boldsymbol{p}_i^{ip}$ induces a moment $\boldsymbol{m}_i^i$ on the rigid body, see Fig. 5.2. The resulting rotation is described using Euler's second law

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{L}_i = \boldsymbol{m}_i^i, \tag{5.20}$$

where $\boldsymbol{m}_i^i$ is the resultant external moment (torque) acting on the body w.r.t. the inertial frame, and

$$\boldsymbol{L}_i = \int_V \boldsymbol{p}_i^{ip} \times \boldsymbol{v}_i^{ip} \rho \mathrm{d}V \tag{5.21}$$

is the total angular momentum w.r.t. to the inertial frame. The latter is expanded using the geometric relations observed in Fig. 5.2, i.e.,

$$\boldsymbol{L}_i = \int_V (\boldsymbol{p}_i^{ib} + \boldsymbol{p}_i^{bp}) \times (\boldsymbol{v}_i^{ib} + \boldsymbol{v}_i^{bp})\rho \mathrm{d}V, \tag{5.22a}$$

$$\begin{aligned} &= \int_V \boldsymbol{p}_i^{ib} \times \boldsymbol{v}_i^{ib} \rho \mathrm{d}V + \int_V \boldsymbol{p}_i^{ib} \times \boldsymbol{v}_i^{bp} \rho \mathrm{d}V \\ &\quad + \int_V \boldsymbol{p}_i^{bp} \times \boldsymbol{v}_i^{ib} \rho \mathrm{d}V + \int_V \boldsymbol{p}_i^{bp} \times \boldsymbol{v}_i^{bp} \rho \mathrm{d}V. \end{aligned} \tag{5.22b}$$

The four individual summands require further attention:

1. The first summand evaluates to

$$\int_V \boldsymbol{p}_i^{ib} \times \boldsymbol{v}_i^{ib} \rho \mathrm{d}V = \int_V \rho \mathrm{d}V \boldsymbol{p}_i^{ib} \times \boldsymbol{v}_i^{ib} = m\boldsymbol{p}_i^{ib} \times \boldsymbol{v}_i^{ib}. \tag{5.23}$$

2. The second summand is simplified using

$$\int_V \boldsymbol{p}_i^{ib} \times \boldsymbol{v}_i^{bp} \rho \mathrm{d}V = \boldsymbol{p}_i^{ib} \times \int_V \boldsymbol{v}_i^{bp} \rho \mathrm{d}V, \tag{5.24a}$$

$$= \boldsymbol{p}_i^{ib} \times \int_V \boldsymbol{\omega}_i^{b} \times \boldsymbol{p}_i^{bp} \rho \mathrm{d}V, \tag{5.24b}$$

$$= \boldsymbol{p}_i^{ib} \times \left(\boldsymbol{\omega}_i^{b} \times \int_V \boldsymbol{p}_i^{bp} \rho \mathrm{dV}\right), \tag{5.24c}$$

$$= m\boldsymbol{p}_i^{ib} \times (\boldsymbol{\omega}_i^{b} \times \boldsymbol{p}_i^{bg}). \tag{5.24d}$$

3. The third summand simplifies to

$$\int_V \boldsymbol{p}_i^{bp} \times \boldsymbol{v}_i^{ib} \rho \mathrm{d}V = \int_V \boldsymbol{p}_i^{bp} \rho \mathrm{d}V \times \boldsymbol{v}_i^{ib}, \tag{5.25a}$$

$$= m\boldsymbol{p}_i^{bg} \times \boldsymbol{v}_i^{ib}. \tag{5.25b}$$

4. Finally, the fourth summand yields

$$\int_V \boldsymbol{p}_i^{bp} \times \boldsymbol{v}_i^{bp} \mathrm{d}V = \int_V \boldsymbol{p}_i^{bp} \times R_i^b S(\boldsymbol{\omega}_b^b)\boldsymbol{p}_b^{bp} \rho \mathrm{d}V, \tag{5.26a}$$

$$= R_i^b \int_V \boldsymbol{p}_b^{bp} \times (\boldsymbol{\omega}_b^b \times \boldsymbol{p}_b^{bp}) \rho \mathrm{d}V, \tag{5.26b}$$

$$= R_i^b \int_V ((\boldsymbol{p}_b^{bp})^\top \boldsymbol{p}_b^{bp})\boldsymbol{\omega}_b^b - ((\boldsymbol{p}_b^{bp})^\top \boldsymbol{\omega}_b^b)\boldsymbol{p}_b^{bp} \rho \mathrm{d}V, \tag{5.26c}$$

$$= R_i^b \int_V \begin{bmatrix} (\xi^2 + \upsilon^2 + \zeta^2)p \\ (\xi^2 + \upsilon^2 + \zeta^2)q \\ (\xi^2 + \upsilon^2 + \zeta^2)r \end{bmatrix} - \begin{bmatrix} (\xi p + \upsilon q + \zeta r)x \\ (\xi p + \upsilon q + \zeta r)y \\ (\xi p + \upsilon q + \zeta r)z \end{bmatrix} \rho \mathrm{d}V, \tag{5.26d}$$

$$= R_i^b J_b \boldsymbol{\omega}_b^b, \tag{5.26e}$$

$$= J_i \boldsymbol{\omega}_i^b, \tag{5.26f}$$

where

$$J_b = \int_V \begin{bmatrix} (\upsilon^2+\zeta^2) & -\xi\upsilon & -\xi\zeta \\ -\xi\upsilon & (\xi^2+\zeta^2) & -\upsilon\zeta \\ -\xi\zeta & -\upsilon\zeta & (\xi^2+\upsilon^2) \end{bmatrix} \rho \mathrm{d}V \tag{5.27}$$

is the mass moment of inertia tensor w.r.t. the body-fixed frame, and $J_i = R_i^b J_b R_b^i$ is the (time dependent) mass moment of inertia w.r.t. the inertial frame.

Finally, inserting (5.23), (5.24d), (5.25b), and (5.26f) in (5.22) the total angular momentum reads

$$\boldsymbol{L}_i = m[\boldsymbol{p}_i^{ib} \times \boldsymbol{v}_i^{ib} + \boldsymbol{p}_i^{ib} \times (\boldsymbol{\omega}_i^b \times \boldsymbol{p}_i^{bg}) + \boldsymbol{p}_i^{bg} \times \boldsymbol{v}_i^{ib}] + J_i \boldsymbol{\omega}_i^b, \tag{5.28a}$$

$$= m[\boldsymbol{p}_i^{ib} \times (\boldsymbol{v}_i^{ib} + \boldsymbol{\omega}_i^b \times \boldsymbol{p}_i^{bg}) + \boldsymbol{p}_i^{bg} \times \boldsymbol{v}_i^{ib}] + J_i \boldsymbol{\omega}_i^b, \tag{5.28b}$$

$$= m[\boldsymbol{p}_i^{ib} \times \boldsymbol{v}_i^{ig} + \boldsymbol{p}_i^{bg} \times \boldsymbol{v}_i^{ib}] + J_i \boldsymbol{\omega}_i^b. \tag{5.28c}$$

Differentiating (5.28c) w.r.t. time yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{L}_i = m[\boldsymbol{v}_i^{ib} \times \boldsymbol{v}_i^{ig} + \boldsymbol{p}_i^{ib} \times \dot{\boldsymbol{v}}_i^{ig} + \boldsymbol{v}_i^{bg} \times \boldsymbol{v}_i^{ib} + \boldsymbol{p}_i^{bg} \times \dot{\boldsymbol{v}}_i^{ib}] + \boldsymbol{\omega}_i^b \times J_i \boldsymbol{\omega}_i^b + J_i \dot{\boldsymbol{\omega}}_i^b, \tag{5.29a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{L}_i = m[(\boldsymbol{v}_i^{bg} - \boldsymbol{v}_i^{ig}) \times \boldsymbol{v}_i^{ib} + \boldsymbol{p}_i^{ib} \times \dot{\boldsymbol{v}}_i^{ig} + \boldsymbol{p}_i^{bg} \times \dot{\boldsymbol{v}}_i^{ib}] + \boldsymbol{\omega}_i^b \times J_i \boldsymbol{\omega}_i^b + J_i \dot{\boldsymbol{\omega}}_i^b, \tag{5.29b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{L}_i = m[\boldsymbol{p}_i^{ib} \times \dot{\boldsymbol{v}}_i^{ig} + \boldsymbol{p}_i^{bg} \times \dot{\boldsymbol{v}}_i^{ib}] + \boldsymbol{\omega}_i^b \times J_i \boldsymbol{\omega}_i^b + J_i \dot{\boldsymbol{\omega}}_i^b, \tag{5.29c}$$

since $(\boldsymbol{v}_i^{bg} - \boldsymbol{v}_i^{ig}) \times \boldsymbol{v}_i^{ib} = -\boldsymbol{v}_i^{ib} \times \boldsymbol{v}_i^{ib} = \boldsymbol{0}$. Inserting (5.29c) into (5.20),

$$m[\boldsymbol{p}_i^{ib} \times \dot{\boldsymbol{v}}_i^{ig} + \boldsymbol{p}_i^{bg} \times \dot{\boldsymbol{v}}_i^{ib}] + \boldsymbol{\omega}_i^b \times J_i \boldsymbol{\omega}_i^b + J_i \dot{\boldsymbol{\omega}}_i^b = \boldsymbol{m}_i^i \tag{5.30}$$

is obtained. Furthermore, taking into account that $\boldsymbol{m}_i^i = \boldsymbol{p}_i^{ib} \times \boldsymbol{f}_i^g + \boldsymbol{m}_i^b = \boldsymbol{p}_i^{ib} \times m\dot{\boldsymbol{v}}_i^{ig} + \boldsymbol{m}_i^b$ (5.30) can be simplified as

$$J_i \dot{\boldsymbol{\omega}}_i^b + \boldsymbol{\omega}_i^b \times J_i \boldsymbol{\omega}_i^b + m\boldsymbol{p}_i^{bg} \times \dot{\boldsymbol{v}}_i^{ib} = \boldsymbol{m}_i^b. \tag{5.31}$$

Rotational Motion Expressed with Body-Fixed Quantities

Multiplying with R_b^i from the left

$$J_b \dot{\boldsymbol{\omega}}_b^b + \boldsymbol{\omega}_b^b \times J_b \boldsymbol{\omega}_b^b + m\boldsymbol{p}_b^{bg} \times {}^i\dot{\boldsymbol{v}}_b^{ib} = \boldsymbol{m}_b^b, \tag{5.32}$$

is obtained and, finally, in view of (5.14),

$$J_b \dot{\boldsymbol{\omega}}_b^b + \boldsymbol{\omega}_b^b \times J_b \boldsymbol{\omega}_b^b + m\boldsymbol{p}_b^{bg} \times ({}^b\dot{\boldsymbol{v}}_b^{ib} + \boldsymbol{\omega}_b^b \times \boldsymbol{v}_b^{ib}) = \boldsymbol{m}_b^b, \tag{5.33}$$

where $\boldsymbol{m}_b^b = [\tau_K, \tau_M, \tau_N]^\top$. Assuming that $y_g = z_g = J_{xy} = J_{xz} = J_{yz} = 0$, i.e., the CG is somewhere along the x_b-axis and the body-fixed origin coincides with a main axis of inertia, (5.33) expressed in component form reads

$$J_{xx}\dot{p} - (J_{yy} - J_{zz})qr = \tau_K, \tag{5.34}$$

$$J_{yy}\dot{q} - mx_g\dot{w} + (J_{xx} - J_{zz})pr + mx_g(qu - pv) = \tau_M, \tag{5.35}$$

$$J_{zz}\dot{r} + mx_g\dot{v} - (J_{xx} - J_{yy})pq + mx_g(ru - pw) = \tau_N. \tag{5.36}$$

5.1.3. Matrix-Vector Representation

The kinetic relations (5.18), and (5.33) are expressed in matrix-vector notation by rearranging as follows

$$m\dot{\boldsymbol{v}}_b^{ib} - mS(\boldsymbol{p}_b^{bg})\dot{\boldsymbol{\omega}}_b^b - mS(\boldsymbol{v}_b^{ib})\boldsymbol{\omega}_b^b - mS(\boldsymbol{\omega}_b^b)S(\boldsymbol{p}_b^{bg})\boldsymbol{\omega}_b^b = \boldsymbol{f}_b^g, \tag{5.37a}$$

$$mS(\boldsymbol{p}_b^{bg})\dot{\boldsymbol{v}}_b^{ib} + J_b\dot{\boldsymbol{\omega}}_b^b + mS(\boldsymbol{p}_b^{bg})S(\boldsymbol{\omega}_b^b)\boldsymbol{v}_b^{ib} - S(J_b\boldsymbol{\omega}_b^b)\boldsymbol{\omega}_b^b = \boldsymbol{m}_b^b. \tag{5.37b}$$

from which the representation

$$M_{\text{rb}}\begin{bmatrix}\dot{\boldsymbol{v}}_b^{ib}\\ \dot{\boldsymbol{\omega}}_b^b\end{bmatrix} = -C_{\text{rb}}(\boldsymbol{v}_b^{ib}, \boldsymbol{\omega}_b^b)\begin{bmatrix}\boldsymbol{v}_b^{ib}\\ \boldsymbol{\omega}_b^b\end{bmatrix} + \begin{bmatrix}\boldsymbol{f}_b^g\\ \boldsymbol{m}_b^b\end{bmatrix} \tag{5.38}$$

is derived. Therein,

$$M_{\text{rb}} = \begin{bmatrix} mI & -mS(\boldsymbol{p}_b^{bg})\\ mS(\boldsymbol{p}_b^{bg}) & J_b\end{bmatrix}, \tag{5.39}$$

is the mass matrix, and

$$C_{\text{rb}}(\boldsymbol{v}_b^{ib}, \boldsymbol{\omega}_b^b) = \begin{bmatrix} 0 & -mS(\boldsymbol{v}_b^{ib}) - mS(\boldsymbol{\omega}_b^b)S(\boldsymbol{p}_b^{bg})\\ -mS(\boldsymbol{v}_b^{ib}) + mS(\boldsymbol{p}_b^{bg})S(\boldsymbol{\omega}_b^b) & -S(J_b\boldsymbol{\omega}_b^b)\end{bmatrix}, \tag{5.40}$$

is the Coriolis matrix with $C_{\text{rb}}(\boldsymbol{v}_b^{ib}, \boldsymbol{\omega}_b^b) = -C_{\text{rb}}^\top(\boldsymbol{v}_b^{ib}, \boldsymbol{\omega}_b^b)$, i.e., it is skew-symmetric[2].

In the following, focus is put exclusively on the 3DOF case, i.e., when $p = q = w = 0$ or, in other words, the pitch, roll, and heave motions are assumed to be negligible, which also signifies $R_i^b = R_{z,\psi}$. To this end, the pose of the ship $\boldsymbol{\eta} = [x, y, \psi]^\top$, the body-fixed velocities $\boldsymbol{\nu} = [u, v, r]^\top$, and the vector of external forces and moment $\boldsymbol{\tau} = [\tau_X, \tau_Y, \tau_N]^\top$ are considered. Furthermore, assuming that the body-fixed origin coincides

[2] Note that the term $-mS(\boldsymbol{v}_b^{ib})$ does not appear in (5.37b) and needs to be added to retain skew-symmetry of $C_{\text{rb}}(\boldsymbol{v}_b^{ib}, \boldsymbol{\omega}_b^b)$, which is only possible since $S(\boldsymbol{v}_b^{ib})\boldsymbol{v}_b^{ib} = \boldsymbol{0}$.

with a principal axis of inertia, $J_{xy} = J_{xz} = J_{yz} = 0$, the 3DOF equations of motion in matrix-vector form become

$$\dot{\boldsymbol{\eta}} = R_{z,\psi}\boldsymbol{\nu}, \tag{5.41a}$$

$$M_{\mathrm{rb}}\dot{\boldsymbol{\nu}} = -C_{\mathrm{rb}}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{\tau}, \tag{5.41b}$$

where

$$M_{\mathrm{rb}} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & J_{zz} \end{bmatrix}, \quad C_{\mathrm{rb}}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -m(v + x_g r) \\ 0 & 0 & mu \\ m(v + x_g r) & -mu & 0 \end{bmatrix} \tag{5.42}$$

are the 3DOF mass and Coriolis matrices, respectively, and $C_{\mathrm{rb}}(\boldsymbol{\nu}) = -C_{\mathrm{rb}}^{\top}(\boldsymbol{\nu})$.

5.2. External Forces and Moments

In this section, the vector of external forces and moments $\boldsymbol{\tau}$ is further investigated. Apart from the MSV behaving as a rigid body it is evident that a vessel moves through water that needs to be displaced. This, among others, induces hydrodynamic forces to the vessel, which can be split into acceleration-dependent terms (so-called added mass), and into velocity-dependent terms, which represent damping effects such that $\boldsymbol{\tau}_{\mathrm{hy}} = \boldsymbol{\tau}_{\mathrm{a}}(\dot{\boldsymbol{\nu}}) + \boldsymbol{\tau}_{\mathrm{v}}(\boldsymbol{\nu})$ is the vector of hydrodynamic forces and moments. Furthermore, the vessel is subject to environmental forces such as wind, waves, and ocean currents, which act as forces on the vessel hull. Here, emphasis is put on wind effects $\boldsymbol{\tau}_{\mathrm{w}}$. Lastly, the vessel can be controlled by means of propellers, rudders, tunnel thrusters, and other actuators, which exert a force $\boldsymbol{\tau}_{\mathrm{c}}$. All in all, the sum of all external forces yields

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathrm{a}} + \boldsymbol{\tau}_{\mathrm{v}} + \boldsymbol{\tau}_{\mathrm{w}} + \boldsymbol{\tau}_{\mathrm{c}}. \tag{5.43}$$

5.2.1. Acceleration-Dependent Forces

The acceleration-dependent forces $\boldsymbol{\tau}_{\mathrm{a}}(\dot{\boldsymbol{\nu}})$ can be approximated using a first-order Taylor polynomial around $\dot{\boldsymbol{\nu}} = \dot{\boldsymbol{\nu}}_0 = [0,\ 0,\ 0]^{\top}$, i.e.,

$$\boldsymbol{\tau}_{\mathrm{a}}(\dot{\boldsymbol{\nu}}) \approx \boldsymbol{\tau}(\mathbf{0}) + \frac{\partial \boldsymbol{\tau}}{\partial \dot{\boldsymbol{\nu}}}(\mathbf{0})\dot{\boldsymbol{\nu}} = M_{\mathrm{am}}\dot{\boldsymbol{\nu}}, \tag{5.44}$$

where

$$M_{\mathrm{am}} = \begin{bmatrix} \frac{\partial X}{\partial \dot{u}} & \frac{\partial X}{\partial \dot{v}} & \frac{\partial X}{\partial \dot{r}} \\ \frac{\partial Y}{\partial \dot{u}} & \frac{\partial Y}{\partial \dot{v}} & \frac{\partial Y}{\partial \dot{r}} \\ \frac{\partial N}{\partial \dot{u}} & \frac{\partial N}{\partial \dot{v}} & \frac{\partial N}{\partial \dot{r}} \end{bmatrix} = \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{r}} \end{bmatrix}, \tag{5.45}$$

	Coefficient	Typical Value		Coefficient	Typical Value
M_{am}	$X_{\dot{u}} \in \mathbb{R}^-$	$X_{\dot{u}} \in [-0.05m, -0.1m]$	D_{l}	$X_u \in \mathbb{R}_0^-$	
	$Y_{\dot{v}} \in \mathbb{R}^-$	$-m$		$Y_v \in \mathbb{R}_0^-$	
	$Y_{\dot{r}} \in \mathbb{R}$	$\lvert Y_{\dot{r}} - mx_g \rvert \ll 1$		$Y_r \in \mathbb{R}$	$\lvert Y_r \rvert \ll mu_0$
	$N_{\dot{v}} \in \mathbb{R}$	$\lvert N_{\dot{v}} - mx_g \rvert \ll 1$		$N_v \in \mathbb{R}$	
	$N_{\dot{r}} \in \mathbb{R}^-$	$N_{\dot{r}} - J_{zz} \approx -1.8J_{zz}$		$N_r \in \mathbb{R}_0^-$	$N_r - mx_g u_0 \ll 0$

Table 5.1.: Coefficient domains and typical relations w.r.t. fundamental vessel parameters m, x_g, J_{zz}, i.e., with constant forward surge speed u_0 according to [1, Chap. 4].

is the added mass matrix, and the notation $X_{\dot{u}} = \frac{\partial X}{\partial \dot{u}}$ is used [94, Sec. 3.4], which indicates, e.g., how the surge force changes if an acceleration in the x_b-axis is present. The name added mass arises due to the unit of mass/inertia of the hydrodynamic derivatives in the added mass matrix. The added mass matrix can be further simplified because these derivatives are evaluated at $\dot{\boldsymbol{\nu}} = \mathbf{0}$ and, for port-starboard symmetric vessels, it can be deduced that $X_{\dot{v}}$ must be a symmetric (even) function $X(\dot{v}) = X(-\dot{v})$ such that $X_{\dot{v}} = 0$. The same holds true for $X(\dot{r})$ such that $X_{\dot{r}} = 0$, and for $Y_{\dot{u}}$, $N_{\dot{u}}$. On the other hand, it is argued in [94, Sec. 8.1] that $Y(\dot{v})$, $Y(\dot{r})$, $N(\dot{v})$, $N(\dot{r})$ must be odd functions such that $Y_{\dot{v}} \neq 0$, $Y_{\dot{r}} \neq 0$, $N_{\dot{v}} \neq 0$, $N_{\dot{r}} \neq 0$, i.e.,

$$M_{\mathrm{am}} = \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{r}} \\ 0 & N_{\dot{v}} & N_{\dot{r}} \end{bmatrix}, \tag{5.46}$$

and, furthermore it is often assumed that $Y_{\dot{r}} = N_{\dot{v}}$. The individual coefficients can be determined using

$$M_{ij} = -\rho_w \oint_S \varphi_i \frac{\partial \varphi_j}{\partial \boldsymbol{n}} \mathrm{d}S, \tag{5.47}$$

for $i = 1, 2, 3,\ j = 1, 2, 3$, and where ρ_w is the water density, S is the whetted surface area, $\boldsymbol{n}$ is the outward normal vector of the surface area, and φ_i is the i-th velocity potential of the surrounding fluid that needs to fulfill the Laplace equation $\nabla^2 \varphi_i = 0$. This expression is, in general, hard to evaluate, and, therefore, approximations are needed. These are usually one of the following. The method of an equivalent ellipsoid [86, Sec. 3.1] approximates the entire vessel as an elongated ellipsoid, where an analytical solution to (5.47) is available [138]. Alternatively, the strip theory method [86, Sec. 3.5.1] separates the vessel into thin transversal strips, such that each coefficient in the added mass matrix can be approximated by an integral along the length of the hull that depends on the geometry of the respective strip, which in turn is expressed using the draft and width of the ship. Lastly, experimental identification of the added mass matrix coefficients can be applied. Typical values of the added mass coefficients are given in Tab. 5.1.

5.2.2. Velocity-Dependent Forces

The velocity-dependent forces $\boldsymbol{\tau}_{\mathrm{d}}(\boldsymbol{\nu})$ can be approximated using a third-order Taylor polynomial that is denoted using $T_k \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{x}_0)$ around $\boldsymbol{\nu} = \boldsymbol{\nu}_0 = [0,\ 0,\ 0]^\top$, which is reduced in complexity by taking into account that $X(v)$ and $X(r)$ must be even functions, and $Y(v), Y(r), N(v), N(r)$ must be odd functions in their respective arguments, see also [94, Sec. 8.1] [1, Chap. 10], [149, Sec. 3.2], [147, Sec. 3.3.1], which eliminates several terms. With this

$$\boldsymbol{\tau}_{\mathrm{v}}(\boldsymbol{\nu}) \approx \sum_{|\alpha| \leq 3} \frac{D^\alpha \boldsymbol{\tau}_{\mathrm{v}}(\boldsymbol{\nu})}{\alpha!} (\boldsymbol{\nu} - \boldsymbol{\nu}_0)^\alpha = \begin{bmatrix} \tau_{\mathrm{v},X}(\boldsymbol{\nu}) \\ \tau_{\mathrm{v},Y}(\boldsymbol{\nu}) \\ \tau_{\mathrm{v},N}(\boldsymbol{\nu}) \end{bmatrix}, \tag{5.48}$$

where

$$\begin{aligned} \tau_{\mathrm{v},X}(\boldsymbol{\nu}) = {} & X_0 + X_u u + X_{uu} u^2 + X_{vv} v^2 + X_{rr} r^2 + X_{vr} vr \\ & + X_{uuu} u^3 + X_{uvv} uv^2 + X_{uvr} uvr + X_{urr} ur^2, \qquad (5.49a) \\ \tau_{\mathrm{v},Y}(\boldsymbol{\nu}) = {} & Y_v v + Y_r r + Y_{uv} uv + Y_{ur} ur + Y_{vr} vr \\ & + Y_{vvv} v^3 + Y_{vvr} v^2 r + Y_{vrr} vr^2 + Y_{rrr} r^3, \qquad (5.49b) \\ \tau_{\mathrm{v},N}(\boldsymbol{\nu}) = {} & N_v v + N_r r + N_{uv} uv + N_{ur} ur + N_{vr} vr \\ & + N_{vvr} v^2 r + N_{vrr} vr^2 + N_{rrr} r^3. \qquad (5.49c) \end{aligned}$$

Note that this generalization of a Taylor polynomial for vector-valued functions makes use of multi-index notation, i.e., $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{n_x})$, $|\alpha| = \alpha_1 + \alpha_2 + \ldots + \alpha_{n_x}$, $D^\alpha = {}^{\partial^{\alpha_1}}\!/\!{}_{\partial x_1^{\alpha_1}}\ {}^{\partial^{\alpha_1}}\!/\!{}_{\partial x_2^{\alpha_2}} \ldots {}^{\partial^{\alpha_2}}\!/\!{}_{\partial x_{n_x}^{\alpha_{n_x}}}$, $\boldsymbol{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_{n_x}^{\alpha_{n_x}}$, and $\alpha! = \alpha_1! \alpha_2! \ldots \alpha_{n_x}!$. Furthermore, according to [47, Sec. 7.1.2] and [131] the following substitutions can be made $X_{vr} = -Y_{\dot{v}},\ X_{rr} = -Y_{\dot{r}},\ Y_{ur} = X_{\dot{u}},\ N_{ur} = Y_{\dot{r}},\ N_{uv} = -(X_{\dot{u}} - Y_{\dot{v}})$ such that (5.49) can be simplified as

$$\begin{aligned} \tau_{\mathrm{v},X}(\boldsymbol{\nu}) = {} & X_u u + X_{uu} u^2 + X_{vv} v^2 - Y_{\dot{r}} r^2 - Y_{\dot{v}} vr \\ & + X_{uuu} u^3 + X_{uvv} uv^2 + X_{uvr} uvr + X_{urr} ur^2, \qquad (5.50a) \\ \tau_{\mathrm{v},Y}(\boldsymbol{\nu}) = {} & Y_v v + Y_r r + Y_{uv} uv + X_{\dot{u}} ur + Y_{vr} vr \\ & + Y_{vvv} v^3 + Y_{vvr} v^2 r + Y_{vrr} vr^2 + Y_{rrr} r^3, \qquad (5.50b) \\ \tau_{\mathrm{v},N}(\boldsymbol{\nu}) = {} & N_v v + N_r r - (X_{\dot{u}} - Y_{\dot{v}}) uv - Y_{\dot{r}} ur + N_{vr} vr \\ & + N_{vvr} v^2 r + N_{vrr} vr^2 + N_{rrr} r^3 \qquad (5.50c) \end{aligned}$$

for $X_0 = 0$, which can then be further simplified by means of a matrix-vector representation, i.e., $\boldsymbol{\tau}_{\mathrm{v}}(\boldsymbol{\nu}) = D_{\mathrm{l}}\boldsymbol{\nu} + D_{\mathrm{nl}}(\boldsymbol{\nu})\boldsymbol{\nu} + C_{\mathrm{am}}(\boldsymbol{\nu})\boldsymbol{\nu}$, where

$$D_{\mathrm{l}} = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix}, \tag{5.51a}$$

$$D_{\mathrm{nl}}(\boldsymbol{\nu}) = \begin{bmatrix} X_{uu}u + X_{uuu}u^2 & 0 & 0 \\ 0 & Y_{vvv}v^2 + Y_{vvr}vr & Y_{rrr}r^2 + Y_{vrr}vr \\ 0 & N_{vvv}v^2 + N_{vvr}vr & N_{rrr}r^2 + N_{vrr}vr \end{bmatrix} \tag{5.51b}$$

denote the linear and nonlinear damping matrix, respectively, for $X_{vv} = X_{uvv} = X_{urr} = X_{uvr} = 0$. Typical values of the linear damping matrix can be seen in Tab. 5.1. Furthermore,

$$C_{\mathrm{am}}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -Y_{\dot{v}}v - Y_{\dot{r}}r \\ 0 & 0 & X_{\dot{u}}u \\ Y_{\dot{v}}v + Y_{\dot{r}}r & -X_{\dot{u}}u & 0 \end{bmatrix} \tag{5.52}$$

is a Coriolis matrix of the added mass.

Remark 1 (Nonlinear Damping Terms) *The nonlinear damping matrix $D_{nl}(\boldsymbol{\nu})$ in* (5.51) *is often approximated by means of so-called second-order modulus terms*

$$X_{uuu}u^3 \approx X_{|u|u}|u|u, \; Y_{vvr}v^2r \approx Y_{|v|r}|v|r, \tag{5.53}$$

etc., such that an alternative representation

$$D_{nl}(\boldsymbol{\nu}) = \begin{bmatrix} X_{|u|u}|u| & 0 & 0 \\ 0 & Y_{|v|v}|v| + Y_{|r|v}|r| & Y_{|r|r}|r| + Y_{|v|r}|v| \\ 0 & N_{|v|v}|v| + N_{|r|v}|r| & N_{|r|r}|r| + N_{|v|r}|v| \end{bmatrix} \tag{5.54}$$

can be obtained. This representation is not continuously-differentiable, but has a directional derivative and appears to fit empirical data more accurately [142, Sec. B.1.2].

Remark 2 (Added Mass Coriolis Matrix) *The motivation for the substitutions suggested by, e.g., [131] in the damping term* (5.50) *resulting in the added mass Coriolis matrix* (5.52) *appear to be somewhat arbitrary. However, they are motivated by derivation of the forces induced due to the water based on Kirchhoff's equations as shown in, e.g., [109, Sec 18.43], and [132, Sec. 3.1], where* (5.44) *and* (5.52) *are considered independently of the Taylor polynomial of the velocity-dependent forces* (5.48).

5.2.3. Wind Forces

The wind forces acting on the ship can be calculated using

$$\boldsymbol{\tau}_{\mathrm{w}} = \frac{1}{2}\rho_{\mathrm{a}}V_{\mathrm{w,a}}^2 \begin{bmatrix} C_X(\gamma_{\mathrm{w,a}})A_{\mathrm{t}} \\ C_Y(\gamma_{\mathrm{w,a}})A_{\mathrm{l}} \\ C_N(\gamma_{\mathrm{w,a}})A_{\mathrm{l}}L_{\mathrm{oa}} \end{bmatrix}, \tag{5.55}$$

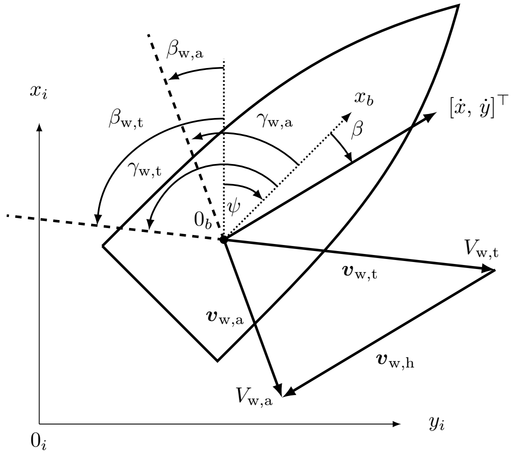


Figure 5.3.: Relations of true and apparent wind w.r.t. the NED frame and the body frame, where $\beta_{\mathrm{w,t}}$, $\beta_{\mathrm{w,a}}$ is the angle of attack of the true and apparent wind in the NED frame, respectively, $\gamma_{\mathrm{w,t}}$, $\gamma_{\mathrm{w,a}}$ are the respective angles of attack of the true and apparent wind w.r.t. the body frame, β is the drift angle, $\boldsymbol{v}_{\mathrm{w,t}}$ is the true wind velocity $\boldsymbol{v}_{\mathrm{w,h}} = -[\dot{x},\ \dot{y}]^\top$ is the headwind velocity, and $\boldsymbol{v}_{\mathrm{w,a}}$ is the apparent wind velocity with $V_{\mathrm{w,t}} = ||\boldsymbol{v}_{\mathrm{w,t}}||_2$, $V_{\mathrm{w,a}} = ||\boldsymbol{v}_{\mathrm{w,a}}||_2$.

where $V_{\mathrm{w,a}}$ is the apparent wind speed, A_{t} is the transversal projected area, A_{l} the lateral projected area, L_{oa} the length over all of the ship, and ρ_{a} is the air density. The wind coefficients $C_X(\gamma_{\mathrm{w,a}}), C_Y(\gamma_{\mathrm{w,a}}), C_N(\gamma_{\mathrm{w,a}})$ are functions of the angle of attack of the apparent wind $\gamma_{\mathrm{w,a}}$ and can be calculated using the empirical formulas stated by Isherwood [76]. Often the much simpler approximation

$$\boldsymbol{\tau}_{\mathrm{w}} = \frac{1}{2}\rho_{\mathrm{a}} V_{\mathrm{w,a}}^2 \begin{bmatrix} -c_x \cos(\gamma_{\mathrm{w,a}}) A_{\mathrm{t}} \\ -c_y \sin(\gamma_{\mathrm{w,a}}) A_{\mathrm{l}} \\ c_n \sin(2\gamma_{\mathrm{w,a}}) A_{\mathrm{l}} L_{\mathrm{oa}} \end{bmatrix}, \tag{5.56}$$

is used [see 47, Sec. 8.1.2]. Another prevalent approximation is given by

$$\boldsymbol{\tau}_{\mathrm{w}} = \frac{1}{2}\rho_{\mathrm{a}} V_{\mathrm{w,a}}^2 \begin{bmatrix} -c_x \cos(\gamma_{\mathrm{w,a}}) A_{\mathrm{t}} \\ -c_y \sin(\gamma_{\mathrm{w,a}}) A_{\mathrm{l}} \\ -c_y \sin(\gamma_{\mathrm{w,a}}) \left(\frac{x_{A,0}}{L_{oa}} + \frac{1}{4} - \frac{|\gamma_{\mathrm{w,a}}|}{2\pi} \right) A_{\mathrm{l}} L_{\mathrm{oa}} \end{bmatrix}, \tag{5.57}$$

where $x_{A,0}$ is the distance to the center of the lateral projected area of the ship, see also Wnęk et al. [159], and Höffmann [71]. The true wind direction w.r.t. the north axis is denoted using $\beta_{\mathrm{w,t}} \in [-\pi,\ \pi)$, see Fig. 5.3. The true wind can be expressed with the true wind speed $V_{\mathrm{w,t}}$ and the true wind direction $\beta_{\mathrm{w,t}}$ using

$$\boldsymbol{v}_{\mathrm{w,t}} = - \begin{bmatrix} V_{\mathrm{w,t}} \cos(\beta_{\mathrm{w,t}}) \\ V_{\mathrm{w,t}} \sin(\beta_{\mathrm{w,t}}) \end{bmatrix}. \tag{5.58}$$

Based on this and the headwind $\boldsymbol{v}_{\mathrm{w,h}} = -[\dot{x},\ \dot{y}]^\top$ the apparent wind is given by

$$\boldsymbol{v}_{\mathrm{w,a}} = \boldsymbol{v}_{\mathrm{w,t}} + \boldsymbol{v}_{\mathrm{w,h}} = -\begin{bmatrix} \dot{x} + V_{\mathrm{w,t}} \cos(\beta_{\mathrm{w,t}}) \\ \dot{y} + V_{\mathrm{w,t}} \sin(\beta_{\mathrm{w,t}}) \end{bmatrix}. \tag{5.59}$$

From this, the apparent wind speed is given by

$$V_{\mathrm{w,a}} = ||\boldsymbol{v}_{\mathrm{w,a}}||_2 = \sqrt{V_{\mathrm{w,t}}^2 + U^2 + 2V_{\mathrm{w,t}} u \cos(\beta_{\mathrm{w,t}} - \psi) + 2V_{\mathrm{w,t}} v \sin(\beta_{\mathrm{w,t}} - \psi)} \tag{5.60}$$

where $U = \sqrt{u^2 + v^2}$ is the vessel speed, and the apparent wind direction w.r.t. the north axis is calculated using

$$\beta_{\mathrm{w,a}} = \operatorname{atan2}(\dot{y} + V_{\mathrm{w,t}} \sin(\beta_{\mathrm{w,t}}),\ \dot{x} + V_{\mathrm{w,t}} \cos(\beta_{\mathrm{w,t}})) \in [-\pi,\ \pi). \tag{5.61}$$

such that the relative angle of attack w.r.t. the x_b-axis is given by

$$\gamma_{\mathrm{w,a}} = \beta_{\mathrm{w,a}} - \psi \in [-\pi, \pi). \tag{5.62}$$

5.2.4. Control Forces

In this section, the control forces $\boldsymbol{\tau}_{\mathrm{c}}$ are analyzed based on actuators, which make up the propulsion (or actuator configuration) of a vessel and take into account both the geometric configuration of the actuators and the (nonlinear) relationship of the control surfaces and the provided thrust of the actuators. To this end, these terms are defined as follows.

Definition 1 (Actuator) *A pneumatic, electric, piezo-electric, hydraulic, or a combination the aforementioned device(s) that relates an (environmental) state, its activating/deflecting/-switching mechanisms (control surfaces) such as valves, pistons, or solenoids, and its orientation (which, depending on the actuator, may be part of the control surfaces) and location w.r.t. to a (body-fixed) coordinate frame to a generalized force (if a location is defined) or thrust vector (if no location is defined). Also called actor or effector.*

Definition 2 (Propulsion System) *A system that consists a set of actuators in a certain geometric constellation and calculates the total generalized force exerted by the individual actuators that results on a (rigid) body. Also called actuator configuration.*

In the general, nonlinear equations of motion of an MSV, (5.72) the vector of generalized input forces $\boldsymbol{\tau}_{\mathrm{c}}$ can be studied in more detail. If the actuator configuration is known as, e.g., shown in Fig. 5.4, the relationship between the actual control inputs, i.e., the control surfaces (or manipulated variables) of the vessel $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ and the generalized input forces can be formulated mathematically using the actuator configuration model

$$\boldsymbol{\tau}_{\mathrm{c}} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}), \tag{5.63}$$

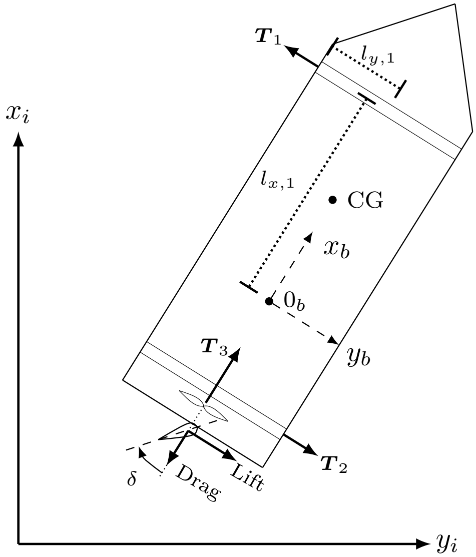


Figure 5.4.: Example actuator configuration with $n_a = 3$ actuators, namely, a combined propeller-rudder system with rudder angle δ and two tunnel thrusters with actuator location $\boldsymbol{p}_b^{ba_j} = [l_{x,j},\ l_{y,j}]^\top$ shown for the bow tunnel thruster $\boldsymbol{T}_1$.

which is, in general, nonlinear. In view of the derivation of dynamical vessel model, the force and moment generated by each actuator must be expressed w.r.t. the body-fixed frame. To this end, the geometric location $\boldsymbol{p}_b^{ba_j} = [l_{x,j},\ l_{y,j}]^\top$, $j = 1, \dots, n_a$ of the n_a individual actuators is defined such that the respective thrust $\boldsymbol{T}_j(\boldsymbol{x}, \boldsymbol{u}_j) = [T_{x,j},\ T_{y,j}]^\top \in \mathbb{R}^2$, $j = 1, \dots, n_a$ generated by each actuator can be described depending on the system states and on a set of n_{u_j} control surfaces, i.e., $\boldsymbol{u}_j(t) \in \mathbb{R}^{n_{u_j}}$ for the j-th actuator. Note that the actuator's orientation α_j is either considered to be constant in time or part of the control surfaces. The geometric actuator locations are combined in the so-called thrust-configuration matrix

$$B_T = \begin{bmatrix} B_{T,1} & \dots & B_{T,n_a} \end{bmatrix} \in \mathbb{R}^{3\times 2n_a}, \quad B_{T,j} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -l_{y,j} & l_{x,j} \end{bmatrix}, \quad j = 1, \dots, n_a, \tag{5.64}$$

which effectively translates the respective actuator's thrust to a force acting in the CG of the vessel according to Newton's second law, and a resultant moment w.r.t. the body-fixed frame according to Euler's second law, i.e.,

$$\boldsymbol{\tau}_{c,j} = \begin{bmatrix} T_{x,j} \\ T_{y,j} \\ \boldsymbol{p}_b^{ba_j} \times \boldsymbol{T}_j \end{bmatrix} = B_{T,j} \boldsymbol{T}_j(\boldsymbol{x}, \boldsymbol{u}_j). \tag{5.65}$$

With this $\boldsymbol{\tau}_\mathrm{c} = \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u}) = B_T\boldsymbol{T}(\boldsymbol{x},\boldsymbol{u})$ gives the sum off all forces resulting from the individual actuator, where

$$\boldsymbol{T}(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix} \boldsymbol{T}_1(\boldsymbol{x},\boldsymbol{u}_1) \\ \vdots \\ \boldsymbol{T}_{n_a}(\boldsymbol{x},\boldsymbol{u}_{n_a}) \end{bmatrix}, \tag{5.66}$$

with $\boldsymbol{T}(\cdot,\cdot) \in \mathbb{R}^{2n_a}$, $\boldsymbol{u}(t) = [\boldsymbol{u}_1^\top(t),\dots,\boldsymbol{u}_{n_a}^\top(t)]^\top \in \mathbb{R}^{n_u}$ such that $n_u = \sum_j^{n_a} n_{u_j}$ is the total number of control surfaces.

Tunnel Thruster

The thrust generated by a tunnel thruster, which usually generates thrust in the direction of the positive or negative y_b-axis, i.e., its orientation $\alpha_\mathrm{tt} = \pi/2 = \text{const.}$ can be modeled using $n_{u,\mathrm{tt}} = 1$ control surface, i.e., the rotational speed of the impeller ω_tt, which yields

$$\boldsymbol{T}_\mathrm{tt}(\boldsymbol{x},\boldsymbol{u}_\mathrm{tt}) = \begin{bmatrix} 0 \\ T_{|\omega_\mathrm{tt}|\omega_\mathrm{tt}}|\omega_\mathrm{tt}|\omega_\mathrm{tt} \end{bmatrix}, \tag{5.67}$$

where $T_{|\omega_\mathrm{tt}|\omega_\mathrm{tt}} > 0$ is a thrust coefficient.

Propeller-Rudder System

A propeller-rudder system can be modeled as a combined actuator. As can be seen from Fig. 5.4, this constellation has two inputs, namely, the rotational speed of the propeller and the rudder angle δ_pr. The orientation of the system is usually constant in time and aligned with the x_b-axis, i.e., $\alpha_\mathrm{pr} = 0 = \text{const.}$. The total thrust generated by this propeller-rudder configuration can be calculated with

$$\boldsymbol{T}_\mathrm{pr}(\boldsymbol{x},\boldsymbol{u}_\mathrm{pr}) = \begin{bmatrix} \overline{T}_{x,\mathrm{pr}} - T_\mathrm{D}(u,\delta_\mathrm{pr}) \\ T_\mathrm{L}(u,\delta_\mathrm{pr}) \end{bmatrix}, \tag{5.68}$$

where $T_\mathrm{D}(u,\delta_\mathrm{pr})$ is the drag force caused by a nonzero rudder angle, which impedes the nominal thrust

$$\overline{T}_{x,\mathrm{pr}} = \begin{cases} T^+_{|\omega_\mathrm{pr}|\omega_\mathrm{pr}}|\omega_\mathrm{pr}|\omega_\mathrm{pr} - T^+_{|\omega_\mathrm{pr}|u}|\omega_\mathrm{pr}|u, & \text{if } \omega_\mathrm{pr} \geq 0 \\ T^-_{|\omega_\mathrm{pr}|\omega_\mathrm{pr}}|\omega_\mathrm{pr}|\omega_\mathrm{pr} - T^-_{|\omega_\mathrm{pr}|u}|\omega_\mathrm{pr}|u, & \text{else} \end{cases} \tag{5.69}$$

generated by the propeller. Compared to a tunnel thruster setup with impeller the nominal thrust generated by the propeller in this configuration must take into account the relative speed of the vessel through the water. Furthermore, this configuration generates a lift force $T_\mathrm{L}(u,\delta_\mathrm{pr})$ induced due to a nonzero rudder angle that is used to turn or brake the ship. The lift force depends on the nominal thrust generated by its preceding propeller, which accelerates the water toward the rudder. This only takes effect if the vessel moves forward through the water. Otherwise, the rudder influence becomes very small for generating lift. This can be expressed mathematically using

$$u_{\mathrm{rud,pr}} = \begin{cases} u + k_u \left(\sqrt{\frac{8}{\pi \rho d_{\mathrm{pr}}^2} \overline{T}_{x,\mathrm{pr}} + u^2} - u \right) & \text{if } u \geq 0 \\ u & \text{else} \end{cases}, \tag{5.70}$$

where $u_{\mathrm{rud,pr}}$ is the water velocity at the rudder, ρ is the water density, d_{pr} is the propeller diameter, and $k_u \approx 0.5$ if the propeller is close to the rudder. In total, the generated lift is expressed in terms of the rudder angle and the water speed at the rudder as

$$T_{\mathrm{L}}(u, \delta_{\mathrm{pr}}) = \begin{cases} (L^{+}_{\delta_{\mathrm{pr}}} \delta_{\mathrm{pr}} - L^{+}_{|\delta_{\mathrm{pr}}|\delta_{\mathrm{pr}}} |\delta_{\mathrm{pr}}|\delta_{\mathrm{pr}}) |u_{\mathrm{rud,pr}}| u_{\mathrm{rud,pr}} & \text{if } u_{\mathrm{rud,pr}} \geq 0 \\ (L^{-}_{\delta_{\mathrm{pr}}} \delta_{\mathrm{pr}} - L^{-}_{|\delta_{\mathrm{pr}}|\delta_{\mathrm{pr}}} |\delta_{\mathrm{pr}}|\delta_{\mathrm{pr}}) |u_{\mathrm{rud,pr}}| u_{\mathrm{rud,pr}} & \text{else} \end{cases}. \tag{5.71}$$

The drag force $T_{\mathrm{D}}(u, \delta_{\mathrm{pr}})$ can be seen as an additional contribution in the damping matrix $D(\boldsymbol{\nu})$ and is therefore ignored in the actuator model.

5.3. Nonlinear Input-Affine Ordinary Differential Equation

All in all, the equations of motion for the MSV can be expressed mathematically as

$$\dot{\boldsymbol{\eta}} = R_{z,\psi} \boldsymbol{\nu}, \qquad t > t_0, \quad \boldsymbol{\eta}(t_0) = \hat{\boldsymbol{\eta}}_0, \tag{5.72a}$$

$$M \dot{\boldsymbol{\nu}} = -(C(\boldsymbol{\nu}) + D(\boldsymbol{\nu}))\boldsymbol{\nu} + \boldsymbol{\tau}_{\mathrm{c}}, \qquad t > t_0, \quad \boldsymbol{\nu}(t_0) = \hat{\boldsymbol{\nu}}_0, \tag{5.72b}$$

where $\boldsymbol{\eta} = [x,\ y,\ \psi]^\top$ is the pose of the MSV, $\boldsymbol{\nu} = [u,\ v,\ r]^\top$ are the body-fixed velocities, $\boldsymbol{\tau}_{\mathrm{c}} \in \mathbb{R}^3$ is the vector of generalized input forces,

$$M = M_{\mathrm{rb}} - M_{\mathrm{am}}, \tag{5.73a}$$

$$C(\boldsymbol{\nu}) = C_{\mathrm{rb}}(\boldsymbol{\nu}) - C_{\mathrm{am}}(\boldsymbol{\nu}), \tag{5.73b}$$

$$D(\boldsymbol{\nu}) = D_{\mathrm{l}} + D_{\mathrm{nl}}(\boldsymbol{\nu}), \tag{5.73c}$$

are the mass-, Coriolis-, and damping matrix, respectively. Depending on the application, the actuator configuration according to Sec. 5.2.4 may or may not be substituted for the vector of control forces $\boldsymbol{\tau}_{\mathrm{c}}$ in the ODE. For the general case, when the actuator models are not explicitly included, (5.72a) can be expressed as a general, nonlinear input-affine ODE

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + B_\tau \boldsymbol{u}, \qquad t > t_0, \quad \boldsymbol{x}(0) = \hat{\boldsymbol{x}}_0, \tag{5.74}$$

where

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} R_{z,\psi} \boldsymbol{\nu} \\ -M^{-1}(C(\boldsymbol{\nu}) + D(\boldsymbol{\nu}))\boldsymbol{\nu} \end{bmatrix}, \quad B_\tau = \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix}, \tag{5.75}$$

and $\boldsymbol{x} = [\boldsymbol{\eta}^\top,\ \boldsymbol{\nu}^\top]^\top$ is the state vector and $\boldsymbol{u} = \boldsymbol{\tau}_{\mathrm{c}}$ is the input with $B_\tau \in \mathbb{R}^{n_x \times 3}$.

5.4. Autopilot (Nomoto) Model

The nonlinear model (5.72), or (5.74) is useful for control design and simulation. However, simplified models can be used to provide further insight to the behavior of marine craft and can also be used to design simple controllers such as autopilots. For a vessel moving at a constant forward speed, i.e., $\boldsymbol{\nu}_0 = [u_0,\ 0,\ 0]^\top$ and $\boldsymbol{\tau}_{\mathrm{c},0} = (C(\boldsymbol{\nu}_0) + D(\boldsymbol{\nu}_0))\boldsymbol{\nu}_0$ a linearization of the kinetic equation (5.72b) leads to a decoupled surge and sway-yaw system

$$M\Delta\dot{\boldsymbol{\nu}} = \begin{bmatrix} a_{11}(u_0) & \mathbf{0}^\top \\ \mathbf{0} & N(u_0) \end{bmatrix} \Delta\boldsymbol{\nu} + \Delta\boldsymbol{\tau}_{\mathrm{c}}, \tag{5.76}$$

where $\Delta\boldsymbol{\nu} = \boldsymbol{\nu} - \boldsymbol{\nu}_0,\ \Delta\boldsymbol{\tau}_{\mathrm{c}} = \boldsymbol{\tau}_{\mathrm{c}} - \boldsymbol{\tau}_{\mathrm{c},0}$, and

$$N(u_0) = \begin{bmatrix} Y_v & Y_r + (X_{\dot{u}} - m)u_0 \\ N_v + (Y_{\dot{v}} - X_{\dot{u}})u_0 & N_r + (Y_{\dot{v}} - mx_g)u_0 \end{bmatrix}. \tag{5.77a}$$

Based on this, an autopilot model can be derived with $\bar{\boldsymbol{\tau}}_{\mathrm{c}} = [\Delta\tau_{\mathrm{c},v},\ \Delta\tau_{\mathrm{c},r}]^\top = [\tau_{\mathrm{c},v},\ \tau_{\mathrm{c},r}]^\top = \boldsymbol{b}\delta$, i.e., the sway-yaw dynamics are controlled using a rudder with ruder angle δ. Furthermore, introducing $\bar{\boldsymbol{\nu}} = [\Delta v,\ \Delta r]^\top = [v,\ r]^\top$,

$$\bar{M} = \begin{bmatrix} m_{22} & m_{23} \\ m_{32} & m_{33} \end{bmatrix},$$

and assuming that the yaw rate is measured, the kinetic autopilot model can be written as

$$\bar{M}\dot{\bar{\boldsymbol{\nu}}} = N(u_0)\bar{\boldsymbol{\nu}} + \boldsymbol{b}\delta, \tag{5.78a}$$

$$y = \boldsymbol{c}^\top\bar{\boldsymbol{\nu}}, \tag{5.78b}$$

where $\boldsymbol{c}^\top = [0,\ 1]$. Therefore, the input-output behavior can be expressed using the Laplace transform of (5.78), i.e.,

$$g(s) = \frac{r(s)}{\delta(s)} = \boldsymbol{c}^\top(s\bar{M} - N(u_0))^{-1}\boldsymbol{b} = \frac{K(1 + T_3 s)}{(1 + T_1 s)(1 + T_2 s)} \tag{5.79}$$

which is known as the Nomoto model.

Based on the Nomoto model, the characteristic polynomial of the system matrix can be used to determine the directional stability of the vessel. More specifically, this evaluates to

$$\det(s\bar{M} - N(u_0)) = as^2 + bs + c = 0 \tag{5.80a}$$

$$= \det\bar{M}s^2 + \mathrm{tr}(\mathrm{adj}(\bar{M})N(u_0))s + \det N(u_0) = 0 \tag{5.80b}$$

and the directional stability of the vessel model can be assessed based on the zeros of this particular polynomial. Therein,

$$\begin{aligned}
a &= (m - Y_{\dot{v}})(J_{zz} - N_{\dot{r}}) - (mx_g - N_{\dot{v}})(mx_g - Y_{\dot{r}}), && (5.81\text{a})\\
b &= (m - Y_{\dot{v}})(N_r + (Y_{\dot{v}} - mx_g)u_0) + Y_v(J_{zz} - N_{\dot{r}})\\
&\quad - (mx_g - Y_{\dot{r}})(N_v + (Y_{\dot{v}} - X_{\dot{u}})u_0) && (5.81\text{b})\\
&\quad - (Y_r + (X_{\dot{u}} - m)u_0)(mx_g - N_{\dot{v}}),\\
c &= Y_v(N_r + (Y_{\dot{v}} - mx_g)u_0) - (Y_r + (X_{\dot{u}} - m)u_0)(N_v + (Y_{\dot{v}} - X_{\dot{u}})u_0), && (5.81\text{c})
\end{aligned}$$

where it can be shown that both a and b are large positive quantities, see [1, Chap. 4], [93, Sec. 10.2.1]. For a second order polynomial to have only negative eigenvalues and, thus, implying a directionally stable vessel, it is sufficient that all coefficients have the same sign [100, Sec. 8.3.2]. As a consequence, the directional stability of the vessel depends on the coefficient (5.81c) that can be rewritten as

$$\begin{aligned}
c = Y_v(N_r - mx_g u_0 + Y_{\dot{v}}) - N_v\,&\underbrace{(Y_r - mu_0 + X_{\dot{u}}u_0)}_{(*)}\\
- \underbrace{((Y_{\dot{v}} - X_{\dot{u}})u_0)}_{(**)}\,&\underbrace{(Y_r - mu_0 + X_{\dot{u}}u_0)}_{(*)}, \qquad (5.82)
\end{aligned}$$

where it can be deduced using, e.g., Tab. 5.1 that the first term is a positive quantity, the sign of the second term depends solely on the sign of N_v since $(*)$ is a large negative quantity. Therefore, if $N_v > 0$, c increases and course stability is improved. On the other hand, $(**) \approx -0.9mu_0$ is a large negative quantity that has a destabilizing effect on the vessel's course. It is also known as the Munk moment that tends to rotate the vessel, which arises due to pressure differences along the vessel hull.

In conclusion, the course stability of a vessel depends largely on the sign and magnitude of N_v and its relation to the Munk moment $(Y_{\dot{v}} - X_{\dot{u}})u_0$. Furthermore, note that the directional stability can be greatly increased by means of the location of the CG: Increasing x_g, i.e., putting the CG further forward in the vessel, directly affects the first term in (5.82). However, note that this may not be desirable since a very course-stable vessel is not as maneuverable as a less stable one.

6. Nonlinear and Optimization-Based Control Design

This chapter deals with the synthesis of nonlinear and optimization-based control laws for the MSV model (5.74). The former typically relies on a nonlinear state transformation that yields a linear transformed dynamical system under a nonlinear feedback law. Subsequently, concepts from linear control theory can be utilized to stabilize the system and track reference trajectories see, e.g., [2, 77]. As an example, the method of exact feedback linearization and flatness-based control for MSVs is presented that is especially suited for dynamic positioning of MSVs.

The latter control method can be further divided into two categories, namely, control allocation and optimal control problems. On the one hand, control allocation deals with the assignment of control inputs of overactuated systems to the individual actuators of the vessel's propulsion system and usually achieves this task by means of a static constrained optimization problem [78, 68]. Therefore, the underlying control allocation problem is introduced and present a numerical solution method based on an interior-point SQP line-search algorithm similar to the algorithms presented in Wächter [153], Byrd et al. [24], Bonnans [19], Chachuat [28].

On the other hand, the theory of optimal control constitutes a special class of a dynamic optimization problem, which is a very powerful alternative to the classical nonlinear approaches such as the exact feedback linearization in the sense that it is able to take into account physical limitations of the system in the form of constraints. These constraints are usually imposed on the input of the system, e.g., to take into account rate constraints or other actuator limitations. However, OCPs are also able to handle state constraints that represent static/dynamic obstacles and traffic-restricted areas on the water. For this reason optimal control problems gain popularity in the context of autonomous surface vessels and are applied in a variety of different applications, e.g., dynamic control allocation [21], dynamic positioning [104], trajectory planning [92], and path following [172]. In this context, the theory of OCPs is presented and its solutions methods along with MPC, which extends the concept of OCPs to a receding horizon control scheme and thus achieves closed-loop control of the system.

6.1. Feedback Linearization

Feedback linearization is a control method that aims to transform the nonlinear system (5.74) into a linear system by introducing a nonlinear state transformation and a nonlinear control law that compensates for the nonlinearities in the transformed system dynamics. This transformation can be found using the concept of the relative degree of a system. In this context it is common (although not necessary) to describe the system dynamics in general input-affine form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \sum_{i=1}^{n_u} \boldsymbol{g}_i(\boldsymbol{x}) u_i, \qquad t > t_0, \qquad \boldsymbol{x}(0) = \hat{\boldsymbol{x}}_0 \in \mathbb{R}^{n_x}, \tag{6.1a}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}), \tag{6.1b}$$

where $\boldsymbol{f} : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_x}$, $\boldsymbol{g}_i : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_x}$, $i = 1, \dots, n_u$ are the drift and input vector fields, respectively, and $\boldsymbol{h} : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_y}$ is the measurement function, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ is the control input, and $\boldsymbol{y} \in \mathbb{R}^{n_y}$ is the output of the system. In the following, it is assumed that $n_y = n_u$. To construct the nonlinear state transformation that transforms the system into the nonlinear equivalent of the controller canonical form, a few concepts of nonlinear control theory are in order [77, 117].

Definition 3 (Lie Derivative) *The Lie derivative of a function $h(\boldsymbol{x})$ along a vector field $\boldsymbol{f}(\boldsymbol{x})$ is defined as*

(i) $L_f^k h(\boldsymbol{x}) = \left(\frac{\partial}{\partial \boldsymbol{x}} L_f^{k-1} h(\boldsymbol{x})\right) \boldsymbol{f}(\boldsymbol{x}), \quad k \in \mathbb{N}.$

(ii) $L_f^0 h(\boldsymbol{x}) = h(\boldsymbol{x}).$

Definition 4 (Vector Relative Degree) *A system of the form* (6.1) *has vector relative degree* $\boldsymbol{r} = [r_1, r_2, \dots, r_{n_y}]^\top$ *and relative degree* $r = \sum_{i=1}^{n_y} r_i$ *if*

(i) $L_{\boldsymbol{g}_j} L_f^k h_i(\boldsymbol{x}) = 0, \quad i, j = 1, \dots, n_y, \quad k = 0, 1, \dots, r_i - 2.$

(ii) the coupling matrix

$$\mathcal{D}(\boldsymbol{x}) = \begin{bmatrix} L_{\boldsymbol{g}_1} L_f^{r_1-1} h_1 & \dots & L_{\boldsymbol{g}_{n_y}} L_f^{r_1-1} h_1 \\ \vdots & & \vdots \\ L_{\boldsymbol{g}_1} L_f^{r_{n_y}-1} h_{n_y} & \dots & L_{\boldsymbol{g}_{n_y}} L_f^{r_{n_y}-1} h_{n_y} \end{bmatrix} \in \mathbb{R}^{n_y \times n_y}$$

is regular.

Based on this, suppose there is a (fictitious) output $\boldsymbol{y} = \boldsymbol{\lambda}(\boldsymbol{x})$ with relative degree $r = n_x$. Then the nonlinear transformation

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}_1 \\ \hdashline \vdots \\ \hdashline \boldsymbol{z}_{n_y} \end{bmatrix} = \begin{bmatrix} z_{1,1} \\ z_{1,2} \\ \vdots \\ z_{1,r_1} \\ \hdashline \vdots \\ \hdashline z_{n_y,1} \\ z_{n_y,2} \\ \vdots \\ z_{n_y,r_{n_y}} \end{bmatrix} = \Phi(\boldsymbol{x}) = \begin{bmatrix} \Phi_1(\boldsymbol{x}) \\ \hdashline \vdots \\ \hdashline \Phi_{n_y}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \lambda_1(\boldsymbol{x}) \\ \vdots \\ L_f^{r_1-1}\lambda_1(\boldsymbol{x}) \\ \hdashline \vdots \\ \hdashline \lambda_{n_y}(\boldsymbol{x}) \\ \vdots \\ L_f^{r_{n_y}-1}\lambda_{n_y}(\boldsymbol{x}) \end{bmatrix} \tag{6.2}$$

transforms the system (6.1a) into the nonlinear controller canonical form

$$\begin{aligned}
\dot{z}_{1,1} &= z_{1,2} && (6.3a)\\
\dot{z}_{1,2} &= z_{1,3} && (6.3b)\\
&\vdots\\
\dot{z}_{1,r_1} &= L_f^{r_1}\lambda_1(\boldsymbol{x}) + [L_{g_1}L_f^{r_1-1}\lambda_1(\boldsymbol{x}), \dots, L_{g_{n_y}}L_f^{r_1-1}\lambda_1(\boldsymbol{x})]\boldsymbol{u} && (6.3c)\\
&\vdots\\
\dot{z}_{n_y,1} &= z_{n_y,2} && (6.3d)\\
\dot{z}_{n_y,2} &= z_{n_y,3} && (6.3e)\\
&\vdots\\
\dot{z}_{n_y,r_{n_y}} &= L_f^{r_{n_y}}\lambda_{n_y}(\boldsymbol{x}) + [L_{g_1}L_f^{r_{n_y}-1}\lambda_{n_y}(\boldsymbol{x}), \dots, L_{g_{n_y}}L_f^{r_{n_y}-1}\lambda_{n_y}(\boldsymbol{x})]\boldsymbol{u} && (6.3f)
\end{aligned}$$

with transformed state $\boldsymbol{z}$, see [23]. Also refer to [2, 77] on more information about the existence and calculation of a fictitious output $\boldsymbol{\lambda}(\boldsymbol{x})$ with relative degree $r = n_x$ and the case where $r < n_x$, i.e., the case when there is no such fictitious output. Using the transformation (6.2) it is evident that the nonlinear control law

$$\boldsymbol{u} = \mathcal{D}^{-1}(\boldsymbol{x})(-\boldsymbol{a}(\boldsymbol{x}) + \boldsymbol{v}), \tag{6.4}$$

exactly linearizes and decouples the nonlinear system, where

$$\boldsymbol{a}(\boldsymbol{x}) = \begin{bmatrix} L_f^{r_1}\lambda_1(\boldsymbol{x}) \\ \vdots \\ L_f^{r_{n_y}}\lambda_{n_y}(\boldsymbol{x}) \end{bmatrix} \tag{6.5}$$

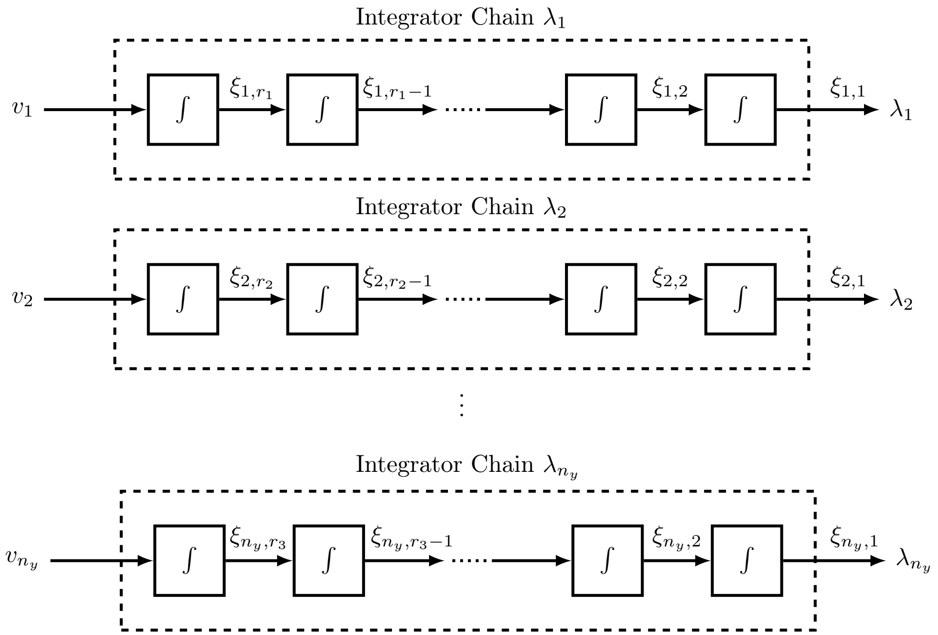


Figure 6.1.: Brunovsky canonical form of the nonlinear system (6.1) under the nonlinear control law (6.4) with new input $\boldsymbol{v}$, vectorial relative degree $\boldsymbol{r} = [r_1,\ r_2, \dots, r_{n_y}]^\top$ and (fictitious) output $\boldsymbol{\lambda}(\boldsymbol{x})$.

and $\boldsymbol{v} \in \mathbb{R}^{n_y}$ is a new input such that the closed-loop dynamics in the transformed state is in Brunovsky canonical form

$$\dot{\boldsymbol{z}} = \begin{bmatrix} \dot{\boldsymbol{z}}_1 \\ \dot{\boldsymbol{z}}_2 \\ \vdots \\ \dot{\boldsymbol{z}}_{n_y} \end{bmatrix} = \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_{n_y} \end{bmatrix} \boldsymbol{z} + \begin{bmatrix} \boldsymbol{e}_1 & 0 & \dots & 0 \\ 0 & \boldsymbol{e}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \boldsymbol{e}_{n_y} \end{bmatrix} \boldsymbol{v}, \tag{6.6}$$

where

$$A_i = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{r_i \times r_i} \tag{6.7}$$

and $\boldsymbol{e}_i = [0, \dots,\ 0,\ 1]^\top \in \mathbb{R}^{r_i}$ for $i = 1, \dots, n_y$. The Brunovsky canonical form is illustrated in Fig. 6.1.

For the vessel model (5.74), it is evident that $\boldsymbol{g}_i(\boldsymbol{x})$, $i = 1, 2, 3$ is the i-th column of B_τ. Furthermore, $\boldsymbol{\lambda}(\boldsymbol{x}) = \boldsymbol{\eta}$ is an output with relative degree of $r = r_1 + r_2 + r_3 = n_x$, where $r_1 = r_2 = r_3 = 2$. This can be verified by differentiating $\boldsymbol{\eta}$ w.r.t. time twice, i.e.,

$$\dot{\boldsymbol{\eta}} = R_{z,\psi}\boldsymbol{\nu} =: \begin{bmatrix} L_{\boldsymbol{f}}^{r_1-1}\lambda_1(\boldsymbol{x}) \\ L_{\boldsymbol{f}}^{r_2-1}\lambda_2(\boldsymbol{x}) \\ L_{\boldsymbol{f}}^{r_3-1}\lambda_3(\boldsymbol{x}) \end{bmatrix}, \tag{6.8a}$$

$$\begin{aligned} \ddot{\boldsymbol{\eta}} &= \frac{\partial R_{z,\psi}}{\partial \psi}\dot{\psi}\boldsymbol{\nu} + R_{z,\psi}\dot{\boldsymbol{\nu}} \\ &= \frac{\partial R_{z,\psi}}{\partial \psi}\dot{\psi}\boldsymbol{\nu} - R_{z,\psi}M^{-1}(C(\boldsymbol{\nu}) + D(\boldsymbol{\nu}))\boldsymbol{\nu} + R_{z,\psi}M^{-1}\boldsymbol{\tau}_{\text{fb}} \\ &= \boldsymbol{a}(\boldsymbol{x}) + \mathcal{D}(\boldsymbol{x})\boldsymbol{u}, \end{aligned} \tag{6.8b}$$

where

$$\boldsymbol{a}(\boldsymbol{x}) = \frac{\partial R_{z,\psi}}{\partial \psi} r\boldsymbol{\nu} - R_{z,\psi}M^{-1}(C(\boldsymbol{\nu}) + D(\boldsymbol{\nu}))\boldsymbol{\nu} \tag{6.9}$$

is the compensation term (6.5) and

$$\mathcal{D}(\boldsymbol{x}) = R_{z,\psi}M^{-1} \tag{6.10}$$

is the coupling matrix in Def. 4, which is invertible and $\mathcal{D}^{-1}(\boldsymbol{x}) = MR_{z,\psi}^\top$.

Thus, the feedback law

$$\begin{aligned} \boldsymbol{\tau}_{\text{fb}} &= \mathcal{D}^{-1}(\boldsymbol{x})(-\boldsymbol{a}(\boldsymbol{x}) + \boldsymbol{v}) \\ &= -MR_{z,\psi}^\top \frac{\partial R_{z,\psi}}{\partial \psi} r\boldsymbol{\nu} + (C(\boldsymbol{\nu}) + D(\boldsymbol{\nu}))\boldsymbol{\nu} + MR_{z,\psi}^\top \boldsymbol{v} \end{aligned} \tag{6.11}$$

exactly linearizes the system and $\boldsymbol{v}$ is the new control input that can be designed to realize an output tracking of the desired target trajectory $\boldsymbol{\lambda}^* = \boldsymbol{\eta}^*$ or, more explicitly, $\lambda_1^*(\boldsymbol{x}) = x^*,\ \lambda_2^*(\boldsymbol{x}) = y^*,\ \lambda_3^*(\boldsymbol{x}) = \psi^*$, using methods from linear control theory such as, e.g., a pole placement controller of the form

$$\boldsymbol{v} = \begin{bmatrix} \lambda_1^{*(r_1)} - \sum_{j=0}^{r_1-1} p_{1,j}(L_{\boldsymbol{f}}^j\lambda_1(\boldsymbol{x}) - \lambda_1^{*(j)}) \\ \lambda_2^{*(r_2)} - \sum_{j=0}^{r_2-1} p_{2,j}(L_{\boldsymbol{f}}^j\lambda_2(\boldsymbol{x}) - \lambda_2^{*(j)}) \\ \lambda_3^{*(r_3)} - \sum_{j=0}^{r_3-1} p_{3,j}(L_{\boldsymbol{f}}^j\lambda_3(\boldsymbol{x}) - \lambda_3^{*(j)}) \end{bmatrix} = \begin{bmatrix} x^{*(r_1)} - \boldsymbol{p}_1^\top\,(\Phi_1(\boldsymbol{x}) - \boldsymbol{\xi}^*) \\ y^{*(r_2)} - \boldsymbol{p}_2^\top\,(\Phi_2(\boldsymbol{x}) - \boldsymbol{\Upsilon}^*) \\ \psi^{*(r_3)} - \boldsymbol{p}_3^\top\,(\Phi_3(\boldsymbol{x}) - \boldsymbol{\Psi}^*) \end{bmatrix}, \tag{6.12}$$

where $\boldsymbol{p}_i^\top = [p_0, \ldots, p_{r_i}],\ i = 1, 2, 3$ are coefficients of a Hurwitz polynomial. Furthermore,

$$\boldsymbol{\xi}^* = \begin{bmatrix} x^* \\ \dot{x}^* \\ \vdots \\ x^{*(r_1-1)} \end{bmatrix}, \quad \boldsymbol{\Upsilon}^* = \begin{bmatrix} y^* \\ \dot{y}^* \\ \vdots \\ y^{*(r_2-1)} \end{bmatrix}, \quad \boldsymbol{\Psi}^* = \begin{bmatrix} \psi^* \\ \dot{\psi}^* \\ \vdots \\ \psi^{*(r_3-1)} \end{bmatrix} \tag{6.13}$$

are the respective target trajectories, which need to be r_i-times continuously differentiable, and $\boldsymbol{\Phi}_i(\boldsymbol{x}),\ i = 1, 2, 3$ defined according to (6.2).

This pole placement controller dictates the eigenvalues of the trajectory error dynamics in Brunovsky canonical form (6.6).

Remark 3 (Differential Flatness) *Since the vessel model is feedback-linearizable, it is also differentially flat, i.e., $\boldsymbol{z} = \boldsymbol{\eta}$ is also a flat output of the system. Thus, the system states and inputs can be expressed in terms of the flat output and its time derivatives, i.e.,*

$$\boldsymbol{x} = \boldsymbol{\theta}_{\boldsymbol{x}}(\boldsymbol{z}, \dot{\boldsymbol{z}}, \dots, \boldsymbol{z}^{(\beta-1)}), \tag{6.14a}$$

$$\boldsymbol{u} = \boldsymbol{\theta}_{\boldsymbol{u}}(\boldsymbol{z}, \dot{\boldsymbol{z}}, \dots, \boldsymbol{z}^{(\beta)}). \tag{6.14b}$$

Based on (5.72a) *it is evident that*

$$\boldsymbol{x} = \boldsymbol{\theta}_{\boldsymbol{x}}(\boldsymbol{z}, \dot{\boldsymbol{z}}, \dots, \boldsymbol{z}^{(\beta-1)}) = \begin{bmatrix} \boldsymbol{z} \\ R_{z,z_3}^{\top} \dot{\boldsymbol{z}} \end{bmatrix}, \tag{6.15}$$

Moreover, by differentiating $\boldsymbol{\nu} = R_{z,z_3}^{\top} \dot{\boldsymbol{z}}$ w.r.t. time $\dot{\boldsymbol{\nu}} = \frac{\partial R_{z,z_3}}{\partial z_3} \dot{z}_3 \dot{\boldsymbol{z}} + R_{z,z_3}^{\top} \ddot{\boldsymbol{z}}$ is obtained, which can be substituted into (5.72b) *to yield the input parameterization*

$$\begin{aligned} \boldsymbol{\tau}_c &= \boldsymbol{\theta}_{\boldsymbol{u}}(\boldsymbol{z}, \dot{\boldsymbol{z}}, \dots, \boldsymbol{z}^{(\beta)}) \\ &= M R_{z,z_3}^{\top} \ddot{\boldsymbol{z}} - M \frac{\partial R_{z,z_3}}{\partial z_3} \dot{z}_3 \dot{\boldsymbol{z}} + (C(\boldsymbol{\nu})|_{\boldsymbol{\nu}=R_{z,z_3}^{\top}\dot{\boldsymbol{z}}} + D(\boldsymbol{\nu})|_{\boldsymbol{\nu}=R_{z,z_3}^{\top}\dot{\boldsymbol{z}}}) R_{z,z_3}^{\top} \dot{\boldsymbol{z}}. \end{aligned} \tag{6.16}$$

This parameterization can be used to design a feedforward control based on the desired output $\boldsymbol{\eta}^*$ that must be twice continuously differentiable. Efforts have also been made in, e.g., Agrawal and Sira-Ramirez [3, Sec. 9.5] to derive a flat parameterization of (5.72) *for the underactuated case. However, this parameterization relies on several unrealistic assumptions on the parameters and, furthermore, includes a variety of singularities that would render the resulting flat parameterization of the system useless in a real application. An optimization-based control for underactuated systems can be achieved using constraints as shown in [174].*

6.2. Control Allocation

Overactuated mechanical systems are defined by having more independent actuators or control inputs, compared to their DOFs [81]. Suppose the desired generalized force vector $\boldsymbol{\tau}_c^*$ for the ship model (5.74) has been calculated by a high-level controller. In this context, $(\cdot)^*$ denotes the desired value (as calculated by, e.g., a high-level controller) of the respective variable. If the ship is overactuated, the desired generalized force can be realized with several combinations of the individual actuators of the vessel. The overactuation can thus be viewed as a degree of freedom of the controller in the sense that additional goals such as minimum wear of the actuators can be realized. Furthermore, it is often assumed that the actuator dynamics can be neglected since most actuators are provided with low level controllers. In that case, the control allocation problem can be formulated as a static optimization problem, which is the topic of this chapter. Therefore, the principal concepts of control allocation are introduced and an interior-point SQP line-search algorithm for solving the control allocation problem is presented with illustrative examples in Appendix A.

6.2.1. Problem Formulation

The control allocation calculates set points for subordinate or low-level controllers based on the relationship between the generalized forces of the vessel's propulsion system and the thrust provided by the individual actuators that can be stated mathematically using the nonlinear propulsion model as in (5.63), which is stated here again for clarity

$$\boldsymbol{\tau}_{\mathrm{c}} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}). \tag{6.17}$$

With this, the principal goal of control allocation consists in solving the equation

$$\boldsymbol{\tau}_{\mathrm{c}}^{*} - \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{0}, \tag{6.18}$$

for $\boldsymbol{u}$ that has multiple solutions due to the overactuation. In reality, the linear case

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) = B\boldsymbol{u} \tag{6.19}$$

can be solved more efficiently and is more widely-studied. Therein, $B \in \mathbb{R}^{3\times n_u}$ is the (linear) control effectiveness matrix. It is therefore desirable to simplify the nonlinear model. In a digital system, a successive linearization around the previously realized control input $\boldsymbol{u}_{-1} = \boldsymbol{u}(t - \Delta t)$, where Δt is the fundamental step time of the digital system, yields the linearized propulsion model

$$\boldsymbol{\tau}_{\mathrm{c}}^{*} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) \approx \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}_{-1}) + \underbrace{\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}_{-1})}_{B(\boldsymbol{x})}(\boldsymbol{u} - \boldsymbol{u}_{-1}) \tag{6.20}$$

such that the linear relation

$$\boldsymbol{v}^{*} = B(\boldsymbol{x})\boldsymbol{u}, \tag{6.21}$$

is obtained using the new virtual desired input

$$\boldsymbol{v}^{*} = \boldsymbol{\tau}_{\mathrm{c}}^{*} - \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}_{-1}) + B(\boldsymbol{x})\boldsymbol{u}_{-1} \tag{6.22}$$

and a linear control allocation can be performed using this model. Note that the control effectiveness matrix of the linearized model $B(\boldsymbol{x})$ can be state-dependent in the nonlinear case. Based on this, a control allocation can be performed if $\operatorname{rank} B_u < n_u$, where $B_u = B_\tau B$ with B_τ defined in (5.75). In that case B_u has a null space of dimension $3 - \operatorname{rank} B_u$ or, in other words, there exist different assignments of the actuators $\boldsymbol{u}$ that do not alter the rate of change of the system states, which is the underlying idea of a control allocation algorithm. Therefore, secondary goals apart from providing the desired generalized forces $\boldsymbol{\tau}_\mathrm{c}^*$ (or $\boldsymbol{v}^*$) can be taken into account including actuator bounds, rate constraints, prioritized actuators, etc. More specifically, if the propulsion model is obtained using a thrust-configuration matrix as described in Sec. 5.2.4, the nonlinear model reads

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) = B_T \boldsymbol{T}(\boldsymbol{x}, \boldsymbol{u}), \tag{6.23}$$

and (6.22) evaluates to $\boldsymbol{v}^* = \boldsymbol{\tau}_\mathrm{c}^* - B_T \boldsymbol{T}(\boldsymbol{x}, \boldsymbol{u}_{-1}) + B_T \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}_{-1})\boldsymbol{u}_{-1}$ such that $B(\boldsymbol{x}) = B_T \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}_{-1})$ is the (linearized) control effectiveness matrix.

Based on the linear or linearized actuator model, the control allocation can be formulated using a (quadratic) optimization problem[1]

$$\begin{aligned} \min_{\boldsymbol{u},\, \boldsymbol{s}}\ & c(\boldsymbol{u}, \boldsymbol{s}) = \frac{1}{2}||\boldsymbol{u} - \boldsymbol{u}_\mathrm{ref}||^2_{W_u} + \frac{\gamma}{2}||\boldsymbol{s}||^2_{W_\tau} && \text{(6.24a)} \\ \text{s.t.}\ & \\ & \boldsymbol{v}^* = B(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{s} && \text{(6.24b)} \\ & \boldsymbol{u}^- \leq \boldsymbol{u} \leq \boldsymbol{u}^+ && \text{(6.24c)} \\ & -\Delta t \dot{\boldsymbol{u}}^\mathrm{max} \leq \boldsymbol{u} - \boldsymbol{u}_{-1} \leq \Delta t \dot{\boldsymbol{u}}^\mathrm{max}, && \text{(6.24d)} \end{aligned}$$

where (6.24a) is the cost function to be minimized, (6.24b) is the fundamental control allocation constraint in the virtual input, (6.24c) are bound constraints with lower and upper bounds $\boldsymbol{u}^-$, $\boldsymbol{u}^+$, respectively, and (6.24d) are rate constraints of the control inputs, where $\boldsymbol{u}_{-1}$ are the inputs of the previous iteration. Moreover, $\boldsymbol{u}_\mathrm{ref}$ is a user-defined reference value for the actuators that is used, e.g., to define preferred, trimmed, or previous control surface values [78]. Furthermore, $\boldsymbol{w} = [\boldsymbol{u}^\top,\ \boldsymbol{s}^\top]^\top \in \mathbb{R}^{n_w}$ is the vector of decision variables, $\boldsymbol{s} \in \mathbb{R}^3$ is a vector of slack variables, $\dot{\boldsymbol{u}}^\mathrm{max}$ is the maximum rate of change of the control surfaces. Additionally, $W_u \in \mathbb{R}^{n_u \times n_u}$ is the control weighting matrix, and $W_\tau \in \mathbb{R}^{3\times 3}$ is the input weighting matrix. Both these matrices must be positive definite and the former is used to prioritize individual actuators and the latter penalizes deviations from the individual desired general forces, respectively. Furthermore, $\gamma \gg 1$ is a user-defined control allocation weight that emphasizes the importance of fulfilling the fundamental control allocation constraint (6.24b) over achieving the reference control input $\boldsymbol{u}_\mathrm{ref}$. See also Härkegård [67] for more details and how W_τ can be chosen. The optimization problem (6.24) is a special case of the more general nonlinear constrained program that is discussed in the following section.

[1] This is only the case due to the assumption that actuator dynamics can be neglected. If this is not a viable assumption, methods as shown in, e.g., [21] can be used, is where the control allocation is formulated as an OCP, i.e., a dynamic optimization problem.

6.2.2. Numerical Solution

The control allocation relies on a static optimization problem to be solved. In this section, the main theoretical results of nonlinear constrained static optimization problems are discussed. Moreover, numerical methods to solve these types of problems are discussed using an exemplary line search interior-point algorithm. An NLP has the form

$$\min_{\boldsymbol{w}} c(\boldsymbol{w}) \tag{6.25a}$$

$$\text{s.t.}$$

$$\boldsymbol{g}(\boldsymbol{w}) = \boldsymbol{0} \tag{6.25b}$$

$$\boldsymbol{h}(\boldsymbol{w}) \leq \boldsymbol{0}, \tag{6.25c}$$

where $c : \mathbb{R}^{n_w} \mapsto \mathbb{R}$ is the cost function to be minimized, $\boldsymbol{w} \in \mathbb{R}^{n_w}$ are the decision variables, $\boldsymbol{g} : \mathbb{R}^{n_w} \mapsto \mathbb{R}^{n_g}$ are the equality and $\boldsymbol{h} : \mathbb{R}^{n_w} \mapsto \mathbb{R}^{n_h}$ are the inequality constraints. An inequality constraint $h_i(\boldsymbol{w})$ is said to be inactive if at the solution $\boldsymbol{w}^*$ it holds that $h_i(\boldsymbol{w}^*) < 0$. Likewise, an inequality constraint $h_i(\boldsymbol{w})$ is said to be active if at the solution $\boldsymbol{w}^*$ it holds that $h_i(\boldsymbol{w}^*) = 0$. Thus, every equality constraint is active. A point $\boldsymbol{w}$ is said to be feasible, if (6.25b) and (6.25c) are satisfied.

The (first order necessary) optimality Karush-Kuhn-Tucker (KKT) conditions for problems of the form (6.25) are based on the introduction of Lagrange multipliers $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ associated with the equality and inequality constraints, respectively. They are formulated as

$$\nabla_{\boldsymbol{w}} c(\boldsymbol{w}^*) + \nabla_{\boldsymbol{w}} \boldsymbol{g}(\boldsymbol{w}^*)\boldsymbol{\lambda}^* + \nabla_{\boldsymbol{w}} \boldsymbol{h}(\boldsymbol{w}^*)\boldsymbol{\mu}^* = \boldsymbol{0} \tag{6.26a}$$

$$\boldsymbol{h}^\top(\boldsymbol{w}^*)\boldsymbol{\mu}^* = 0 \tag{6.26b}$$

$$\boldsymbol{\mu}^* \geq \boldsymbol{0} \tag{6.26c}$$

$$\boldsymbol{g}(\boldsymbol{w}^*) = \boldsymbol{0} \quad \boldsymbol{h}(\boldsymbol{w}^*) \leq \boldsymbol{0}, \tag{6.26d}$$

where (6.26a) signifies that at the optimal solution the gradient of the cost function is a linear combination (expressed using the Lagrange multipliers) of the constraint gradients. Furthermore, (6.26b) is known as the complementary slackness condition, implying that all Lagrange multipliers of inactive constraints must be zero. Finally, (6.26c), and (6.26d) ensure feasibility of the solution.

There are several strategies to solve the optimization problem (6.25) numerically. For example, active set methods aim to iteratively identify the set of active inequality constraints at the solution and solve a sequence of reduced equality constrained problems with this so-called working set. After each iteration, the working set is updated and the problem solved anew. Alternatively, interior-point methods incorporate the inequality constraints into the cost function, which also results in a sequence of equality constrained problems. However, the cost function is modified in such a way that the solution of the problem is always feasible, which is not the case for active set methods. For a thorough overview of different numerical algorithms see Nocedal and Wright [119], Chachuat [28], Betts [15], Bonnans [19], Conn et al. [30], Luenberger and Ye [99]. In the following, a merit

```
Input: $\boldsymbol{w}_0$, $\mu_0$, $maxIter$, $\epsilon_{\text{opt}}$, $\epsilon_{\text{con}}$
Output: $\boldsymbol{w}^*$, $\boldsymbol{\lambda}^*$, $\boldsymbol{\sigma}^*$
$i \leftarrow 1$
$\boldsymbol{s}_0 \leftarrow \max(1\text{e}{-1}, -\boldsymbol{h}(\boldsymbol{w}_0))$
$(\boldsymbol{\lambda}_0, \boldsymbol{\sigma}_0) \leftarrow \operatorname{argmin}_{\boldsymbol{\lambda},\boldsymbol{\sigma}} \frac{1}{2}||\mathcal{L}(\boldsymbol{w}_0, \boldsymbol{s}_0, \boldsymbol{\lambda}, \boldsymbol{\sigma})||_2^2$
$isConverged$ ← false
while $i <= maxIter \wedge\ !isConverged$ do
    solve the underlying subproblem using algorithm 2
    if (6.56)-(6.58) are fulfilled then
        $isConverged$ ← true
    end
    calculate $\mu_{i+1}$ according to (6.29) or (6.30)
    $i \leftarrow i + 1$
end
```

Algorithm 1: The outer loop of the merit function-based interior-point line search SQP algorithm.

function-based line search interior-point algorithm is presented that is based on the ideas presented in Nocedal and Wright [119], Bonnans [19], Waltz et al. [154], Wächter [153]. The algorithm consists of an outer loop, which is illustrated in Algorithm 1, and an inner loop, which is summarized in Algorithm 2.

6.2.2.1. Barrier Problem

The general nonlinear program (6.25) can be transformed into a box- and equality constrained nonlinear program

$$\min_{\boldsymbol{w},\boldsymbol{s}} c(\boldsymbol{w}) \tag{6.27a}$$

$$\text{s.t.}$$

$$\boldsymbol{g}(\boldsymbol{w}) = \boldsymbol{0} \tag{6.27b}$$

$$\boldsymbol{h}(\boldsymbol{w}) + \boldsymbol{s} = \boldsymbol{0} \tag{6.27c}$$

$$\boldsymbol{s} \geq \boldsymbol{0} \tag{6.27d}$$

at the cost of additional slack variables $\boldsymbol{s}$ and box constraints (6.27d). However, the box constraint can be dealt with by means of a logarithmic barrier function that is augmented to the cost function, i.e.,

$$\min_{\boldsymbol{w},\boldsymbol{s}} \varphi_\mu(\boldsymbol{w},\boldsymbol{s}) = c(\boldsymbol{w}) - \mu \sum_j^{n_h} \log(s_j) \tag{6.28a}$$

s.t.

$$\boldsymbol{g}(\boldsymbol{w}) = \boldsymbol{0} \tag{6.28b}$$

$$\boldsymbol{h}(\boldsymbol{w}) + \boldsymbol{s} = \boldsymbol{0}, \tag{6.28c}$$

where φ_μ is the barrier cost function which approaches infinity as the slack variables reach their lower bound of zero. Therefore, a barrier parameter μ is introduced and a sequence of problems of the form (6.28) is solved for decreasing values of the barrier parameter μ. The barrier parameter is reduced according to [154, Sec.3.5.]

$$\mu \leftarrow \frac{\mu}{100} \tag{6.29}$$

if the underlying sub-problem has been solved in less than four iterations, and

$$\mu \leftarrow \frac{\mu}{5} \tag{6.30}$$

otherwise. With this, an equality-constrained NLP is obtained and it can be shown that $\lim_{i\to\infty} \mu = 0 \implies \boldsymbol{w} \to \boldsymbol{w}^*$, where i indicates the iteration count [153, Sec. 2.7],[127]. Introducing the Lagrangian of (6.28), i.e.,

$$\mathcal{L}(\boldsymbol{w}, s, \boldsymbol{\lambda}, \boldsymbol{\sigma}) = \varphi_\mu + \boldsymbol{\lambda}^\top \boldsymbol{g} + \boldsymbol{\sigma}^\top(\boldsymbol{h} + \boldsymbol{s})$$

the first order necessary optimality conditions, i.e.,

$$\nabla\mathcal{L} = [(\nabla_{\boldsymbol{w}}\mathcal{L})^\top, (\nabla_{\boldsymbol{s}}\mathcal{L})^\top, (\nabla_{\boldsymbol{\lambda}}\mathcal{L})^\top, (\nabla_{\boldsymbol{\sigma}}\mathcal{L})^\top]^\top = \boldsymbol{0} \tag{6.31}$$

then read

$$\nabla_{\boldsymbol{w}} c + (\nabla_{\boldsymbol{w}}\boldsymbol{g})^\top\boldsymbol{\lambda} + (\nabla_{\boldsymbol{w}}\boldsymbol{h})^\top\boldsymbol{\sigma} = \boldsymbol{0} \tag{6.32a}$$

$$-\mu S^{-1}\boldsymbol{e} + \boldsymbol{\sigma} = \boldsymbol{0} \tag{6.32b}$$

$$\boldsymbol{g} = \boldsymbol{0} \tag{6.32c}$$

$$\boldsymbol{h} + \boldsymbol{s} = \boldsymbol{0}, \tag{6.32d}$$

where $S = \text{diag}(s_1, \ldots, s_{n_h})$ and $\boldsymbol{e} = [1, \ldots, 1]^\top \in \mathbb{R}^{n_h}$. Multiplying (6.32b) by S yields a modified KKT system

$$\nabla_{\boldsymbol{w}} c + (\nabla_{\boldsymbol{w}}\boldsymbol{g})^\top\boldsymbol{\lambda} + (\nabla_{\boldsymbol{w}}\boldsymbol{h})^\top\boldsymbol{\sigma} = \boldsymbol{0} \tag{6.33a}$$

$$-\mu\boldsymbol{e} + S\boldsymbol{\sigma} = \boldsymbol{0} \tag{6.33b}$$

$$\boldsymbol{g} = \boldsymbol{0} \tag{6.33c}$$

$$\boldsymbol{h} + \boldsymbol{s} = \boldsymbol{0}. \tag{6.33d}$$

From this it is clear that, in the limit, as $\mu \to 0$ (6.33b) together with (6.33d) approaches the complementary slackness condition for inequality-constrained nonlinear programs. In summary, (6.33) constitutes the first order necessary optimality conditions for the barrier problem (6.28). Finding a solution to the KKT equations, therefore, reduces to a multivariate zero-finding problem in the variables $\boldsymbol{w}$, $\boldsymbol{s}$, $\boldsymbol{\lambda}$, $\boldsymbol{\sigma}$ that can be achieved by means of the Newton's method and any variant thereof, i.e., iterate

$$\tilde{\boldsymbol{w}}_{k+1} = \tilde{\boldsymbol{w}}_k + \tilde{\boldsymbol{d}}_k, = \tilde{\boldsymbol{w}}_k - (\nabla^2 \mathcal{L}_k)^{-1} \nabla \mathcal{L}_k \tag{6.34}$$

until some stopping criterion is met, where $\tilde{\boldsymbol{d}}_k = -(\nabla^2 \mathcal{L}_k)^{-1} \nabla \mathcal{L}_k$ is the Newton method search direction and where $\tilde{\boldsymbol{w}} = [\boldsymbol{w}^\top, \boldsymbol{s}^\top, \boldsymbol{\lambda}^\top, \boldsymbol{\sigma}^\top]^\top$. The Newton iteration (6.34) is also known as a local SQP method and can be extended to a global line search method by means of a merit function or a filter in order achieve global convergence. In the following, focus is put on the merit function-based line search method. To this end, the subsequent sections discuss calculating the search direction, extending the local SQP method with a merit function-based line search, taking into account second order corrections, and replacing the Newton method with a quasi-Newton method.

```
Input: $\boldsymbol{w}_0$, $\boldsymbol{s}_0$, $\boldsymbol{\lambda}_0$, $\boldsymbol{\sigma}_0$, $\mu_i$, $maxIter$, $\eta \in (0,1)$, $\rho \in [0.5,1)$, $\tau \in [0,1)$, $B$(optional)
Output: Solution $\boldsymbol{w}^*$, $\boldsymbol{s}^*$, $\boldsymbol{\lambda}^*$, $\boldsymbol{\sigma}^*$, $B$
$j \leftarrow 1$
if useQuasiNewton then
    if B given by user then
        $B_0 \leftarrow$ user-supplied matrix
    else
        approximate Hessian of Lagrangian $B_0$ using finite differences
        $B_0 \leftarrow \frac{1}{2}(B_0 + B_0^\top)$
        if $B_0$ is not positive definite then
            $B_0 \leftarrow I$
        end
    end
end
isConverged $\leftarrow$ false
while $j <= maxIter \wedge !isConverged$ do
    check and possibly correct matrix inertia of $\nabla^2 \mathcal{L}_j$
    calculate primal and dual Newton search directions $\boldsymbol{d}_\mathrm{p}, \boldsymbol{d}_\mathrm{d}$ by solving (6.36)
    apply the fraction-to-the-boundary rule (6.42) to obtain $\alpha_\mathrm{p}^+, \alpha_\mathrm{d}^+$
    update the dual variable estimates $\boldsymbol{\lambda}_j \leftarrow \boldsymbol{\lambda}_j + \alpha_\mathrm{d}^+ \boldsymbol{d}_\mathrm{d}$, and $\boldsymbol{\sigma}_j \leftarrow \boldsymbol{\sigma}_j + \alpha_\mathrm{d}^+ \boldsymbol{d}_\mathrm{d}$
    perform the step length computation according to algorithm 3 to obtain $\alpha_\mathrm{p}$
    perform the Newton step for the primal variables, i.e., $\boldsymbol{z}_j \leftarrow \boldsymbol{z}_j + \alpha_\mathrm{p} \boldsymbol{d}_\mathrm{p}$
    if useQuasiNewton then
        update Hessian of Lagrangian approximation according to (6.54) with (6.55)
    end
    if (6.56)-(6.58) are fulfilled then
        isConverged $\leftarrow$ true
    end
    $j \leftarrow j + 1$
end
```

Algorithm 2: The inner loop, i.e., the globalized line search SQP algorithm.

```
Input: $\boldsymbol{w}_0$, $s_0$, $\boldsymbol{\lambda}_0$, $\boldsymbol{\sigma}_0$, $\mu_i$, $maxIter$, $\eta \in (0,1)$, $\rho \in [0.5,1)$, $\tau \in [0,1)$
Output: Step length $\alpha_\mathrm{p}$
$j \leftarrow 1$
if $\min(\alpha_p^+, \alpha_d^+) > 1\mathrm{e}{-6}$ then
    update the penalty parameter $\nu$ according to (6.47)
    calculate directional derivative $D\phi_\nu(\boldsymbol{z}_j; \mu_i, \boldsymbol{d}_\mathrm{p})$ of the merit function using (6.45)
    if $D\phi_\nu(\boldsymbol{z}_j; \mu_i, \boldsymbol{d}_p) \geq 0$ then
        $lineSearchFinished \leftarrow$ true
    else
        $lineSearchFinished \leftarrow$ false
    end
    $l \leftarrow 0$
    while $\neg lineSearchFinished \land l \leq 25$ do
        if Armijo-type condition (6.44) is fulfilled then
            $lineSearchFinished \leftarrow$ true
        else
            if $l = 0 \land$ decrease in $(\varphi_{\mu_i})_j$ does not compensate increase in $\nu||\tilde{\boldsymbol{g}}_j||$ then
                perform SOC according to (6.50) to obtain $\boldsymbol{d}_\mathrm{soc}, \tilde{\boldsymbol{\lambda}}_\mathrm{soc}$
                apply the fraction-to-the-boundary rule (6.42) to update $\alpha_\mathrm{p}^+, \alpha_\mathrm{d}^+$
                if corrected actual reduction is better than uncorrected actual reduction then
                    $\boldsymbol{d}_\mathrm{p} \leftarrow \alpha_\mathrm{p}^+ \boldsymbol{d}_\mathrm{p} + \boldsymbol{d}_\mathrm{soc}$
                    $\boldsymbol{d}_\mathrm{d} \leftarrow \alpha_\mathrm{d}^+ \boldsymbol{d}_\mathrm{d} + \tilde{\boldsymbol{\lambda}}_\mathrm{soc} - \tilde{\boldsymbol{\lambda}}_j$
                    update the dual variable estimates $\boldsymbol{\lambda}_j \leftarrow \boldsymbol{\lambda}_i + \alpha_\mathrm{d}^+ \boldsymbol{d}_\mathrm{d}$, and $\boldsymbol{\sigma}_j \leftarrow \boldsymbol{\sigma}_i + \alpha_\mathrm{d}^+ \boldsymbol{d}_\mathrm{d}$
                    if Armijo-type condition (6.44) is fulfilled for corrected step then
                        $lineSearchFinished \leftarrow$ true
                    else
                        $\alpha_\mathrm{p} \leftarrow \rho\alpha_\mathrm{p}$
                    end
                else
                    reject the SOC step and $\alpha_\mathrm{p} \leftarrow \rho\alpha_\mathrm{p}$
                end
            else
                $\alpha_\mathrm{p} \leftarrow \rho\alpha_\mathrm{p}$
            end
        end
        $l \leftarrow l + 1$
    end
    $\alpha_\mathrm{p} \leftarrow \alpha_\mathrm{p}^+ \alpha_\mathrm{p}$
else
    $\alpha_\mathrm{p} \leftarrow \alpha_\mathrm{p}^+$
end
```

Algorithm 3: Merit function-based step length computation.

6.2.2.2. Search Direction

Applying the Newton method to (6.33) the search direction $\tilde{\boldsymbol{d}}_k = [\boldsymbol{d}_\mathrm{p}^\top, \boldsymbol{d}_\mathrm{d}^\top]^\top$ is obtained, where $\boldsymbol{d}_\mathrm{p} = [\boldsymbol{d}_w^\top, \boldsymbol{d}_s^\top]^\top$ is the primal search direction and $\boldsymbol{d}_\mathrm{d} = [\boldsymbol{d}_\lambda^\top, \boldsymbol{d}_\sigma^\top]^\top$ is the dual search direction by means of the asymmetric, linear system of equations

$$\nabla^2 \mathcal{L}_k \tilde{\boldsymbol{d}}_k = -\nabla \mathcal{L}_k, \tag{6.35}$$

where

$$\nabla^2 \mathcal{L}_k = \begin{bmatrix} \nabla^2_{\boldsymbol{w}\boldsymbol{w}}\mathcal{L}_k & 0 & \nabla_{\boldsymbol{w}}\boldsymbol{g}_k & \nabla_{\boldsymbol{w}}\boldsymbol{h}_k \\ 0 & \Sigma_k & 0 & S_k \\ (\nabla_{\boldsymbol{w}}\boldsymbol{g}_k)^\top & 0 & 0 & 0 \\ (\nabla_{\boldsymbol{w}}\boldsymbol{h}_k)^\top & I & 0 & 0 \end{bmatrix} \tag{6.36a}$$

$$\nabla \mathcal{L}_k = \begin{bmatrix} \nabla_{\boldsymbol{w}} c_k + (\nabla_{\boldsymbol{w}}\boldsymbol{g}_k)^\top \boldsymbol{\lambda}_k + (\nabla_{\boldsymbol{w}}\boldsymbol{h}_k)^\top \boldsymbol{\sigma}_k \\ -\mu \boldsymbol{e} + S_k \boldsymbol{\sigma}_k \\ \boldsymbol{g}_k \\ \boldsymbol{h}_k + \boldsymbol{s}_k \end{bmatrix}, \tag{6.36b}$$

with $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_{n_h})$, and $\tilde{\boldsymbol{d}}_k = [\boldsymbol{d}_w^\top, \boldsymbol{d}_s^\top, \boldsymbol{d}_\lambda^\top, \boldsymbol{d}_\sigma^\top]^\top$. Note that several modifications can be made in attempt to make use of symmetric solvers for linear systems, e.g., the second row of (6.35) can be multiplied by S^{-1} to obtain the equivalent, symmetric, linear system of equations

$$\nabla^2 \bar{\mathcal{L}}_k = \begin{bmatrix} \nabla^2_{\boldsymbol{w}\boldsymbol{w}}\mathcal{L}_k & 0 & \nabla_{\boldsymbol{w}}\boldsymbol{g}_k & \nabla_{\boldsymbol{w}}\boldsymbol{h}_k \\ 0 & Z_k & 0 & I \\ (\nabla_{\boldsymbol{w}}\boldsymbol{g}_k)^\top & 0 & 0 & 0 \\ (\nabla_{\boldsymbol{w}}\boldsymbol{h}_k)^\top & I & 0 & 0 \end{bmatrix} \tag{6.37}$$

$$\nabla \bar{\mathcal{L}}_k = \begin{bmatrix} \nabla_{\boldsymbol{w}} c_k + (\nabla_{\boldsymbol{w}}\boldsymbol{g}_k)^\top \boldsymbol{\lambda}_k + (\nabla_{\boldsymbol{w}}\boldsymbol{h}_k)^\top \boldsymbol{\sigma}_k \\ -\mu S_k^{-1} \boldsymbol{e} + \boldsymbol{\sigma}_k \\ \boldsymbol{g}_k \\ \boldsymbol{h}_k + \boldsymbol{s}_k \end{bmatrix}, \tag{6.38}$$

where $Z_k = S_k^{-1}\Sigma_k$. The symmetric system of equations can be expressed in a more compact form, i.e.,

$$\begin{bmatrix} W_k & \tilde{A}_k^\top \\ \tilde{A}_k^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_\mathrm{p} \\ \boldsymbol{d}_\mathrm{d} \end{bmatrix} = - \begin{bmatrix} \nabla_{\boldsymbol{z}}\mathcal{L}_k \\ \tilde{\boldsymbol{g}}_k \end{bmatrix}, \tag{6.39}$$

where

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{s} \end{bmatrix}, \quad \tilde{\boldsymbol{g}}(\boldsymbol{z}) = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{w}) \\ \boldsymbol{h}(\boldsymbol{w}) + \boldsymbol{s} \end{bmatrix}, \quad \tilde{A}(\boldsymbol{z}) = \begin{bmatrix} (\nabla_{\boldsymbol{w}}\boldsymbol{g})^\top(\boldsymbol{w}) & 0 \\ (\nabla_{\boldsymbol{w}}\boldsymbol{h})^\top(\boldsymbol{w}) & I \end{bmatrix}, \quad W = \begin{bmatrix} \nabla^2_{\boldsymbol{w}\boldsymbol{w}}\mathcal{L} & 0 \\ 0 & Z \end{bmatrix}.$$

Furthermore, rearranging (6.39) yields the equivalent linear system of equations but in terms of the updated Lagrange multiplier $\tilde{\boldsymbol{\lambda}}_{k+1}$ and the augmented cost φ_μ , i.e.,

$$\begin{bmatrix} W_k & \tilde{A}_k^\top \\ \tilde{A}_k^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_\mathrm{p} \\ \tilde{\boldsymbol{\lambda}}_{k+1} \end{bmatrix} = - \begin{bmatrix} (\nabla_{\boldsymbol{z}}\varphi_\mu)_k \\ \tilde{\boldsymbol{g}}_k \end{bmatrix}, \tag{6.40}$$

where $\tilde{\boldsymbol{\lambda}} = [\boldsymbol{\lambda}^\top, \boldsymbol{\sigma}^\top]^\top$. It can be shown that the KKT matrix $\nabla^2\mathcal{L}$ is positive definite if the so-called matrix inertia is given by $(n_w + n_h, n_g + n_h, 0)$ see, e.g., Nocedal and Wright [119], where the first element is the number of positive, the second the number of negative, and the last the number of zero-eigenvalues, which must be zero. Due to the nature of the Hessian of the Lagrangian, which may be indefinite, the KKT matrix is replaced by $\nabla_{\boldsymbol{ww}}\mathcal{L} + \delta I$ with increasing δ before the search direction is computed until the inertia condition is met. Furthermore, the gradient of the equality constraints may be rank deficient, which can also be taken in the inertia correction by introducing an additional correction term γI, i.e.,

$$\begin{bmatrix} \nabla^2_{\boldsymbol{ww}}\mathcal{L}_k + \delta I & 0 & \nabla_{\boldsymbol{w}}\boldsymbol{g}_k & \nabla_{\boldsymbol{w}}\boldsymbol{h}_k \\ 0 & \Sigma_k & 0 & S_k \\ (\nabla_{\boldsymbol{w}}\boldsymbol{g}_k)^\top & 0 & \gamma I & 0 \\ (\nabla_{\boldsymbol{w}}\boldsymbol{h}_k)^\top & I & 0 & 0 \end{bmatrix} \tag{6.41}$$

where the choice of δ, γ is done according to the algorithm in Nocedal and Wright [119, App. B].

6.2.2.3. Step Length

Taking the full Newton step $\tilde{\boldsymbol{w}}_{k+1} = \tilde{\boldsymbol{w}}_k + \tilde{\boldsymbol{d}}_k$ obtained with (6.36) or (6.39) may lead to constraint violations, especially with regard to the slack variables $\boldsymbol{s}$ and the Lagrange multipliers $\boldsymbol{\sigma}$, which, as $\mu \downarrow 0$, must remain non-negative. It is therefore common to employ a so-called fraction-to-the-boundary rule

$$\alpha_{\mathrm{p}}^{+} = \max\{\alpha \in (0,1] \,:\, \boldsymbol{s}_k + \alpha \boldsymbol{d}_s \geq (1-\tau)\boldsymbol{s}_k\} \tag{6.42a}$$

$$\alpha_{\mathrm{d}}^{+} = \max\{\alpha \in (0,1] \,:\, \boldsymbol{\sigma}_k + \alpha \boldsymbol{d}_\sigma \geq (1-\tau)\boldsymbol{\sigma}_k\} \tag{6.42b}$$

where typically $\tau = 0.995$ to restrict the maximum step length for the primal and dual variables, and thus to avoid the respective variables to approach their boundaries too fast see, e.g., Nocedal and Wright [119, Sec. 19.2]. Therein, α_{p}^{+} and α_{d}^{+} are the maximum primal and dual step length, respectively. Furthermore, a merit function

$$\phi_\nu(\boldsymbol{z};\mu) = \varphi_\mu(\boldsymbol{z}) + \nu||\tilde{\boldsymbol{g}}(\boldsymbol{z})||. \tag{6.43}$$

is used to ensure global convergence using a backtracking line search, that is, accept the largest step length for which the Armijo-type condition

$$\phi_\nu(\boldsymbol{z}_k;\mu) - \phi_\nu(\boldsymbol{z}_k + \alpha_{\mathrm{p}}^{(l)}\boldsymbol{d}_{\mathrm{p}};\mu) \geq -\eta\alpha_{\mathrm{p}}^{(l)} D\phi_\nu(\boldsymbol{z}_k;\mu,\boldsymbol{d}_{\mathrm{p}}) \tag{6.44}$$

is fulfilled. Therein, $\alpha_{\mathrm{p}}^{(l)} = \rho\alpha_{\mathrm{p}}^{(l-1)}$, $l \in \mathbb{N}$ with $\alpha_{\mathrm{p}}^{(0)} = \alpha_{\mathrm{p}}^{+}$, $\eta \in (0,1)$, and $D\phi_\nu(\boldsymbol{z}_k;\mu,\boldsymbol{d}_{\mathrm{p}})$ is the directional derivative of the merit function. In other words, the merit function monitors the progress of the underlying local SQP (Newton) method by means of a trade-off between decreasing the (barrier) cost function and minimizing the constraint violation. The directional derivative is used since the merit function is, in general, non-differentiable due to the presence of the norm but has a directional derivative for which

$$D\phi_\nu(\boldsymbol{z}_k;\mu,\boldsymbol{d}_{\mathrm{p}}) \leq (\nabla_z\varphi_\mu)_k^\top \boldsymbol{d}_{\mathrm{p}} - \nu||\tilde{\boldsymbol{g}}_k|| = -\boldsymbol{d}_{\mathrm{p}}^\top W_k \boldsymbol{d}_{\mathrm{p}} + \tilde{\boldsymbol{g}}_k^\top \tilde{\boldsymbol{\lambda}}_{k+1} - \nu||\tilde{\boldsymbol{g}}_k||, \tag{6.45}$$

holds. Equality holds in (6.45) if the NLP is merely equality constrained. The latter form is obtained from (6.40), where $(\nabla_z \varphi_\mu)_k^\top \boldsymbol{d}_\mathrm{p} = -\boldsymbol{d}_\mathrm{p}^\top W_k \boldsymbol{d}_\mathrm{p} - \boldsymbol{d}_\mathrm{p}^\top \tilde{A}_k^\top \tilde{\boldsymbol{\lambda}}_{k+1}$ and $\tilde{\boldsymbol{g}}_k = -\tilde{A}_k \boldsymbol{d}_\mathrm{p}$. Furthermore, the merit function (6.43) is known to be exact if

$$\nu \geq ||\tilde{\boldsymbol{\lambda}}_{k+1}||_* \tag{6.46}$$

where $||\cdot||_*$ is the dual norm w.r.t. the norm used in (6.43) [20]. Essentially, the local minimizer of an exact merit function coincides with the minimizer $\boldsymbol{w}^*$ of (6.25). In other words, the condition (6.46) ensures that a deviation from the local minimum is so heavily penalized that $\phi_\nu(\boldsymbol{w}^*, \boldsymbol{s}^*; \mu) \geq c(\boldsymbol{w}^*)$. To ensure that (6.46) is fulfilled, the penalty parameter ν is updated each iteration using

$$\nu_k = \begin{cases} \frac{\nu_{k-1} + ||\tilde{\boldsymbol{\lambda}}_{k+1}||_*}{2}, & \text{if } \nu_{k-1} \geq 1.1(||\tilde{\boldsymbol{\lambda}}_{k+1}||_* + \bar{\nu}) \\ \nu_{k-1}, & \text{if } ||\tilde{\boldsymbol{\lambda}}_{k+1}||_* + \bar{\nu} \leq \nu_{k-1} < 1.1||\tilde{\boldsymbol{\lambda}}_{k+1}||_* + \bar{\nu}, \\ \max(1.5\nu_{k-1}, ||\tilde{\boldsymbol{\lambda}}_{k+1}||_* + \bar{\nu}), & \text{else,} \end{cases} \tag{6.47}$$

where $\bar{\nu}$ is a positive constant [see 19, Sec. 17.1].

6.2.2.4. Maratos Effect and Second Order Corrections

The Maratos effect is a phenomenon that hinders convergence of a merit function-based line search algorithm. Usually, a new step is chosen such that any possible increase in the constraint violation is compensated for by a larger decrease in the barrier function value. The Maratos effect occurs when this is no longer satisfied, i.e., close to a local minimizer, a unit step (taking into account the maximal step length due to the fraction-to-the-boundary condition) increases the constraint violation more than it decreases the barrier function value. Thus, the step length would be decreased leading to slower convergence. This is especially relevant for highly nonlinear constraints. A remedy to this problem is given by a so-called second order correction (SOC). This is motivated by the observation that the increase in the constraint violation is mainly due to the error induced by the linearized constraints $\tilde{\boldsymbol{g}}_k + \tilde{A}_k \boldsymbol{d}_\mathrm{p} = \boldsymbol{0}$, which are unable to capture any curvature information. Thus, a SOC step $\boldsymbol{d}_\mathrm{soc}$ aims to take into account this information by means of a Taylor polynomial approximation at the current (new) iterate $\boldsymbol{z}_k + \alpha_\mathrm{p}^+ \boldsymbol{d}_\mathrm{p}$, which yields

$$\tilde{\boldsymbol{g}}(\boldsymbol{z}_k + \alpha_\mathrm{p}^+ \boldsymbol{d}_\mathrm{p} + \boldsymbol{d}_\mathrm{soc}) \approx \tilde{\boldsymbol{g}}(\boldsymbol{z}_k + \alpha_\mathrm{p}^+ \boldsymbol{d}_\mathrm{p}) + \tilde{A}(\boldsymbol{z}_k + \alpha_\mathrm{p}^+ \boldsymbol{d}_\mathrm{p}) \boldsymbol{d}_\mathrm{soc} + O(||\boldsymbol{d}_\mathrm{soc}||^2). \tag{6.48}$$

The SOC step should also satisfy $\boldsymbol{d}_\mathrm{soc} = o(||\boldsymbol{d}_\mathrm{p})||)$, i.e., it should not alter the original step too much. Requiring that

$$\tilde{\boldsymbol{g}}(\boldsymbol{z}_k + \alpha_\mathrm{p}^+ \boldsymbol{d}_\mathrm{p}) + \tilde{A}(\boldsymbol{z}_k + \alpha_\mathrm{p}^+ \boldsymbol{d}_\mathrm{p}) \boldsymbol{d}_\mathrm{soc} = \boldsymbol{0} \tag{6.49}$$

results in $\tilde{\boldsymbol{g}}(\boldsymbol{z}_k + \alpha_\mathrm{p}^+ \boldsymbol{d}_\mathrm{p} + \boldsymbol{d}_\mathrm{soc}) = O(||\boldsymbol{d}_\mathrm{soc}||^2)$. One possible solution of (6.49) is the given by the least squared approximation

$$\begin{bmatrix} I & \tilde{A}^\top(\boldsymbol{z}_k + \alpha_\mathrm{p}^+ \boldsymbol{d}_\mathrm{p}) \\ \tilde{A}(\boldsymbol{z}_k + \alpha_\mathrm{p}^+ \boldsymbol{d}_\mathrm{p}) & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_\mathrm{soc} \\ \tilde{\boldsymbol{\lambda}}_\mathrm{soc} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{0} \\ \tilde{\boldsymbol{g}}(\boldsymbol{z}_k + \alpha_\mathrm{p}^+ \boldsymbol{d}_\mathrm{p}) \end{bmatrix}. \tag{6.50}$$

See Conn et al. [30, Sec. 15.3.2.3] and Wächter [153, Sec. 3.4.3] for alternative solutions. Typically, the fraction-to-the-boundary (6.42) is recalculated based on the corrected step $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \alpha_\mathrm{p}^+ \boldsymbol{d}_\mathrm{p} + \boldsymbol{d}_\mathrm{p}$ and is accepted if the Armijo-type condition (6.44) is fulfilled.

6.2.2.5. Hessian Approximations

In the Newton method (6.34), the KKT matrix $\nabla^2 \mathcal{L}_k$ requires knowledge of the Hessian of the Lagrangian, i.e., $\nabla^2_{\boldsymbol{w}\boldsymbol{w}} \mathcal{L}_k$, which is often expensive to compute. For this reason, the Hessian of the Lagrangian is often replaced with an approximation B_k, which results in a Quasi-Newton method. The underlying idea is to only make small changes to update the approximation iteratively in the sense that

$$B_{k+1} = \operatorname{argmin}_B \|B - B_k\| \tag{6.51a}$$

$$\text{s.t.}$$

$$B = B^\top \tag{6.51b}$$

$$B\boldsymbol{d}_w = \boldsymbol{y}_k \tag{6.51c}$$

where $\boldsymbol{d}_w = \boldsymbol{w}_{k+1} - \boldsymbol{w}_k$, and

$$\boldsymbol{y}_k = \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}_{k+1}, \boldsymbol{s}_{k+1}, \boldsymbol{\lambda}_{k+1}, \boldsymbol{\sigma}_{k+1}) - \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}_k, \boldsymbol{s}_{k+1}, \boldsymbol{\lambda}_{k+1}, \boldsymbol{\sigma}_{k+1}). \tag{6.52}$$

Depending on the norm chosen in (6.51), different Hessian approximations are obtained such as the DFP and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update. Another requirement is that B_{k+1} must be positive definite, which is ensured by the so-called curvature condition

$$\boldsymbol{d}_w^\top \boldsymbol{y}_k > 0, \tag{6.53}$$

that must hold for a positive definite matrix B_{k+1} and that can be verified by pre-multiplying the secant equation (6.51c) with $\boldsymbol{d}_w^\top$. The most commonly used Hessian approximation is the BFGS update, which is given by

$$B_{k+1} = B_k - \frac{B_k \boldsymbol{d}_w \boldsymbol{d}_w^\top B_k}{\boldsymbol{d}_w B_k \boldsymbol{d}_w^\top} + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^\top}{\boldsymbol{y}_k^\top \boldsymbol{d}_w}. \tag{6.54}$$

Note that the inverse $H_{k+1} = B_{k+1}^{-1}$ can be updated using the matrix inversion lemma [see 119, Sec. 6.1]. However, the curvature condition may not always be satisfied when BFGS updating is applied to constrained optimization problems. For this reason, a damped BFGS updated is proposed in Nocedal and Wright [119, Sec. 18.3], where $\boldsymbol{y}_k$ is replaced in (6.54) by $\boldsymbol{r}_k = \theta_k \boldsymbol{y}_k + (1 - \theta_k) B_k \boldsymbol{d}_w$ and

$$\theta_k = \begin{cases} 1 & \text{if } \boldsymbol{d}_w^\top \boldsymbol{y}_k \geq 0.2 \boldsymbol{d}_w^\top B_k \boldsymbol{d}_w, \\ \frac{0.8 \boldsymbol{d}_w^\top B_k \boldsymbol{d}_w}{\boldsymbol{d}_w^\top B_k \boldsymbol{d}_w - \boldsymbol{d}_w^\top \boldsymbol{y}_k} & \text{otherwise.} \end{cases} \tag{6.55}$$

This is motivated by the fact that when $\theta_k = 0 \implies B_{k+1} = B_k$ and $\theta_k = 1$ gives the unmodified, possibly indefinite BFGS update (6.54). Therefore, the choice of $\boldsymbol{r}_k$ and θ_k therein interpolates between the possibly indefinite update and keeping the approximation unchanged.

6.2.2.6. Stopping Criterion

An important aspect of numerical methods is the choice of a suitable stopping criterion. Convergence is typically assessed based on optimality, complementarity, and feasibility. In other words, optimality is achieved if

$$\left\| \nabla_{\boldsymbol{w}} c_k + (\nabla_{\boldsymbol{w}} \boldsymbol{g}_k)^\top \boldsymbol{\lambda}_k + (\nabla_{\boldsymbol{w}} \boldsymbol{h}_k)^\top \boldsymbol{\sigma}_k \right\|_\infty \leq \epsilon_{\text{opt}} \tag{6.56}$$

where ϵ_{opt} is a user-defined optimality tolerance. Complementarity is obtained if

$$||S_k \boldsymbol{\sigma}_k - \mu \boldsymbol{e}_k||_\infty \leq \epsilon_{\text{opt}} \tag{6.57}$$

Feasibility is assessed by checking if the constraints are satisfied, i.e., if

$$\left\| \begin{bmatrix} \boldsymbol{g}_k \\ \boldsymbol{h}_k + \boldsymbol{s}_k \end{bmatrix} \right\|_\infty \leq \epsilon_{\text{con}}, \tag{6.58}$$

where ϵ_{con} is a user-defined constraint tolerance. Based on this, convergence is achieved if optimality, complementarity and feasibility are given w.r.t. the user-defined tolerances. See, e.g., Waltz et al. [154, Sec. 3.4.], for details on the stopping criterion assessment.

6.2.2.7. Example

The presented algorithm is illustrated by means of two examples in Appendix A.1. The first example consists of a hanging chain as presented by Bonnans [19], where the potential energy of the chain is minimized. This results in a static optimization problem where the cost function is subject to nonlinear equality constraints that ensure that the length of each individual chain element is taken into account. Furthermore, linear inequality constraints ensure that the chain is above or on the user-defined floor. The solution is obtained using the presented algorithm and is compared to the solution obtained with `fmincon`, which is a state-of-the-art SQP solver. In this example, the proposed algorithm is able to find a feasible solution with a lower cost than `fmincon`.

In the second example a swing-up of an inverted pendulum on a cart is considered. This problem is modeled as an OCP that is solved using a direct multiple shooting method that, in turn, results in a large-scale NLP. The cost to be minimized is the energy required to move pendulum. The problem's (nonlinear) equality constraints ensure that the pendulum dynamics are satisfied. Linear inequality constraints take into account the input bounds, and linear equality constraints ensure that the boundary conditions are satisfied. The solution obtained with the proposed algorithm is compared to the solution obtained using an indirect collocation method. For different final times of the problem, the proposed algorithm is able to find a feasible solution with a lower cost than the indirect collocation method.

6.3. Optimal and Model Predictive Control

In this section, the theoretical basis of OCPs is discussed along with approaches to obtain numerical solutions for this type of problem. The numerical solution methods can be categorized into indirect and direct methods, where the former attempts to iteratively and approximately solve the necessary conditions for optimality of the OCP and the latter discretizes the original OCP in time to obtain a large-scale NLP that can be solved with methods discussed in Sec. 6.2. Moreover, the OCP is extended to a MPC problem to achieve closed-loop control. Therein, similar variants of an OCP are solved on a fixed time horizon repeatedly. Finally, efficient handling of obstacle constraints in the underlying OCPs is discussed in detail.

6.3.1. Problem Formulation

OCPs aim to minimize a cost functional $J : \mathbb{R}^{n_u} \mapsto \mathbb{R}$ subject to the system dynamics, terminal constraints and general state and input constraints. This is expressed mathematically as

$$
\begin{aligned}
&\min_{\boldsymbol{u}} J(\boldsymbol{u}) = \varphi(t_\mathrm{f}, \boldsymbol{x}(t_\mathrm{f})) + \int_{t_0}^{t_\mathrm{f}} l(\boldsymbol{x}(t), \boldsymbol{u}(t)) \mathrm{d}t && \text{(6.59a)} \\
&\text{s.t.} && \\
&\quad \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \qquad t > t_0, \quad \boldsymbol{x}(t_0) = \hat{\boldsymbol{x}}_0 && \text{(6.59b)} \\
&\quad \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) \leq \boldsymbol{0} && \text{(6.59c)} \\
&\quad \boldsymbol{\psi}(t_\mathrm{f}, \boldsymbol{x}(t_\mathrm{f})) = \boldsymbol{0} && \text{(6.59d)} \\
&\quad \boldsymbol{\kappa}(t_\mathrm{f}, \boldsymbol{x}(t_\mathrm{f})) \leq \boldsymbol{0}, && \text{(6.59e)}
\end{aligned}
$$

where $\varphi : \mathbb{R} \times \mathbb{R}^{n_x} \mapsto \mathbb{R}$ is the terminal cost, $l : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \mapsto \mathbb{R}$ is the running cost, $\boldsymbol{\psi} : \mathbb{R} \times \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_\psi}$, $\boldsymbol{\kappa} : \mathbb{R} \times \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_\kappa}$ are terminal equality and inequality constraints, respectively. Note that, if present, the terminal constraints (6.59d),(6.59e) imply that $\boldsymbol{x}(t_\mathrm{f}) \in \mathcal{X}_\mathrm{f}$, i.e., the final state is constrained to a terminal set $\mathcal{X}_\mathrm{f}$. The path constraints (6.59c) are used to express state and/or input constraints, i.e., they imply that $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{u} \in \mathcal{U}$, respectively, where $\mathcal{X}$ is the set of feasible states and $\mathcal{U}$ is the set of admissible inputs. This also implies that not all components of (6.59c) are assumed to explicitly depend on the input $\boldsymbol{u}$, i.e., there may be components $h_j(\boldsymbol{x}, \boldsymbol{u}) = h_j(\boldsymbol{x}) \leq 0$ and *vice versa* $h_i(\boldsymbol{x}, \boldsymbol{u}) = h_i(\boldsymbol{u}) \leq 0, \ i \neq j$. The latter is used to realize input bound and rate constraints.

Assuming that an optimal solution exists, it is denoted using $\boldsymbol{u}^*(t)$ and the resulting optimal state trajectory is denoted with $\boldsymbol{x}^*(t)$.

Similar to the static optimization case, the first order necessary optimality conditions of (6.59) are formulated by following, e.g., [85, 28, 69] where the fundamental idea is that for a candidate optimal input the first variation (or Gâteaux derivative of $J(\boldsymbol{u})$) δJ must vanish. This leads to the Euler-Lagrange equations, whose derivation can be found, e.g., in Kirk [85], Hartl et al. [69]. They are summarized here in terms of the Hamiltonian associated with (6.59), i.e.,

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = l(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \tag{6.60}$$

where $\boldsymbol{\lambda}$ is called the adjoint state, the Lagrangian

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathcal{H}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) + \boldsymbol{\mu}^\top \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}), \tag{6.61}$$

for some Lagrange multiplier $\boldsymbol{\mu}$, and the auxiliary function

$$\Phi(t_\mathrm{f}, \boldsymbol{x}(t_\mathrm{f})) = \varphi(t_\mathrm{f}, \boldsymbol{x}(t_\mathrm{f})) + (\boldsymbol{\nu}^*)^\top \boldsymbol{\psi}(t_\mathrm{f}, \boldsymbol{x}(t_\mathrm{f})) + (\boldsymbol{\zeta}^*)^\top \boldsymbol{\kappa}(t_\mathrm{f}, \boldsymbol{x}(t_\mathrm{f})), \tag{6.62}$$

with additional (time-invariant) Lagrange multipliers $\boldsymbol{\nu}^*$, $\boldsymbol{\zeta}^*$ in the compact form

$$\begin{aligned}
\dot{\boldsymbol{x}}^* &= \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*), & \boldsymbol{x}(0) &= \hat{\boldsymbol{x}}_0, & (6.63\text{a})\\
\dot{\boldsymbol{\lambda}}^* &= -\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*), & \boldsymbol{\lambda}^*(t_\mathrm{f}) &= \nabla_{\boldsymbol{x}(t_\mathrm{f})} \Phi(t_\mathrm{f}, \boldsymbol{x}^*(t_\mathrm{f})), & (6.63\text{b})\\
\boldsymbol{0} &= \nabla_{\boldsymbol{u}} \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*). & & & (6.63\text{c})
\end{aligned}$$

Furthermore, it must hold that

$$\mathcal{H}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*) \le \mathcal{H}(\boldsymbol{x}^*, \boldsymbol{v}, \boldsymbol{\lambda}^*), \quad \forall \boldsymbol{v} \in \mathcal{U}, \tag{6.64}$$

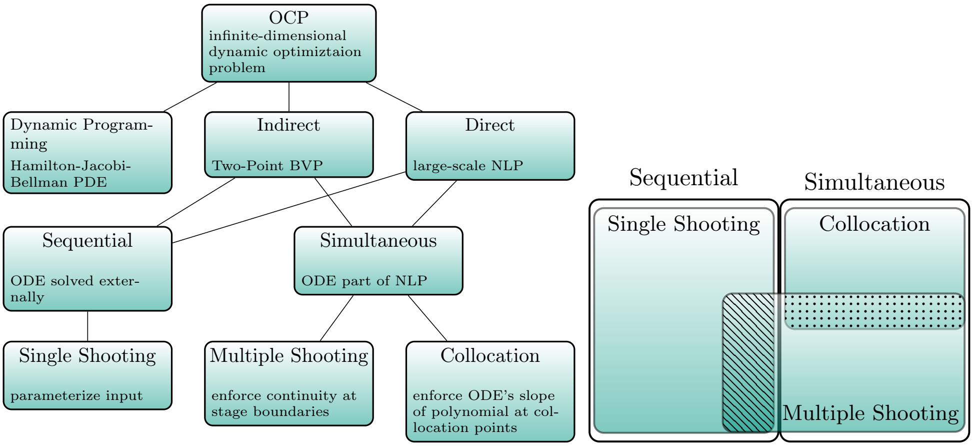
i.e., the Hamiltonian is minimized w.r.t. $\boldsymbol{u}$ over the set of all admissible inputs,

$$\begin{aligned}
(\boldsymbol{\mu}^*)^\top \boldsymbol{h}(\boldsymbol{x}^*, \boldsymbol{u}^*) &= 0 & (6.65\text{a})\\
\boldsymbol{\mu}^* &\ge \boldsymbol{0} & (6.65\text{b})\\
(\boldsymbol{\zeta}^*)^\top \boldsymbol{\kappa}(t_\mathrm{f}, \boldsymbol{x}^*(t_\mathrm{f})) &= 0 & (6.65\text{c})\\
\boldsymbol{\zeta}^* &\ge \boldsymbol{0}, & (6.65\text{d})
\end{aligned}$$

which can be understood as a complementary slackness condition similar to (6.26b), and

$$\boldsymbol{\psi}(t_\mathrm{f}, \boldsymbol{x}^*(t_\mathrm{f})) = \boldsymbol{0} \tag{6.66}$$

needs to be fulfilled. This set of equations constitutes a two-point boundary value problem and is, in general, very hard to solve. This can be attributed to several factors. First, the nonlinearity that is often induced by the system dynamics and/or inequality constraints greatly complicates the process of finding an analytical solution, which is usually only possible for a few special cases, e.g., when the system has linear dynamics or the problem



(a) Structural overview of numerical solution methods of OCPs. Note that indirect and direct methods can both be solved with the same algorithms. Dynamic programming relies on the principle of optimality, which involves solving the Hamilton-Jacobi-Bellman equations, which constitute a partial differential equation (PDE). Pontryagin's Maximum Principle is a special case of the Hamilton-Jacobi-Bellman equations.

(b) Venn diagram of sequential and simultaneous methods. Overlap between collocation and multiple shooting methods (dotted area) occurs when the collocation points coincide with the stage boundaries. Overlap between single and multiple shooting (diagonal lines) is given when the number of stages in the multiple shooting method is equal to one.

Figure 6.2.: Overview of solution approaches to OCPs. On the left, a structured overview is given. See also appendix A.4 for further insights into that topic. On the right, a Venn diagram is shown highlighting similarities.

at hand solely introduces bound constraints on the input. Second, pure state inequality-constrained optimization problems introduce an additional layer of complexity since the constraints need to be differentiated in time repeatedly until the input appears explicitly in one of the time derivatives (and, thus, the degree of the respective constraint similar to Definition 4 is motivated). Finally, the quality of and convergence to a numerical solution is highly dependent on the provided initial guess, which is often not known or hard to obtain. Usually, large efforts are made for two-point boundary value solvers to generate a feasible initial guess, see, e.g., Kierzenka and Shampine [82] for a detailed discussion. Any attempt to solve the necessary optimality conditions is categorized as an indirect method. For the above-mentioned reasons, the arising difficulties lead to the fact that the majority of numerical solutions of (6.59) rely on so-called direct methods, which are discussed in the following.

6.3.2. Numerical Solution

Alternatively to solving the two-point boundary value problem associated with the first order necessary optimality conditions of (6.59), which would result in an indirect method, a direct method discretizes the OCP in time using $N \in \mathbb{N}$ discretization steps and evaluates the constraints and costs in (6.59) at the discrete time steps

$$t_k = t_0 + k\Delta t, \quad k = 0, \dots, N, \quad \Delta t = \frac{t_{\mathrm{hor}}}{N} \tag{6.67}$$

where Δt is the discretization time step. In this way a large scale NLP of the form (6.25) is obtained that can be solved using methods presented in, e.g., Sec. 6.2.2. To achieve this, the infinite-dimensional control input is discretized in time using

$$\boldsymbol{u}(t) = \boldsymbol{U}^{(k)}(t, \boldsymbol{w}_u^{(k)}), \quad t_{k-1} \le t \le t_k, \tag{6.68}$$

for each stage $[t_{k-1}, t_k],\ k = 1, \dots, N$, e.g., using a piece-wise linear parameterization[2]

$$\boldsymbol{u}(t) = \boldsymbol{U}^{(k)}(t, \boldsymbol{w}_u^{(k)}) = \boldsymbol{u}_{k-1}^{(k)} + \frac{t - t_{k-1}}{t_k - t_{k-1}}(\boldsymbol{u}_k^{(k)} - \boldsymbol{u}_{k-1}^{(k)}), \quad t_{k-1} \le t \le t_k, \tag{6.69}$$

where $\boldsymbol{w}_u^{(k)} = [(\boldsymbol{u}_{k-1}^{(k)})^\top, (\boldsymbol{u}_k^{(k)})^\top]^\top$ and $\boldsymbol{w}_u = [(\boldsymbol{w}_u^{(1)})^\top, \dots, (\boldsymbol{w}_u^{(N)})^\top]^\top,\ k = 1, \dots, N$ are a finite set of decision variables. Note that a piece-wise linear continuous input can be realized by enforcing $\boldsymbol{u}_k^{(k)} = \boldsymbol{u}_{k-1}^{(k+1)}$. With this, the set of input decision variables may be reordered (and reduced) into $\boldsymbol{w}_u = [\boldsymbol{u}_0^\top, \dots, \boldsymbol{u}_N^\top]^\top \in \mathbb{R}^{n_{w_u}}$, where $n_{w_u} = (N+1)n_u$ and $\boldsymbol{u}_k = \boldsymbol{u}(t_k)$. With this, depending on how the ODE constraint is handled, one can further split direct methods into sequential and simultaneous methods. In the former, which is also known as single shooting or control vector parameterization, only the input is parameterized and the solution of the ODE is not part of the OCP, i.e., it is handled by an external solver to obtain the state trajectory. Thus, each time the input parameter weights $\boldsymbol{w}_u^k,\ k = 1, \dots, N$ are altered by the algorithm, the ODE needs to be re-solved to evaluate the costs and constraints, hence, the name sequential method. In the latter, the solution to the ODE is included to the OCP explicitly and, hence, the name simultaneous methods. To this end, equality constraints are imposed in the resulting NLP to obtain continuity of the individual solutions of the stages over the entire horizon. Note that depending on how the ODE solution of the stages is handled, simultaneous methods are further distinguished into multiple shooting and collocation methods. Multiple shooting follows the same principle as single shooting but solves the ODE externally for each stage $[t_{k-1}, t_k],\ k = 1, \dots, N$ instead of the entire interval $[t_i, t_{\mathrm{f}}]$. Subsequently the respective initial conditions of the stages are added as additional decision variables to the NLP. Collocation methods rely on a polynomial parameterization of the states and typically solve the ODE in the individual stages by further dividing the stages into collocation points. The polynomial parameters are found by ensuring that the time derivative of the polynomial coincides with the theoretical value given by the right-hand-side of the ODE. Continuity w.r.t. the stage boundaries is again enforced with equality constraints in the NLP. An attempt to visualize the different solution approaches is given in Fig. 6.2.

[2] See, e.g., Chachuat [28, Sec. 5.4.1] for a detailed discussion of different polynomial approximations.

Remark 4 (Ambiguity of Classification) *Often ambiguity w.r.t. the classification of numerical solution methods arises. For example, in the context of multiple shooting if the external solver is a one-step explicit method (such as the explicit Euler) for each stage, this results in what is typically understood as a full discretization [28, Sec. 5.4.2], where the set of continuity constraints essentially realizes an N-step integration of the entire horizon with this method. On the other hand, if each stage is handled by, e.g., a step-size controlled Runge-Kutta method of higher order, this would not be understood as a classical full discretization and, furthermore, obtaining an analytic expression for the respective gradients could only be done numerically or, alternatively, relies on the capability of the underlying Runge-Kutta implementation to return this information.*

Furthermore, note that depending on the choice of the integration method used in multiple shooting it may be equivalent to a direct collocation since every collocation method corresponds to a Runge-Kutta method (but not vise-versa) with coefficients

$$a_{ij} = \int_0^{c_i} l_j(\tau)\,d\tau, \qquad b_i = \int_0^1 l_i(\tau)\,d\tau, \tag{6.70}$$

where $l_i(\tau) = \Pi_{l \neq i}(\tau - c_i)/(c_i - c_l)$ is the Lagrange polynomial (not to be confused with the stage cost) and a_{ij}, b_i, c_i are entries of the Butcher tableau of the corresponding Runge-Kutta method [65, Sec. II.1.2]. See also Fig. 6.2 for an overview of numerical solution methods of OCPs. Prevalent examples of collocation methods are the explicit and implicit Euler methods, the mid-point rule, and the trapezoidal method.

In the following, focus is put on direct multiple shooting using the piece-wise linear continuous input parameterization (6.69) and one-step ODE integration methods of the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}), \tag{6.71}$$

where $\boldsymbol{F}(\cdot)$ is known as the discrete propagation function that is given, e.g., by $\boldsymbol{F}(\cdot) = \boldsymbol{x}_k + \frac{t_\mathrm{f}}{N}\boldsymbol{f}(\boldsymbol{x}_{k+1}, \boldsymbol{u}_{k+1})$ for the implicit Euler, and[3] $\boldsymbol{w}_x = [\boldsymbol{x}_0^\top, \boldsymbol{x}_1^\top, \dots, \boldsymbol{x}_N^\top]^\top \in \mathbb{R}^{(N+1)n_x}$ is the vector of initial conditions for the shooting intervals or, in other words, the vector of state decision variables. With this, the cost functional is approximated numerically using a (possibly different compared with the ODE) numerical quadrature rule, e.g., the trapezoidal rule, i.e.,

$$J(\boldsymbol{u}) \approx c(\boldsymbol{w}) = \varphi(t_N, \boldsymbol{x}_N) + \frac{t_\mathrm{f}}{2N} \sum_{k=0}^{N-1l} \big(l_k + l_{k+1}\big), \tag{6.72}$$

[3] Note that strictly speaking $\boldsymbol{x}_0$ is not a decision variable but is added here to avoid confusion.

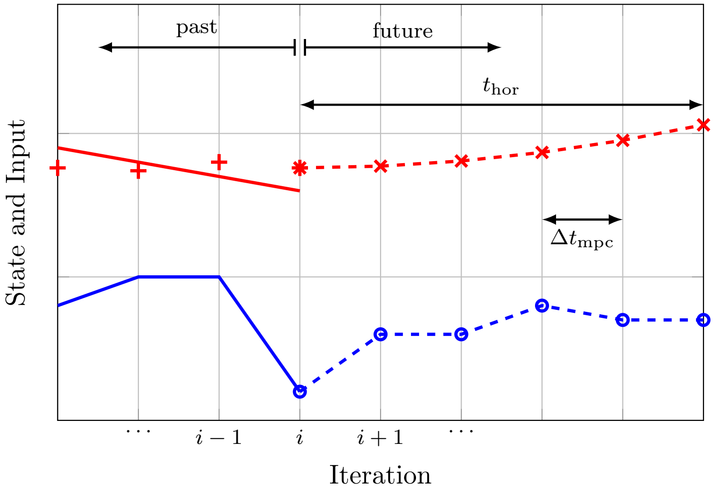


Figure 6.3.: Illustration of the MPC approach with the true state (solid red), estimated state (red pluses), predicted states (red crosses) with interpolated values (dashed red), applied control (solid blue), predicted inputs (blue circles) and piece-wise linear interpolation (dashed blue), prediction horizon t_{hor} and MPC sample time Δt_{mpc}.

where $l_k = l(\boldsymbol{x}_k, \boldsymbol{u}_k)$, and $\boldsymbol{w}^\top = [\boldsymbol{w}_x^\top, \boldsymbol{w}_u^\top]^\top$ is the vector of total decision variables. Finally, the infinite-dimensional OCP (6.59) is approximated as a static constrained and, in general, large-scale nonlinear program of the form as in (6.25), i.e.,

$$\min_{\boldsymbol{w}} c(\boldsymbol{w}) = \varphi(t_N, \boldsymbol{x}_N) + \frac{t_{\mathrm{f}}}{2N} \sum_{k=0}^{N-1} (l_{k+1} + l_k), \tag{6.73a}$$

s.t.

$$\begin{aligned}
&\boldsymbol{x}_0 = \hat{\boldsymbol{x}}_i && && (6.73\mathrm{b})\\
&\boldsymbol{x}_{k+1} = \boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}), && k = 0, \ldots, N-1 && (6.73\mathrm{c})\\
&\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq \boldsymbol{0}, && k = 0, \ldots, N && (6.73\mathrm{d})\\
&\boldsymbol{\psi}(t_N, \boldsymbol{x}_N) = \boldsymbol{0} && && (6.73\mathrm{e})\\
&\boldsymbol{\kappa}(t_N, \boldsymbol{x}_N) \leq \boldsymbol{0}, && && (6.73\mathrm{f})
\end{aligned}$$

which can be solved numerically using, e.g., the interior-point method presented in Sec.6.2.2. Note that there exist interesting connections between the solution of the NLP (6.73) and the solution to the Euler-Lagrange equations, which are further discussed, e.g., in Betts [15, Sec. 4.2]. Furthermore, several strategies and insights into the structure of the resulting NLP can be exploited to further reduce the complexity of the numerical solution, which is also discussed extensively in Betts [15].

6.3.3. Model Predictive Control

In contrast to OCPs, which are solved once for the specified horizon and, therefore, yield an open loop optimal control input, MPC relies on repeatedly solving OCPs (with usually shorter horizon) online to take into account measurement information. The optimal solution is applied for a specific time interval and then the OCP is solved again using any new available information.

$$\min_{\boldsymbol{u}\in\mathcal{U}} J(\boldsymbol{u}) = \varphi(T_i, \boldsymbol{x}(T_i)) + \int_{t_i}^{T_i} l(\boldsymbol{x}(t), \boldsymbol{u}(t))\,\mathrm{d}t \tag{6.74a}$$

s.t.

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \qquad t > t_i, \quad \boldsymbol{x}(t_i) = \hat{\boldsymbol{x}}_i \tag{6.74b}$$

$$\boldsymbol{x} \in \mathcal{X} \tag{6.74c}$$

$$\boldsymbol{x}(T_i) \in \mathcal{X}_\mathrm{f} \tag{6.74d}$$

for $t \in [t_i, T_i]$ with $T_i = t_i + t_\mathrm{hor}$, where t_i is the current iteration time and t_hor is the prediction horizon. Furthermore, $\hat{\boldsymbol{x}}_i$ is the state estimate corresponding to t_i, $\mathcal{X} \subset \mathbb{R}^{n_x}$ is the state constraint set, $\mathcal{U} \subset \mathbb{R}^{n_u}$ defines the set of feasible inputs, and $\mathcal{X}_\mathrm{f}$ is a terminal region. Note that (6.74) is a finite horizon OCP, i.e., similar to (6.59) with abbreviated state and input constraints using the sets $\mathcal{X}$, $\mathcal{U}$, respectively, for notational convenience. See Fig. 6.3 for an illustration of the MPC approach. In the context of OCPs, it is well-known that the solution to an infinite horizon OCP is stabilizing if it exists see, e.g., Mayne et al. [107]. For MPC problems, however, the horizon is finite and one can, in general, not expect that the closed-loop MPC solution is equivalent to the stabilizing infinite horizon OCP solution and the stability of the MPC depends on the specific choice of design parameters, i.e., the finite horizon length t_hor, the terminal set $\mathcal{X}_\mathrm{f}$, and running and terminal costs $l(\cdot)$, $\varphi(\cdot)$, respectively. For the sake of completeness, these conditions are stated in the following, which hold under mild assumptions on the right-hand-side of the ODE see, e.g., Fontes [46].

7. Collision Avoidance

In this chapter a set of constraints is derived that can be used to achieve collision avoidance in the context of MPC. In the context of MPC, collision avoidance for MSVs is formulated using the set of feasible states $\mathcal{X}$. Based on the MSV system dynamics of the form (6.74b) collision avoidance constraints can be derived that are embedded into the OCP and MPC problems to realize collision-free trajectories. To achieve collision-free trajectories, the input $\boldsymbol{u}$ in (6.74) must be calculated in a way so that the controlled object represented using the set $\mathcal{V}(\boldsymbol{\eta}) \subset \mathbb{R}^2$ does not collide with an obstacle represented using the set $\mathcal{O} \subset \mathbb{R}^2$. The state $\boldsymbol{x}$ contains the MSV's pose $\boldsymbol{\eta} = [\boldsymbol{p}^{\mathcal{V}}(t)^\top, \psi^{\mathcal{V}}(t)]^\top$, which in turn consists of the vessel's position $\boldsymbol{p}^{\mathcal{V}}(t) = [x(t), y(t)]^\top \in \mathbb{R}^2$ and its heading (w.r.t. the x_i-axis) denoted using $\psi^{\mathcal{V}}(t) \in \mathbb{R}$. Based on this, collision avoidance is expressed mathematically as

$$\mathcal{V}(\boldsymbol{\eta}) \cap \mathcal{O} = \emptyset, \tag{7.1}$$

which must hold for the entire prediction horizon in the MPC problem. Since $\mathcal{V}(\boldsymbol{\eta})$ depends on the pose it is time-varying due to (6.59b). For the sake of clarity the time dependence of $\mathcal{O}$ is omitted. Subsequently, two cases are distinguished to formulate suitable collision avoidance constraints:

- Singleton case, i.e., $\tilde{\mathcal{V}} = \{\boldsymbol{0}\}$.
- Full body case, i.e, $\tilde{\mathcal{V}}$ is nonempty and convex.

In the following, based on the general collision avoidance condition (7.1) and the assumed shape of the controlled object three types of dual collision avoidance constraints are derived using Farkas' Lemma, the distance, and signed distance between two convex sets. The latter two make explicit use of Lagrange multipliers and the principle of duality in order to reformulate the arising primal problems [20] while the former is also known to have a strong connection to the concept of duality for linear programs and proving first order necessary optimality conditions [119]. These dual reformulations are motivated by the fact that in the context of MPC the general condition (7.1) is hard to implement and often non-differentiable. Due to the assumption of convex shapes the proposed reformulations are exact and, furthermore, constitute a set of constraints that can easily be included to the OCPs.

The presented concepts are published in a shorter form in Helling and Meurer [168].

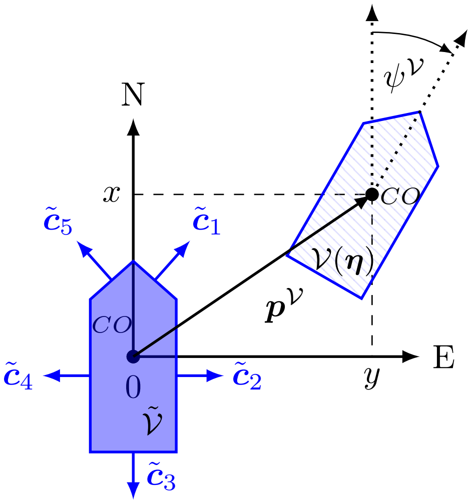


Figure 7.1.: Illustration of the base shape $\tilde{\mathcal{V}}$ (solid blue) and the affine transformation $\mathcal{V}(\boldsymbol{\eta})$ of the controlled object (hatched blue).

7.1. Geometric Ship Model

The space occupied by the MSV is denoted using the set $\mathcal{V}(\boldsymbol{\eta}) \subset \mathbb{R}^2$ and is described mathematically by means of the affine mapping

$$\mathcal{V}(\boldsymbol{\eta}) = R_{z,\psi}\tilde{\mathcal{V}} + \{\boldsymbol{p}^{\mathcal{V}}\}, \tag{7.2}$$

where $\tilde{\mathcal{V}} \subset \mathbb{R}^2$ is the convex set describing the base shape of the controlled object as illustrated in Fig. (7.2). Here, a notation similar to Gilbert and Johnson [59] is used, e.g., (7.2) denotes a matrix multiplication with every element in $\tilde{\mathcal{V}}$ to which the element $\{\boldsymbol{p}^{\mathcal{V}}\}$ added. In the full body case the controlled object is assumed to be a convex polyhedron[1], which is expressed using a linear matrix inequality such that

$$\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p} \in \mathbb{R}^n : C(\boldsymbol{\eta})\boldsymbol{p} \le \boldsymbol{d}(\boldsymbol{\eta})\}, \tag{7.3}$$

where $C(\boldsymbol{\eta}) = [\boldsymbol{c}_1(\boldsymbol{\eta}), \dots, \boldsymbol{c}_L(\boldsymbol{\eta})]^\top \in \mathbb{R}^{L\times 2}$ is a matrix collecting the normal vectors of the edges of the controlled object, and $\boldsymbol{d}(\boldsymbol{\eta}) \in \mathbb{R}^L$ is a vector that includes the x_i-axis offsets and L is the number of edges. For the sake of clarity the state-dependence of $C(\cdot), \boldsymbol{d}(\cdot)$ is not denoted explicitly unless helpful in the context. Moreover, $\boldsymbol{c}_j^\top$ and d_j denote the j-th row vector in C and the j-th entry in $\boldsymbol{d}$, respectively. Likewise, the space occupied by the obstacle is represented using the set

$$\mathcal{O} = \{\boldsymbol{p} \in \mathbb{R}^n : A\boldsymbol{p} \le \boldsymbol{b}\}, \tag{7.4}$$

[1] It is also possible to describe nonconvex geometries with this assumption by approximating the nonconvex space as the union of multiple convex polyhedra, i.e., $\mathcal{V} \approx \bigcup_i \bar{\mathcal{V}}_i$, where $\bar{\mathcal{V}}_i = \{\boldsymbol{p} \in \mathbb{R}^2, : \bar{C}_i\boldsymbol{p} \le \bar{\boldsymbol{d}}_i\}$.

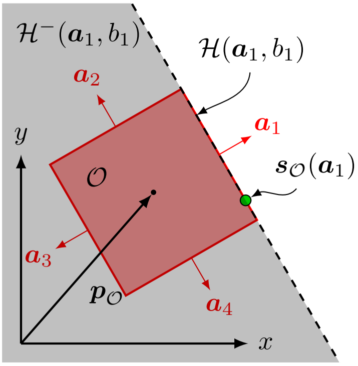

$\mathcal{H}^-(\boldsymbol{a}_1, b_1)$
$\mathcal{H}(\boldsymbol{a}_1, b_1)$
$\boldsymbol{a}_2$
$\boldsymbol{a}_1$
y
$\mathcal{O}$
$\boldsymbol{s}_{\mathcal{O}}(\boldsymbol{a}_1)$
$\boldsymbol{a}_3$
$\boldsymbol{p}_{\mathcal{O}}$
$\boldsymbol{a}_4$
x

Figure 7.2.: Visualization of the underlying concepts of convex analysis, where $\mathcal{O}$ is defined as a square (red area) with outward normal vectors $\boldsymbol{a}_j$, $j = 1, \dots, \check{L}$ and $\check{L} = 4$, $\mathcal{H}^-(\boldsymbol{a}_1, b_1)$ is a half-space (shaded gray area) such that $\mathcal{O} \subset \mathcal{H}^-(\boldsymbol{a}_1, b_1)$, $\mathcal{H}(\boldsymbol{a}_1, b_1)$ is the corresponding (supporting) hyperplane (dashed black line), and $\boldsymbol{s}_{\mathcal{O}}(\boldsymbol{a}_1)$ is a support point (green circle), which, in this case, is not unique.

where $A = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_{\check{L}}]^\top \in \mathbb{R}^{\check{L} \times n}$ and $\boldsymbol{b} \in \mathbb{R}^{\check{L}}$. Similar to the controlled object $\mathcal{V}(\boldsymbol{\eta})$, $\boldsymbol{a}_j^\top$ and b_j denote the j-th row vector in A and the j-th entry in $\boldsymbol{b}$, respectively.

An important property of convex polyhedra is that the obstacle can be viewed as the intersection of all $j = 1, \dots, \check{L}$ (closed lower) half-spaces that contain it, i.e., $\mathcal{O} = \bigcap_j \mathcal{H}^-(\boldsymbol{a}_j, b_j)$, where

$$\mathcal{H}^-(\boldsymbol{a}_j, b_j) = \{\boldsymbol{p} \in \mathbb{R}^n : \boldsymbol{a}_j^\top \boldsymbol{p} \leq b_j\} \tag{7.5}$$

is the half-space with outward normal vector $\boldsymbol{a}_j$, i.e., the j-th face of the convex polyhedron of $\mathcal{O}$. If $||\boldsymbol{a}_j||_2 = 1$, then b_j is the signed distance of the hyperplane from the coordinate origin, i.e.,

$$\mathrm{sd}(\boldsymbol{0}, \mathcal{H}(\boldsymbol{a}_j, b_j)) = b_j, \quad ||\boldsymbol{a}_j||_2 = 1 \tag{7.6}$$

to be defined later and where $\mathcal{H}(\boldsymbol{a}_j, b_j) = \{\boldsymbol{p} \in \mathbb{R}^n : \boldsymbol{a}_j^\top \boldsymbol{p} = b_j\}$ denotes a hyperplane, which is intrinsically connected to the half-space $\mathcal{H}^-(\boldsymbol{a}_j, b_j)$. A hyperplane $\mathcal{H}(\cdot, \cdot)$, where $\mathcal{V}(\boldsymbol{\eta}) \subset \mathcal{H}^-(\cdot, \cdot)$ and $\mathcal{O} \not\subset \mathcal{H}^-(\cdot, \cdot)$ is called a separating hyperplane. Furthermore, if $\exists \boldsymbol{p} \in \mathcal{H}(\cdot, \cdot) : \boldsymbol{p} \in \mathcal{V}(\boldsymbol{\eta}) \wedge \mathcal{V}(\boldsymbol{\eta}) \subset \mathcal{H}^-(\cdot, \cdot)$, then $\mathcal{H}(\cdot, \cdot)$ is called a supporting hyperplane of $\mathcal{V}(\boldsymbol{\eta})$. In other words, the supporting hyperplane is tangent to $\mathcal{V}(\boldsymbol{\eta})$ and $\boldsymbol{p}$ lies on the boundary of $\mathcal{V}(\boldsymbol{\eta})$.

Another very important concept is the support mapping

$$\boldsymbol{z}^\top \boldsymbol{s}_{\mathcal{V}}(\boldsymbol{z}) = \sup_{\boldsymbol{p}} \{\boldsymbol{z}^\top \boldsymbol{p} : \boldsymbol{p} \in \mathcal{V}(\boldsymbol{\eta})\} \tag{7.7}$$

of an object $\mathcal{V}(\boldsymbol{\eta})$, which maps any vector $\boldsymbol{z}$ to a point in $\mathcal{V}(\boldsymbol{\eta})$ that gives the maximum inner product and thus defines the support point $\boldsymbol{s}_{\mathcal{V}}(\boldsymbol{z})$ as the point in $\mathcal{V}(\boldsymbol{\eta})$ corresponding to this inner product given $\boldsymbol{z}$. If $\mathcal{V}(\boldsymbol{\eta})$ is convex, the support mapping implicitly describes the shape since it must always lie on its boundary. This is readily exploited in, e.g., the

GJK algorithm [98, 150]. As the name suggests, the support mapping can be used to define a supporting hyperplane of $\mathcal{V}$, i.e., $\mathcal{H}(\boldsymbol{z}, \boldsymbol{z}^\top \boldsymbol{s}_\mathcal{V}(\boldsymbol{z}))$. In this context, it is also observed that $\boldsymbol{s}_{\mathcal{O}-\mathcal{V}(\boldsymbol{\eta})}(\boldsymbol{z}) = \boldsymbol{s}_\mathcal{O}(\boldsymbol{z}) - \boldsymbol{s}_\mathcal{V}(-\boldsymbol{z})$, where

$$\mathcal{O} - \mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p}_1 - \boldsymbol{p}_2 : \boldsymbol{p}_1 \in \mathcal{O}, \boldsymbol{p}_2 \in \mathcal{V}(\boldsymbol{\eta})\} \tag{7.8}$$

is the Minkowski sum[2] [151]. It is also pointed out that if the origin $\boldsymbol{0}$ is contained in the Minkowski sum (7.8), then $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$ intersect [98]. See also Fig. 7.2 for an illustration of an example obstacle shape including the most important principles of convex analysis.

Finally, we state Farkas' Lemma for later reference.

Lemma 1 (Farkas' Lemma [38]) *Let $A \in \mathbb{R}^{n\times m}$ and $\boldsymbol{b} \in \mathbb{R}^n$. Then exactly one of the following sets must be empty:*

(i) $\{\boldsymbol{p} \in \mathbb{R}^m : A\boldsymbol{p} \leq \boldsymbol{b}\}$.

(ii) $\{\boldsymbol{\mu} \in \mathbb{R}^n : A^\top \boldsymbol{\mu} = \boldsymbol{0},\ \boldsymbol{b}^\top \boldsymbol{\mu} < 0,\ \boldsymbol{\mu} \geq \boldsymbol{0}\}$.

7.2. Indicator Constraints

The simplest approach in order to evaluate collision avoidance is by constructing a set of constraints that must be fulfilled in the case of a collision. Collision avoidance is thus proved by showing that no solution exists for the resulting system of equations by means of Lemma 1. Since the same results can be derived using the notion of an indicator function, see, e.g., [20], we call this approach indicator approach. The indicator conditions provide the simplest means to include dual collision avoidance into the MPC problem. This is due to the fact that no norm constraint arises in the conditions, see also Tab. 7.1. This comes at the price of not being able to determine proximity between the controlled object and obstacles.

7.2.1. Singleton Case

Given $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p}^\mathcal{V}\}$ and $\mathcal{O}$ according to (7.4), then (7.1) is equivalent to

$$\begin{bmatrix} A \\ I \\ -I \end{bmatrix} \boldsymbol{p} \leq \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{p}^\mathcal{V} \\ -\boldsymbol{p}^\mathcal{V} \end{bmatrix} \tag{7.9}$$

[2] Note that strictly-speaking, the Minkowski difference is defined as $\mathcal{O} \ominus \mathcal{V}(\boldsymbol{\eta}) = (\mathcal{O}^c + (-\mathcal{V}(\boldsymbol{\eta})))^c$. Therefore, we refer to (7.8) as a sum.

		Collision avoidance	Domain	Non-negativity	Norm	Consistency
Singleton	Indicator	$(A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu} > 0$	–	$\boldsymbol{\mu} \geq \boldsymbol{0}$	–	–
	Distance	$(A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu} \geq d_{\text{safe}}$	$d_{\text{safe}} \in \mathbb{R}^+$	$\boldsymbol{\mu} \geq \boldsymbol{0}$	$\|\|A^\top \boldsymbol{\mu}\|\|_2 \leq 1$	–
	Signed Distance	$(A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu} \geq d_{\text{safe}}$	$d_{\text{safe}} \in \mathbb{R}$	$\boldsymbol{\mu} \geq \boldsymbol{0}$	$\|\|A^\top \boldsymbol{\mu}\|\|_2 = 1$	–
		$\frac{(A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu}}{\|\|A^\top \boldsymbol{\mu}\|\|_2} \geq d_{\text{safe}}$	$d_{\text{safe}} \in \mathbb{R}$	$\boldsymbol{\mu} \geq \boldsymbol{0}$	–	–
Full body	Indicator	$-\boldsymbol{b}^\top \boldsymbol{\mu} - \boldsymbol{d}^\top \boldsymbol{\lambda} > 0$	–	$\boldsymbol{\mu} \geq \boldsymbol{0}, \boldsymbol{\lambda} \geq \boldsymbol{0}$	–	$A^\top \boldsymbol{\mu} + C^\top \boldsymbol{\lambda} = \boldsymbol{0}$
	Distance	$-\boldsymbol{b}^\top \boldsymbol{\mu} - \boldsymbol{d}^\top \boldsymbol{\lambda} \geq d_{\text{safe}}$	$d_{\text{safe}} \in \mathbb{R}^+$	$\boldsymbol{\mu} \geq \boldsymbol{0}, \boldsymbol{\lambda} \geq \boldsymbol{0}$	$\|\|A^\top \boldsymbol{\mu}\|\|_2 \leq 1$	$A^\top \boldsymbol{\mu} + C^\top \boldsymbol{\lambda} = \boldsymbol{0}$
	Signed Distance	$-\boldsymbol{b}^\top \boldsymbol{\mu} - \boldsymbol{d}^\top \boldsymbol{\lambda} \geq d_{\text{safe}}$	$d_{\text{safe}} \in \mathbb{R}$	$\boldsymbol{\mu} \geq \boldsymbol{0}, \boldsymbol{\lambda} \geq \boldsymbol{0}$	$\|\|A^\top \boldsymbol{\mu}\|\|_2 = 1$	$A^\top \boldsymbol{\mu} + C^\top \boldsymbol{\lambda} = \boldsymbol{0}$
		$\frac{-\boldsymbol{b}^\top \boldsymbol{\mu} - \boldsymbol{d}^\top \boldsymbol{\lambda}}{\|\|A^\top \boldsymbol{\mu}\|\|_2} \geq d_{\text{safe}}$	$d_{\text{safe}} \in \mathbb{R}$	$\boldsymbol{\mu} \geq \boldsymbol{0}, \boldsymbol{\lambda} \geq \boldsymbol{0}$	–	$A^\top \boldsymbol{\mu} + C^\top \boldsymbol{\lambda} = \boldsymbol{0}$

Table 7.1.: Summary of dual conditions/constraints of all three approaches for evaluating collision avoidance between $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$ such that[a] $\mathcal{V}(\boldsymbol{\eta}) \cap \mathcal{O} = \emptyset$.

[a] Note that for the signed distance, this depends on d_{safe}. If $d_{\text{safe}} < 0$, overlapping of $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$ is considered as not colliding.

having no solution, i.e., there is no point that is simultaneously element of $\mathcal{O}$ and equal to $\boldsymbol{p}^{\mathcal{V}}$ (expressed using two inequalities). In this case, according to Lemma 1, the (dual) system of equations

$$\begin{bmatrix} A^\top & I^\top & -I^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\sigma}_1 \\ \boldsymbol{\sigma}_2 \end{bmatrix} = \boldsymbol{0} \tag{7.10a}$$

$$\begin{bmatrix} \boldsymbol{b}^\top & (\boldsymbol{p}^{\mathcal{V}})^\top & -(\boldsymbol{p}^{\mathcal{V}})^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\sigma}_1 \\ \boldsymbol{\sigma}_2 \end{bmatrix} < 0, \tag{7.10b}$$

$$\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\sigma}_1 \\ \boldsymbol{\sigma}_2 \end{bmatrix} \geq \boldsymbol{0} \tag{7.10c}$$

must have a solution. Upon further inspection (see Appendix A.2), it becomes clear that (7.10) reduces to

$$(A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu} > 0, \quad \boldsymbol{\mu} \geq \boldsymbol{0}. \tag{7.11}$$

Finally, (7.1) can be expressed as follows using (7.11).

Proposition 1 *Suppose $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p}^{\mathcal{V}}\}$ and $\mathcal{O} = \{\boldsymbol{p} \in \mathbb{R}^n : A\boldsymbol{p} \leq \boldsymbol{b}\}$, then* (7.1) *can be re-formulated as the feasibility problem*

$$\mathcal{V}(\boldsymbol{\eta}) \cap \mathcal{O} = \emptyset \Leftrightarrow \exists \boldsymbol{\mu} \geq \boldsymbol{0} : (A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu} > 0 \tag{7.12}$$

using Lemma 1.

In other words, if $\boldsymbol{p}^{\mathcal{V}} \notin \mathcal{O}$, then $\exists j \in \{1, \dots, \check{L}\} : \boldsymbol{p}^{\mathcal{V}} \notin \mathcal{H}^-(\boldsymbol{a}_j, b_j)$, which can also be expressed as $\exists j \in \{1, \dots, \check{L}\} : (\boldsymbol{a}_j^\top \boldsymbol{p}^{\mathcal{V}} - b_j)\mu_j > 0$. Consequently, there must be a scalar $\mu_j > 0$ such that (7.12) holds.

If $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p}^{\mathcal{V}}\}$, Prop. 1 can be used to formulate the feasible set of states, i.e.,

$$\mathcal{X} = \left\{(\boldsymbol{x}(t), \boldsymbol{\mu}(t)) \in \mathbb{R}^{n_x} \times \mathbb{R}^{\check{L}} : (A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu} \geq \epsilon,\, \boldsymbol{\mu} \geq \boldsymbol{0}\right\}, \tag{7.13}$$

where $\epsilon \in \mathbb{R}^+$ is used to obtain a non-strict inequality constraint. This is reasonable since if $\boldsymbol{p}^{\mathcal{V}} \notin \mathcal{O}$, then $\exists j \in \{1, \dots, \check{L}\} : (\boldsymbol{a}_j^\top \boldsymbol{p}^{\mathcal{V}} - b_j)\mu_j > 0$ and μ_j can be scaled to give $(\boldsymbol{a}_j^\top \boldsymbol{p}^{\mathcal{V}} - b_j)\mu'_j \geq \epsilon$ with $\epsilon \in \mathbb{R}^+$ and $\mu'_j = \gamma\mu_j$, $\gamma > 1$. Since the feasibility problem (7.12) is included by means of (7.13), $\boldsymbol{\mu}$ is an additional decision variable in the MPC.

7.2.2. Full Body Case

Given the convex polyhedron $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$, then (7.1) is equivalent to

$$\begin{bmatrix} A \\ C(\boldsymbol{\eta}) \end{bmatrix} \boldsymbol{p} \leq \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{d}(\boldsymbol{\eta}) \end{bmatrix} \tag{7.14}$$

having no solution. In other words, there is no point $\boldsymbol{p}$ that is simultaneously element of $\mathcal{V}(\boldsymbol{\eta})$ and element of $\mathcal{O}$. Subsequently, according to Lemma 1, the system of equations

$$\begin{bmatrix} A^\top & C^\top(\boldsymbol{\eta}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\lambda} \end{bmatrix} = \mathbf{0} \tag{7.15a}$$

$$\begin{bmatrix} \boldsymbol{b}^\top & \boldsymbol{d}^\top(\boldsymbol{\eta}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\lambda} \end{bmatrix} < 0, \tag{7.15b}$$

$$\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\lambda} \end{bmatrix} \geq \mathbf{0} \tag{7.15c}$$

must have a solution. As a consequence, (7.1) can be summarized as follows using (7.15).

Proposition 2 *Suppose* $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p} \in \mathbb{R}^2 : C(\boldsymbol{\eta})\boldsymbol{p} \leq \boldsymbol{d}(\boldsymbol{\eta})\}$ *and* $\mathcal{O} = \{\boldsymbol{p} \in \mathbb{R}^2 : A\boldsymbol{p} \leq \boldsymbol{b}\}$, *then* (7.1) *can be re-formulated as the feasibility problem*

$$\begin{aligned} &\mathcal{V}(\boldsymbol{\eta}) \cap \mathcal{O} = \emptyset \\ &\Leftrightarrow \exists \boldsymbol{\lambda} \geq \mathbf{0},\ \boldsymbol{\mu} \geq \mathbf{0} : -\boldsymbol{b}^\top \boldsymbol{\mu} - \boldsymbol{d}^\top(\eta)\boldsymbol{\lambda} > 0 \wedge A^\top \boldsymbol{\mu} + C^\top(\boldsymbol{\eta})\boldsymbol{\lambda} = \mathbf{0} \end{aligned} \tag{7.16}$$

using Lemma 1.

If $\mathcal{V}(\boldsymbol{\eta})$ is given as a convex polytope, Prop. 2 can be used to formulate the feasible set of states with

$$\begin{aligned} \mathcal{X} = \big\{&(\boldsymbol{x}(t), \boldsymbol{\mu}(t), \boldsymbol{\lambda}(t)) \in \mathbb{R}^{n_x} \times \mathbb{R}^{\check{L}} \times \mathbb{R}^{L} : \\ &-\boldsymbol{b}^\top \boldsymbol{\mu} - \boldsymbol{d}^\top(\eta)\boldsymbol{\lambda} \geq \epsilon,\ \boldsymbol{\mu} \geq \mathbf{0},\ \boldsymbol{\lambda} \geq \mathbf{0}, A^\top \boldsymbol{\mu} + C^\top(\eta)\boldsymbol{\lambda} = \mathbf{0}\big\}, \end{aligned} \tag{7.17}$$

where $\epsilon \in \mathbb{R}^+$ is used to obtain a non-strict inequality constraint and the same reasoning as in the singleton case can be applied. In contrast to the singleton case, the consistency constraint $A^\top \boldsymbol{\mu} + C^\top(\boldsymbol{x})\boldsymbol{\lambda} = \mathbf{0}$ needs to be taken into account additionally. Note that the state-dependence of $C(\boldsymbol{x})$ and $\boldsymbol{d}(\boldsymbol{x})$ is denoted explicitly to avoid confusion. Since the feasibility problem (7.16) is included by means of (7.17), $\boldsymbol{\mu}, \boldsymbol{\lambda}$ are additional decision variables in the MPC.

7.3. Distance Constraints

Applying Lemma 1 to assess collision avoidance between $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$ as demonstrated in the previous section, no measure of proximity between the objects is involved. The distance conditions (7.31) and (7.38) provide a more flexible way to include dual collision avoidance into the MPC problem. Since the distance provides a means to evaluate the proximity between $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$, the solution of the feasibility problem yields the distance and, thus, provides information that can be further processed if needed. However, if such a measure is important, collision avoidance can be formulated using the distance between two objects, defined as

$$\operatorname{dist}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) = \operatorname{dist}(\mathbf{0}, \mathcal{O} - \mathcal{V}(\boldsymbol{\eta})) = \inf_{\boldsymbol{p}} \{||\boldsymbol{p}||_2 : \boldsymbol{p} \in \mathcal{O} - \mathcal{V}(\boldsymbol{\eta})\}. \tag{7.18}$$

With this, (7.1) can be expressed using (7.18) as

$$\mathcal{V}(\boldsymbol{\eta}) \cap \mathcal{O} = \emptyset \Leftrightarrow \operatorname{dist}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) \geq d_{\text{safe}}, \tag{7.19}$$

where $d_{\text{safe}} \in \mathbb{R}^+$ is a safety distance. If $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$ collide, $\operatorname{dist}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) = 0$ and, thus, the distance approach can be used if no measure for penetration is needed. Note that (7.18) is a continuous, convex, see [70, Chap. 4], but not continuously differentiable function. Furthermore, observe that (7.18) constitutes an optimization problem, where the optimal solution yields the distance between the two objects. Thus, including (7.18) in an MPC setup in order to evaluate collision avoidance leads to a constraint that is itself an optimization problem. The lack of continuous differentiability of (7.18) can lead to difficulties in constraint gradient evaluation [45, 15]. Therefore, the Lagrange dual problem is used that allows a re-formulation as a feasibility problem, which can be included into the MPC setup directly. Therefore, the subsequent sections rely on the following definitions.

Definition 5 (Lagrange Dual Problem [20]) *Given a general NLP of the form* (6.25)*, the Lagrange dual problem is given by*

$$\sup_{\boldsymbol{\lambda}, \boldsymbol{\mu}} g(\boldsymbol{\lambda}, \boldsymbol{\mu}) \tag{7.20a}$$

$$\textit{s.t. } \boldsymbol{\mu} \geq \mathbf{0}, \tag{7.20b}$$

where $g(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is the dual function, which is defined as

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\boldsymbol{w}} \{c(\boldsymbol{w}) + \boldsymbol{\lambda}^\top \boldsymbol{g}(\boldsymbol{w}) + \boldsymbol{\mu}^\top \boldsymbol{h}(\boldsymbol{w})\}. \tag{7.21}$$

Definition 6 (Conjugate Function [20]) *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function, then the conjugate function $f^* : \mathbb{R}^n \to \mathbb{R}$ is defined as*

$$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x} \in \mathbb{R}^n} \{\boldsymbol{y}^\top \boldsymbol{x} - f(\boldsymbol{x})\}. \tag{7.22}$$

Note that f^* is convex even if f is not.

7.3.1. Singleton Case

If $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p}^{\mathcal{V}}\}$, then (7.18) reduces to

$$\operatorname{dist}(\boldsymbol{p}^{\mathcal{V}}, \mathcal{O}) = \inf_{\boldsymbol{p}}\{||\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{p}||_2 : A\boldsymbol{p} \leq \boldsymbol{b}\}. \tag{7.23}$$

Following, e.g., Boyd and Vandenberghe [20, Sec. 8.1.2], the Lagrange dual problem according to Definition 5 of (7.23) is derived. To this end, the primal problem (7.23) is formulated as the auxiliary optimization problem

$$\inf_{\boldsymbol{p},\boldsymbol{y}} \; ||\boldsymbol{y}||_2 \tag{7.24a}$$

$$\text{s.t. } A\boldsymbol{p} \leq \boldsymbol{b}, \tag{7.24b}$$

$$\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{p} = \boldsymbol{y}, \tag{7.24c}$$

which allows to express the Lagrangian function mathematically as

$$\mathcal{L}(\boldsymbol{p}, \boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = ||\boldsymbol{y}||_2 + \boldsymbol{\mu}^\top(A\boldsymbol{p} - \boldsymbol{b}) + \boldsymbol{\lambda}^\top(\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{p} - \boldsymbol{y}), \tag{7.25}$$

where $\boldsymbol{\mu} \geq \boldsymbol{0}$ and $\boldsymbol{\lambda} \in \mathbb{R}^2$ are the Lagrange multipliers. The dual function is then given by

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\boldsymbol{p},\boldsymbol{y}} \mathcal{L}(\boldsymbol{p}, \boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\boldsymbol{p},\boldsymbol{y}}\{||\boldsymbol{y}||_2 + \boldsymbol{\mu}^\top(A\boldsymbol{p} - \boldsymbol{b}) + \boldsymbol{\lambda}^\top(\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{p} - \boldsymbol{y})\} \tag{7.26}$$

which, after re-arranging and algebraic manipulation, yields

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\boldsymbol{y}}\{||\boldsymbol{y}||_2 - \boldsymbol{\lambda}^\top\boldsymbol{y}\} + \inf_{\boldsymbol{p}}\{(\boldsymbol{\mu}^\top A - \boldsymbol{\lambda}^\top)\boldsymbol{p}\} - \boldsymbol{\mu}^\top\boldsymbol{b} + \boldsymbol{\lambda}^\top\boldsymbol{p}^{\mathcal{V}}. \tag{7.27}$$

In view of Definition 6, the first term in (7.27) corresponds to the (negative value of the) conjugate function of the norm, i.e.,

$$\inf_{\boldsymbol{y}}\{||\boldsymbol{y}||_2 - \boldsymbol{\lambda}^\top\boldsymbol{y}\} = \begin{cases} 0 & \text{if } ||\boldsymbol{\lambda}||_2 \leq 1, \\ -\infty & \text{otherwise.} \end{cases} \tag{7.28}$$

The second term in (7.27) corresponds to a plane in space, which has no global minimum unless all coefficients are zero, i.e.,

$$\inf_{\boldsymbol{p}}\{(\boldsymbol{\mu}^\top A - \boldsymbol{\lambda}^\top)\boldsymbol{p}\} = \begin{cases} 0 & \text{if } \boldsymbol{\mu}^\top A = \boldsymbol{\lambda}^\top, \\ -\infty & \text{otherwise.} \end{cases} \tag{7.29}$$

With these implicitly defined constraints, the Lagrange dual problem of (7.23) can be expressed as

$$\sup_{\boldsymbol{\mu}} \; \boldsymbol{\mu}^\top(A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b}) \tag{7.30a}$$

$$\text{s.t. } ||A^\top\boldsymbol{\mu}||_2 \leq 1, \tag{7.30b}$$

$$\boldsymbol{\mu} \geq \boldsymbol{0}. \tag{7.30c}$$

This allows to express (7.1) as the following proposition.

Proposition 3 *Suppose $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p}^{\mathcal{V}}\}$ and $\mathcal{O} = \{\boldsymbol{p} \in \mathbb{R}^n : A\boldsymbol{p} \leq \boldsymbol{b}\}$, then* (7.1) *can be re-formulated as the feasibility problem*

$$dist(\boldsymbol{p}^{\mathcal{V}}, \mathcal{O}) \geq d_{safe} \Leftrightarrow \exists \boldsymbol{\mu} \geq \boldsymbol{0} : (A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu} \geq d_{safe} \wedge ||A^\top \boldsymbol{\mu}||_2 \leq 1 \tag{7.31}$$

using the Lagrange dual problem of (7.23) *with $d_{safe} \in \mathbb{R}^+$.*

Note that if $\mathrm{dist}(\boldsymbol{p}^{\mathcal{V}}, \mathcal{O}) \geq d_{\mathrm{safe}}$, then $\boldsymbol{z} = A^\top \boldsymbol{\mu}$ defines a separating hyperplane $\mathcal{H}(A^\top \boldsymbol{\mu}, \boldsymbol{b}^\top \boldsymbol{\mu})$ of $\boldsymbol{p}^{\mathcal{V}}$ and $\mathcal{O}$.

If $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p}^{\mathcal{V}}\}$, Prop. 3 can be used to formulate the feasible set of states, i.e.,

$$\mathcal{X} = \left\{(\boldsymbol{x}(t), \boldsymbol{\mu}(t)) \in \mathbb{R}^{n_x} \times \mathbb{R}^{\check{L}} : (A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu} \geq d_{\mathrm{safe}}, \ \boldsymbol{\mu} \geq \boldsymbol{0}, ||A^\top \boldsymbol{\mu}||_2 \leq 1\right\}, \tag{7.32}$$

where $d_{\mathrm{safe}} \in \mathbb{R}^+$ provides a safety distance between $\boldsymbol{p}^{\mathcal{V}}$ and $\mathcal{O}$. Similar to the indicator approach, the feasibility problem (7.31) is included by means of (7.32) and $\boldsymbol{\mu}$ is an additional decision variable in the MPC. Here, the additional norm inequality constraint $||A^\top \boldsymbol{\mu}|| \leq 1$ needs to be taken into account and, therefore, the distance approach constitutes a more complex MPC problem compared with the indicator approach.

7.3.2. Full Body Case

If $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p} \in \mathbb{R}^n : C(\boldsymbol{\eta})\boldsymbol{p} \leq \boldsymbol{d}(\boldsymbol{\eta})\}$, following [20, Sec. 8.2.2], analogously to the previous section, the Lagrange dual problem of (7.18) is derived. To this end, the primal problem (7.18) is formulated as the auxiliary optimization problem

$$\inf_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} ||\boldsymbol{z}||_2 \tag{7.33a}$$

$$\text{s.t. } A\boldsymbol{x} \leq \boldsymbol{b}, \tag{7.33b}$$

$$C\boldsymbol{y} \leq \boldsymbol{d}, \tag{7.33c}$$

$$\boldsymbol{x} - \boldsymbol{y} = \boldsymbol{z}, \tag{7.33d}$$

which allows to express the Lagrangian function mathematically as

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = ||\boldsymbol{z}||_2 + \boldsymbol{\mu}^\top(A\boldsymbol{x} - \boldsymbol{b}) + \boldsymbol{\lambda}^\top(C\boldsymbol{y} - \boldsymbol{d}) + \boldsymbol{\sigma}^\top(\boldsymbol{x} - \boldsymbol{y} - \boldsymbol{z}), \tag{7.34}$$

where $\boldsymbol{\mu} \geq \boldsymbol{0}$, $\boldsymbol{\lambda} \geq \boldsymbol{0}$, and $\boldsymbol{\sigma} \in \mathbb{R}^2$ are the Lagrange multipliers. The dual function is then defined as

$$\begin{aligned} g(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma}) &= \inf_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma}) \\ &= \inf_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \{||\boldsymbol{z}||_2 + \boldsymbol{\mu}^\top(A\boldsymbol{x} - \boldsymbol{b}) + \boldsymbol{\lambda}^\top(C\boldsymbol{y} - \boldsymbol{d}) + \boldsymbol{\sigma}^\top(\boldsymbol{x} - \boldsymbol{y} - \boldsymbol{z})\}, \end{aligned} \tag{7.35}$$

which, after algebraic manipulation, yields

$$\begin{aligned} g(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = &\inf_{\boldsymbol{z}}\{||z||_2 - \boldsymbol{\sigma}^\top \boldsymbol{z}\} - \boldsymbol{\mu}^\top \boldsymbol{b} - \boldsymbol{\lambda}^\top \boldsymbol{d} \\ &+ \inf_{\boldsymbol{x}}\{(\boldsymbol{\mu}^\top A + \boldsymbol{\sigma}^\top)\boldsymbol{x}\} + \inf_{\boldsymbol{y}}\{(\boldsymbol{\lambda}^\top C - \boldsymbol{\sigma}^\top)\boldsymbol{y}\}. \end{aligned} \tag{7.36}$$

Analogously to the previous section, these terms can be simplified, which, finally, results in the Lagrange dual problem

$$\sup_{\boldsymbol{\lambda},\boldsymbol{\mu}} \; -\boldsymbol{\mu}^\top \boldsymbol{b} - \boldsymbol{\lambda}^\top \boldsymbol{d} \tag{7.37a}$$

$$\text{s.t. } A^\top \boldsymbol{\mu} + C^\top \boldsymbol{\lambda} = \boldsymbol{0}, \tag{7.37b}$$

$$||A^\top \boldsymbol{\mu}||_2 \leq 1, \tag{7.37c}$$

$$\boldsymbol{\lambda} \geq \boldsymbol{0}, \; \boldsymbol{\mu} \geq \boldsymbol{0}. \tag{7.37d}$$

This allows to express (7.1) as the following proposition.

Proposition 4 *Suppose* $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p} \in \mathbb{R}^n : C(\boldsymbol{\eta})\boldsymbol{p} \leq \boldsymbol{d}(\boldsymbol{\eta})\}$ *and* $\mathcal{O} = \{\boldsymbol{p} \in \mathbb{R}^n : A\boldsymbol{p} \leq \boldsymbol{b}\}$, *then* (7.1) *can be re-formulated as the feasibility problem*

$$\begin{aligned} \mathit{dist}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) > d_{\mathit{safe}} \Leftrightarrow \exists \boldsymbol{\lambda} \geq \boldsymbol{0}, \; \boldsymbol{\mu} \geq \boldsymbol{0} : & \\ - \boldsymbol{d}^\top(\boldsymbol{\eta})\boldsymbol{\lambda} - \boldsymbol{b}^\top \boldsymbol{\mu} \geq d_{\mathit{safe}} & \\ \wedge A^\top \boldsymbol{\mu} + C^\top(\boldsymbol{\eta})\boldsymbol{\lambda} = \boldsymbol{0} & \\ \wedge ||A^\top \boldsymbol{\mu}||_2 \leq 1. & \end{aligned} \tag{7.38}$$

using the Lagrange dual problem of (7.18) *with* $d_{\mathit{safe}} \in \mathbb{R}^+$.

Note that if $\text{dist}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) > d_{\text{safe}}$, then $\boldsymbol{z} = A^\top \boldsymbol{\mu} = -C^\top(\boldsymbol{\eta})\boldsymbol{\lambda}$ defines a separating hyperplane $\mathcal{H}(A^\top \boldsymbol{\mu}, \boldsymbol{b}^\top \boldsymbol{\mu})$ of $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$.

If $\mathcal{V}(\boldsymbol{\eta})$ is given as a convex polytope, Prop. 4 can be used to formulate the feasible set of states

$$\begin{aligned} \mathcal{X} = \big\{(\boldsymbol{x}(t), \boldsymbol{\mu}(t), \boldsymbol{\lambda}(t)) \in \mathbb{R}^{n_x} \times \mathbb{R}^{\check{L}} \times \mathbb{R}^L : & - \boldsymbol{b}^\top \boldsymbol{\mu} - \boldsymbol{d}^\top(\boldsymbol{\eta})\boldsymbol{\lambda} \geq d_{\text{safe}}, \\ & \boldsymbol{\mu} \geq \boldsymbol{0}, \; \boldsymbol{\lambda} \geq \boldsymbol{0}, \\ & A^\top \boldsymbol{\mu} + C^\top(\boldsymbol{\eta})\boldsymbol{\lambda} = \boldsymbol{0}, \\ & ||A^\top \boldsymbol{\mu}||_2 \leq 1\big\}, \end{aligned} \tag{7.39}$$

where $d_{\text{safe}} \in \mathbb{R}^+$ provides a safety distance. Since the feasibility problem (7.38) is included by means of (7.39), $\boldsymbol{\mu}, \boldsymbol{\lambda}$ are additional decision variables in the MPC. Compared to the singleton distance constraints (7.32), the full body distance constraints are more complex due to the additional consistency equality constraint.

7.4. Signed Distance Constraints

The signed distance conditions (7.50) and (7.55) provide the most flexible way to include dual collision avoidance into the MPC problem since the signed distance provides a means to evaluate the proximity between $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$ as well as the penetration in case of a collision. Thus, the severity of a collision can be processed as part of the MPC using soft-constraints, see, e.g., [175, 161, 137]. A fundamental concept for evaluating if two objects, e.g., $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$, intersect is the signed distance function that extends the concept of the distance between convex sets using a measure of penetration in case of collision. It is defined as

$$\operatorname{sd}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) = \begin{cases} \operatorname{dist}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}), & \text{if } \mathcal{V}(\boldsymbol{\eta}) \cap \mathcal{O} = \emptyset \\ -\operatorname{pen}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}), & \text{if } \mathcal{V}(\boldsymbol{\eta}) \cap \mathcal{O} \neq \emptyset \end{cases}, \tag{7.40}$$

where $\operatorname{dist}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O})$ is the distance between $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$ according to (7.18). Furthermore, the signed distance makes use of the penetration depth, which can be defined as[3]

$$\operatorname{pen}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) = \operatorname{dist}(\mathbf{0}, \mathbb{R}^n \setminus (\mathcal{O} - \mathcal{V}(\boldsymbol{\eta}))). \tag{7.41}$$

Thus, on the one hand, $\mathcal{V}(\boldsymbol{\eta}) \cap \mathcal{O} = \emptyset$ gives

$$\operatorname{dist}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) > 0, \quad \operatorname{pen}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) = 0. \tag{7.42}$$

On the other hand, $\mathcal{V}(\boldsymbol{\eta}) \cap \mathcal{O} \neq \emptyset$ gives

$$\operatorname{dist}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) = 0, \quad \operatorname{pen}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) > 0. \tag{7.43}$$

Consequently, (7.40) can be compactly written as

$$\operatorname{sd}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) = \operatorname{dist}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) - \operatorname{pen}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}). \tag{7.44}$$

Based on this, (7.1) can be expressed using the signed distance function, i.e.,

$$\mathcal{V}(\boldsymbol{\eta}) \cap \mathcal{O} = \emptyset \Leftrightarrow \operatorname{sd}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) \geq d_{\text{safe}} \tag{7.45}$$

where $d_{\text{safe}} \in \mathbb{R}$. In contrast to the distance formulation (7.18), the signed distance allows for $d_{\text{safe}} < 0$ such that a certain amount of overlap between $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$ can still be considered as not colliding. Similar to the distance function, the signed distance function shares the same caveats of being not continuously differentiable and imposing an optimization problem as a constraint within an MPC setup.

[3] Note that for the distance the infimum and the minimum coincide, but for the penetration depth it is actually necessary to use the infimum in the definition since $\mathbb{R}^2 \setminus (\mathcal{O} - \mathcal{V}(\boldsymbol{\eta}))$ is an open set.

7.4.1. Singleton Case

If $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p}^{\mathcal{V}}\}$, then (7.6) can be adapted to give the signed distance between $\boldsymbol{p}^{\mathcal{V}}$ and a supporting hyperplane $\mathcal{H}(\hat{z}, \hat{z}^\top \boldsymbol{s}_{O}(\hat{z}))$, i.e.,

$$\mathrm{sd}(\boldsymbol{p}^{\mathcal{V}}, \mathcal{H}(\hat{z}, \hat{z}^\top \boldsymbol{s}_{O}(\hat{z}))) = \hat{z}^\top (\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{s}_{O}(\hat{z})), \tag{7.46}$$

where $\hat{z} = {}^{z}/_{||z||_2}$ is the normalized vector with direction $\boldsymbol{z}$. Due to the fact that $O = \bigcap_j \mathcal{H}^-(\hat{z}_j, \hat{z}_j^\top \boldsymbol{s}_{O}(\hat{z}_j))$, (7.40) can be expressed as

$$\begin{aligned} \mathrm{sd}(\boldsymbol{p}^{\mathcal{V}}, O) &= \sup_{\hat{z}} \left\{\mathrm{sd}(\boldsymbol{p}^{\mathcal{V}}, \mathcal{H}(\hat{z}, \hat{z}^\top \boldsymbol{s}_{O}(\hat{z})))\right\} && \text{(7.47a)} \\ &= \sup_{\hat{z}} \left\{\hat{z}^\top (\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{s}_{O}(\hat{z}))\right\}. && \text{(7.47b)} \end{aligned}$$

Inserting the definition of the support mapping (7.7) in (7.47b) yields

$$\mathrm{sd}(\boldsymbol{p}^{\mathcal{V}}, O) = \sup_{\hat{z}} \left\{\hat{z}^\top \boldsymbol{p}^{\mathcal{V}} + \inf_{\boldsymbol{p} \in O} \{-\hat{z}^\top \boldsymbol{p}\}\right\}. \tag{7.48}$$

The associated Lagrange dual problem of the inner minimization in (7.48) finally leads to

$$\mathrm{sd}(\boldsymbol{p}^{\mathcal{V}}, O) = \sup_{\boldsymbol{\mu} \geq \boldsymbol{0}} \left\{\frac{(A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu}}{||A^\top \boldsymbol{\mu}||_2}\right\}. \tag{7.49}$$

Note that in, e.g., Boyd and Vandenberghe [20, Ex. 8.5], [161] an alternative formulation is given where $||A^\top \boldsymbol{\mu}||_2 = 1$ is set explicitly in (7.49). In summary, the collision avoidance conditions for the singleton case using the dual signed distance approach can be summarized as follows.

Proposition 5 *Suppose $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p}^{\mathcal{V}}\}$ and $O = \{\boldsymbol{p} \in \mathbb{R}^n : A\boldsymbol{p} \leq \boldsymbol{b}\}$, then* (7.1) *can be re-formulated as the feasibility problem*

$$sd(\boldsymbol{p}^{\mathcal{V}}, O) \geq d_{safe} \Leftrightarrow \exists \boldsymbol{\mu} \geq \boldsymbol{0} : \frac{(A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu}}{||A^\top \boldsymbol{\mu}||_2} \geq d_{safe}, \tag{7.50}$$

using the Lagrange dual problem (7.49).

Note that if $\boldsymbol{p}^{\mathcal{V}} \notin O$, then $\boldsymbol{z} = A^\top \boldsymbol{\mu}$ defines a separating hyperplane $\mathcal{H}(A^\top \boldsymbol{\mu}, \boldsymbol{b}^\top \boldsymbol{\mu})$ of $\boldsymbol{p}^{\mathcal{V}}$ and O. If $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p}^{\mathcal{V}}\}$, Prop. 5 can be used to formulate the feasible set of states

$$\mathcal{X} = \left\{(\boldsymbol{x}(t), \boldsymbol{\mu}(t)) \in \mathbb{R}^{n_x} \times \mathbb{R}^{\check{L}} : \frac{(A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu}}{||A^\top \boldsymbol{\mu}||_2} \geq d_{\mathrm{safe}},\ \boldsymbol{\mu} \geq \boldsymbol{0}\right\} \tag{7.51}$$

where $d_{\mathrm{safe}} \in \mathbb{R}$ provides a safety distance if $d_{\mathrm{safe}} > 0$ and the maximum acceptable penetration depth if $d_{\mathrm{safe}} \leq 0$. Similar to the indicator and distance approach, the feasibility problem (7.50) is included by means of (7.51) and $\boldsymbol{\mu}$ is an additional decision variable in the MPC. In contrast to the distance approach, the inequality norm constraint can be included to the collision avoidance inequality constraint. Alternatively, as proposed in, e.g., [161] the norm can be explicitly set, i.e., $||A^\top \boldsymbol{\mu}||_2 = 1$. This constraint defines a non-convex feasible set for $\boldsymbol{\mu}$ and is therefore more complex than the distance constraint formulation. The proposed approach avoids including this non-convex equality constraint at the cost of including the norm to the collision avoidance inequality constraint by means of a fraction.

7.4.2. Full Body Case

If $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p} \in \mathbb{R}^2 : C(\boldsymbol{\eta})\boldsymbol{p} \leq \boldsymbol{d}(\boldsymbol{\eta})\}$, then (7.40) can be expressed using

$$\begin{aligned} \mathrm{sd}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) &= \mathrm{sd}(\mathbf{0}, \mathcal{O} - \mathcal{V}(\boldsymbol{\eta})) && (7.52a)\\ &= \sup_{\hat{\boldsymbol{z}}}\{\mathrm{sd}(\mathbf{0}, \mathcal{H}(\hat{\boldsymbol{z}}, \hat{\boldsymbol{z}}^\top \boldsymbol{s}_{\mathcal{O}-\mathcal{V}}(\hat{\boldsymbol{z}})))\} && (7.52b)\\ &= \sup_{\hat{\boldsymbol{z}}}\{\mathbf{0} - \hat{\boldsymbol{z}}^\top \boldsymbol{s}_{\mathcal{O}-\mathcal{V}}(\hat{\boldsymbol{z}})\}. && (7.52c) \end{aligned}$$

In other words, (7.52a) calculates the signed distance between the origin and the support point of the Minkwowski sum of $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$. Inserting the definition of the support mapping (7.7) in (7.52c) yields

$$\mathrm{sd}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) = \sup_{\hat{\boldsymbol{z}}}\left\{\inf_{\boldsymbol{p}_1 \in \mathcal{O}, \boldsymbol{p}_2 \in \mathcal{V}(\boldsymbol{\eta})}\left\{\hat{\boldsymbol{z}}^\top(\boldsymbol{p}_2 - \boldsymbol{p}_1)\right\}\right\}. \tag{7.53}$$

With this, the Lagrange dual problem of the inner minimization in (7.53) can be derived, which leads to

$$\mathrm{sd}(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) = \sup_{\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\mu} \geq \mathbf{0}}\left\{\frac{-\boldsymbol{d}^\top(\boldsymbol{\eta})\boldsymbol{\lambda} - \boldsymbol{b}^\top\boldsymbol{\mu}}{||A^\top\boldsymbol{\mu}||_2} : A^\top\boldsymbol{\mu} + C^\top(\boldsymbol{\eta})\boldsymbol{\lambda} = \mathbf{0}\right\}. \tag{7.54}$$

Again, note that in, e.g., [161] an alternative formulation is given where $||A^\top\boldsymbol{\mu}||_2 = 1$ is set explicitly in (7.54). The dual signed distance collision avoidance conditions for the full body case can be summarized as follows.

Proposition 6 *Suppose* $\mathcal{V}(\boldsymbol{\eta}) = \{\boldsymbol{p} \in \mathbb{R}^n : C(\boldsymbol{\eta})\boldsymbol{p} \leq \boldsymbol{d}(\boldsymbol{\eta})\}$ *and* $\mathcal{O} = \{\boldsymbol{p} \in \mathbb{R}^n : A\boldsymbol{p} \leq \boldsymbol{b}\}$, *then* (7.1) *can be re-formulated as the feasibility problem*

$$\begin{aligned} &sd(\mathcal{V}(\boldsymbol{\eta}), \mathcal{O}) \geq d_{safe}\\ &\Leftrightarrow \exists \boldsymbol{\lambda} \geq \mathbf{0},\ \boldsymbol{\mu} \geq \mathbf{0} : \frac{-\boldsymbol{d}^\top(\boldsymbol{\eta})\boldsymbol{\lambda} - \boldsymbol{b}^\top\boldsymbol{\mu}}{||A^\top\boldsymbol{\mu}||_2} \geq d_{safe} \wedge A^\top\boldsymbol{\mu} + C^\top(\boldsymbol{\eta})\boldsymbol{\lambda} = \mathbf{0}. \end{aligned} \tag{7.55}$$

using the Lagrange dual problem (7.54).

Note that if $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$ do not overlap, $\boldsymbol{z} = A^\top \boldsymbol{\mu} = -C^\top \boldsymbol{\lambda}$ defines a separating hyperplane $\mathcal{H}(A^\top \boldsymbol{\mu}, \boldsymbol{b}^\top \boldsymbol{\mu})$ of $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$.

If $\mathcal{V}(\boldsymbol{\eta})$ is given as a convex polytope, Prop. 6 can be used to formulate the feasible set of states

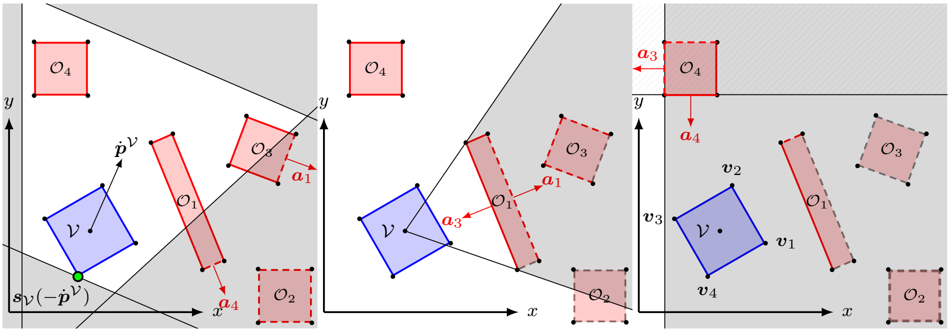
$$\begin{aligned} \mathcal{X} = \Bigg\{ &(\boldsymbol{x}(t), \boldsymbol{\mu}(t), \boldsymbol{\lambda}(t)) \in \mathbb{R}^{n_x} \times \mathbb{R}^{\check{L}} \times \mathbb{R}^{L} : \\ &\frac{-\boldsymbol{b}^\top \boldsymbol{\mu} - \boldsymbol{d}^\top(\boldsymbol{\eta})\boldsymbol{\lambda}}{||A^\top \boldsymbol{\mu}||_2} \geq d_{\text{safe}},\ \boldsymbol{\mu} \geq \boldsymbol{0},\ \boldsymbol{\lambda} \geq \boldsymbol{0}, A^\top \boldsymbol{\mu} + C^\top(\boldsymbol{\eta})\boldsymbol{\lambda} = \boldsymbol{0} \Bigg\}, \end{aligned} \tag{7.56}$$

where $d_{\text{safe}} \in \mathbb{R}$ provides a safety distance if $d_{\text{safe}} > 0$ and the maximum acceptable penetration depth if $d_{\text{safe}} \leq 0$. Since the feasibility problem (7.55) is included by means of (7.56), $\boldsymbol{\mu}, \boldsymbol{\lambda}$ are additional decision variables in the MPC. Compared to the signed distance singleton constraints, the full body constraints are more complex due to the additional consistency constraint.

7.5. Culling Procedure

Due to the disproportionate effect of an increasing number of obstacles on the number of dual decision variables see, e.g., [101, 45, 175], a culling procedure can be applied to the NLP to reduce its complexity, see, e.g., [74]. This concept is a crucial aspect in computer graphics and is inherently connected to the viewpoint and perspective of the object of interest. The culling concept can be divided into (view) frustum , occlusion, and backface culling. The first is based on what is in the field of view of the object of interest, the so-called view frustum, which, in computer graphics, is typically defined using a pyramid from which the frustum is cut out using a near and a far clipping plane. The second deals with inter-object relations, i.e., is an object occluded by another object (that is typically closer to the viewpoint compared to the occluded object). The third operates on an intra-object level, i.e., is an object blocking the sight of other parts of itself in relation to a viewpoint. Any thus identifiable, un-visible parts of obstacles or the object of interest itself need not be considered for rendering in a computer graphics context which translates to ignoring the identified objects/faces in a collision avoidance MPC problem since those parts cannot contribute to the rendering of the scene, and the collision evaluation, respectively. In Fig. 7.3 an example illustration of the three culling principles is given. The culling procedure identifies faces of the detected obstacles that can be neglected in the optimization such that only a minimal set of dual decision variables (and normal vectors) is passed to the optimization, thereby largely reducing NLP problem size. Note that the frustum, occlusion, and backface conditions are evaluated in logical (inclusive) disjunction, i.e., they can be evaluated in any order and if any of the individual

[3] Note that for the signed distance, this depends on d_{safe}. If $d_{\text{safe}} < 0$, overlapping of $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$ is considered as not colliding.



(a) Illustration of a view frustum culling procedure. The near clipping plane of the view frustum is defined using a supporting hyperplane associated with the apparent velocity $\dot{\boldsymbol{p}}^{\mathcal{V}}$ of the controlled object (blue). Faces lying entirely outside the view frustum, i.e., in one of the four half-spaces (shaded gray areas) associated with it, are ignored. In this example $\mathcal{O}_2$ is completely culled and, for $\mathcal{O}_1$, the face associated with the normal vector $\boldsymbol{a}_4$ can be ignored as well as the face associated with the normal vector $\boldsymbol{a}_1$ of $\mathcal{O}_3$.

(b) Illustration of an occlusion (shadow) culling procedure with obstacle $\mathcal{O}_1$ as the occluder with the face associated with $\boldsymbol{a}_3$ defining the near clipping plane. Based on the current position $\boldsymbol{p}^{\mathcal{V}}$ all obstacle faces, which lie in the shadow (shaded gray area) of the occluder can be culled. In this example (the rest of) all faces of obstacle $\mathcal{O}_3$ can be culled and the face of the occluder $\mathcal{O}_1$ associated with $\boldsymbol{a}_1$ can also be ignored in this step.

(c) Illustration of the backface culling procedure for obstacle $\mathcal{O}_4$ (red) and the controlled object (blue). All vertices of the latter lie in the half-space $\mathcal{H}^-(\boldsymbol{a}_3, b_3)$ (shaded gray area) such that this particular face is not visible to the controlled object and can thus be culled. This does not hold for, e.g., the half-space associated with $\boldsymbol{a}_4$ (diagonal gray lines) of $\mathcal{O}_4$, i.e. that face must not be culled. This procedure is evaluated for each obstacle.

Figure 7.3.: Illustration of the three culling techniques. First, a frustum culling procedure is used (Fig. 7.3a). Subsequently, occlusion (shadow) culling is evaluated (Fig. 7.3b) and lastly, backface culling is applied to the scenario (Fig. 7.3c). Faces identified by any previous step are denoted using dashed gray lines while faces identified by the current step are indicated using dashed red lines.

conditions identifies a particular face as unnecessary for evaluating collision avoidance, it can be neglected.

For ease of notation, in the following we assume $K = 1$. Subsequently, the associated (discretized) dual decision variables arising from the collision avoidance constraints in (6.73) are denoted using $\boldsymbol{\mu}_k$, $\boldsymbol{\lambda}_k$. Furthermore, $\mu_{j,k}$, $\lambda_{j,k}$ correspond to the j-th face of $\mathcal{O}_k$ and $\mathcal{V}_k$ and, therefore, to the normal vectors $\boldsymbol{a}_{j,k}$ in $A_k^\top$ and $\boldsymbol{c}_{j,k}$ in $C_k^\top$, at predicted time instants t_k, respectively. If dual variables are found that satisfy any of the three different constraint cases for a given $\boldsymbol{x}_k$, then $\boldsymbol{z}_k = A_k^\top \boldsymbol{\mu}_k = -C_k^\top \boldsymbol{\lambda}_k$ defines a separating hyperplane $\mathcal{H}(\boldsymbol{z}_k, \boldsymbol{b}_k^\top \boldsymbol{\mu}_k)$ of $\mathcal{O}_k$ and $\mathcal{V}_k$. In other words, e.g., $\boldsymbol{\mu}_k$ defines the linear combination of normal vectors in $A_k^\top$ such that $\boldsymbol{z}_k = A_k^\top \boldsymbol{\mu}_k$ and $\mathcal{O} \in \mathcal{H}^-(\boldsymbol{z}_k, \boldsymbol{b}_k^\top \boldsymbol{\mu}_k)$, $\mathcal{V} \notin \mathcal{H}^-(\boldsymbol{z}_k, \boldsymbol{b}_k^\top \boldsymbol{\mu}_k)$.

The culling procedure needs to be evaluated using predicted positions of both the controlled object and any obstacle. Since this information is not necessarily known *a priori* to solving the MPC NLP problem, it is assumed that a (conservative) guess is available for these quantities such that no faces are culled that might become relevant for the collision avoidance constraint calculations. To this end, the possible states need to be evaluated which, e.g., can be achieved based on the inputs of the system. For instance, if one of the inputs in $\boldsymbol{u}$ is primarily associated with forward motion, this input will be kept at its maximum value. Furthermore, if one of the inputs in $\boldsymbol{u}$ is associated with a rotation of the controlled object, the inputs

$$u_k^{(1)} = u(t_{i-1}) \tag{7.57a}$$

$$u_k^{(2)} = \max\{u^-, u(t_{i-1}) - k\Delta t\dot{u}^{\max}\} \tag{7.57b}$$

$$u_k^{(3)} = \min\{u^+, u(t_{i-1}) + k\Delta t\dot{u}^{\max}\} \tag{7.57c}$$

for $k = 0, \dots, N$ can be used to obtain the possible system states as described in [172]. Therein, u^-, u^+ denote the lower and upper input bound and $\dot{u}^{\max}$ the maximum rate of change of the input, respectively. Furthermore, (7.57a) applies the previous input for the entire prediction horizon, (7.57b) applies the lower input bound as quickly as possible (taking into account rate constraints and the previously-applied input) and (7.57c) achieves the same as (7.57b) except for the upper input bound. Alternatively, the previous (optimal) prediction could be used instead of using the last applied input (7.57a).

7.5.1. View Frustum Culling

The view frustum is typically defined using four half-spaces (in 2D), namely, the near, far, left, and right clipping planes, see Fig. 7.3a. As can be seen, the near (and far) clipping plane is defined using a supporting hyperplane with the direction of travel $\dot{\boldsymbol{p}}^{\mathcal{V}}$ as its normal (and the far clipping plane being parallel to the near clipping plane). Therefore, suppose $||\dot{\boldsymbol{p}}_k^{\mathcal{V}}||_2 > 0$ and, for better readability, let $\dot{\boldsymbol{p}}_k := \dot{\boldsymbol{p}}_k^{\mathcal{V}}$ for time step k. With this, the view frustum condition can be expressed mathematically using

$$\text{Neglect } \mu_{j,k} \text{ in (6.73) if } \boldsymbol{v}_{j,k}^{\mathcal{O}} \notin \mathcal{F}_k \wedge \boldsymbol{v}_{j+1,k}^{\mathcal{O}} \notin \mathcal{F}_k, \tag{7.58}$$

where $\mathcal{F}_k = \mathcal{H}^+_{\text{near},k} \cap \mathcal{H}^+_{\text{far},k} \cap \mathcal{H}^+_{\text{left},k} \cap \mathcal{H}^+_{\text{right},k}$ defines the view frustum and, e.g., the near clipping plane defined as $\mathcal{H}_{\text{near},k} = \mathcal{H}(\dot{\boldsymbol{p}}_k, \dot{\boldsymbol{p}}_k^\top \boldsymbol{s}_{\mathcal{V}_k}(-\dot{\boldsymbol{p}}_k))$. Furthermore, $\boldsymbol{v}_{j,k}^{\mathcal{V}}$, $j = 1, \dots, L$ denote the (predicted) vertices of the controlled object, where each face defined, e.g., by the normal vectors $\boldsymbol{a}_{j,k}$ is associated with two vertices $\boldsymbol{v}_{l,k}$, $l = j, j+1$. The left- and right clipping plane is defined such that a user-defined view angle is obtained based on the direction of travel. In other words, it is to be checked if the vertices $\boldsymbol{v}_{j,k}^{\mathcal{O}}$, $\boldsymbol{v}_{j+1,k}^{\mathcal{O}}$ associated with the j-th face with normal vector $\boldsymbol{a}_{j,k}$ lie outside the view frustum defined using the near, far, left, and right clipping plane. Since that particular face is associated with the decision variable $\mu_{j,k}$ it can be neglected. Note that if (7.58) applies to all vertices at a given time step, the corresponding obstacle is completely omitted from the NLP for that time step.

7.5.2. Occlusion Culling

Shadow occlusion culling as depicted in Fig. 7.3b is applied to the scenario in order to determine occluded obstacles and/or obstacle faces in the shadow frustum. The shadow frustum is defined using three hyperplanes, namely, the near, left, and right clipping plane of the shadow frustum. With this, the occlusion condition can be expressed mathematically using

$$\text{Neglect } \mu_{j,k} \text{ in (6.73) if } \boldsymbol{v}^{\mathcal{O}}_{j,k} \in \mathcal{SF}_k \wedge \boldsymbol{v}^{\mathcal{O}}_{j+1,k} \in \mathcal{SF}_k, \tag{7.59}$$

where $\mathcal{SF}_k = \mathcal{H}^-_{\text{near},k} \cap \mathcal{H}^-_{\text{left},k} \cap \mathcal{H}^-_{\text{right},k}$ defines the shadow frustum. The difficulty for (shadow) occlusion culling lies in determining a good occluder, i.e., finding an obstacle which is likely to cast a large shadow and, therefore, has the potential to occlude as many other obstacle faces as possible. Among other principles which can be found, e.g., in [31, 73] possible occluders can be identified using the solid angle (measured in steradian) of all obstacles and choosing those obstacles as occluders which possess a solid angle greater than a user-defined threshold Ω_{min}. The solid angle can be approximated using

$$\Omega = a_k \frac{\boldsymbol{n}_k^\top (\boldsymbol{p}^{\mathcal{O}}_k - \boldsymbol{p}^{\mathcal{V}}_k)}{||\boldsymbol{p}^{\mathcal{O}}_k - \boldsymbol{p}^{\mathcal{V}}_k||_2^2}, \tag{7.60}$$

where a_k is the area of the occluder and $\boldsymbol{n}_k$ is the normal vector of the obstacle [73]. Here, $\boldsymbol{n}_k$ is the normal vector also defining the near clipping plane of the shadow frustum. This plane is defined using two vertices of an obstacle which are furthest apart from each other and whose corresponding normals (i.e. $\boldsymbol{a}_{i,k}$ in A_k) are facing towards the position $\boldsymbol{p}^{\mathcal{V}}_k$ of the controlled object which serves as the apex of the shadow frustum. Consequently, a_k is the length of the difference vector of these particular vertices. The left and right clipping plane are defined such that they pass through the near clipping plane's vertices and the apex, respectively. As can be seen, the solid angle takes into account multiple factors. First, a large obstacle will have a large area a_k. Second, an obstacle closer to the apex is more likely to be a good occluder and, third, how much of the area is head-on w.r.t. the apex.

7.5.3. Backface Culling

Backface culling is evaluated to identify faces that are hidden by other faces of the same object as seen from the perspective of the controlled object's vertices. Those hidden faces need not be considered in the NLP since they do not affect the imminent situation. This is formulated mathematically as

$$\text{Neglect } \mu_{j,k} \text{ in (6.73) if } \mathcal{V}_k \in \mathcal{H}^-(\boldsymbol{a}_j, b_j) \tag{7.61a}$$

or, in other words, check if all vertices of the controlled object $\boldsymbol{v}^{\mathcal{V}}_{l,k}$, $l = 1, \ldots, L$ lie in the supporting half-space $\mathcal{H}^-(\boldsymbol{a}_{j,k}, b_{j,k})$ defined by the j-th face with normal vector $\boldsymbol{a}_{j,k}$. In that case, this face is hidden by other faces of the obstacle and the associated decision variable $\mu_{j,k}$ can be neglected. Analogously, the same idea can be applied to faces of the controlled object from the perspective of the obstacle in order to reduce the number of dual object decision variables $\boldsymbol{w}_\lambda$ in (6.73), i.e.,

$$\text{Neglect } \lambda_{j,k} \text{ in (6.73) if } \mathcal{O}_k \in \mathcal{H}^-(\boldsymbol{c}_{j,k}, d_{j,k}). \tag{7.61b}$$

In Fig. 7.3c, this procedure is demonstrated for the obstacle faces.

8. Kalman Filter-Based Estimation

Typically, a control loop includes a state observer as depicted in Fig. 2.1. The task of the observer is to reconstruct state information needed by the control algorithms based on (usually noisy) measurements. The de-facto standard algorithm used in this context is the KF that exists in a variety of different formulations. The classical KF equations were derived for linear time invariant systems and have since then been adapted to nonlinear models, where, among other approaches, the nonlinear model is either successively linearized, or the so-called unscented transformation is used. The former leads to the EKF formulation while the latter is known as the Unscented Kalman Filter. In this chapter, we focus on the EKF equations that are presented in the predictor-corrector form. Furthermore, various extensions are introduced, such as adaptive covariance estimation, iterative measurement updates, and multi rate Kalman Filtering. For a thorough discussion of the Kalman filter and its variants see, e.g., [12, 56, 61, 141, 155].

8.1. Extended Kalman Filter

The KF is a statistically optimal filter in the sense that it minimizes the error covariance $P = E[(\boldsymbol{x}-\hat{\boldsymbol{x}})(\boldsymbol{x}-\hat{\boldsymbol{x}})^\top]$ of the estimate $\hat{\boldsymbol{x}}$. To this end, the filter assumes that the dynamics of the system can be described mathematically with

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) + G\boldsymbol{w}, && t > t_0, \quad \boldsymbol{x}(t_0) = \hat{\boldsymbol{x}}_0 && (8.1a)\\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v}, && t \geq t_0, && (8.1b) \end{aligned}$$

i.e., it is the nominal model (6.59b) augmented with additive process noise $\boldsymbol{w} \in \mathbb{R}^{n_w}$ and where $G \in \mathbb{R}^{n_x \times n_w}$ is the process noise gain matrix, and $\boldsymbol{v} \in \mathbb{R}^{n_y}$ is additive measurement noise. Since the EKF is, in essence, a discrete filter, it makes use of a discretization of the dynamical model, i.e.,

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{F}(\boldsymbol{x}_k,\boldsymbol{u}_k) + G\boldsymbol{w}_k, \quad \boldsymbol{x}_0 = \hat{\boldsymbol{x}}_0, && \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0},Q), && (8.2a)\\ \boldsymbol{y}_k &= \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k, && \boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0},R), && (8.2b) \end{aligned}$$

for $k \in \mathbb{N}_0$, where $\boldsymbol{F} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \mapsto \mathbb{R}^{n_x}$ is the discrete propagation function depending on the particular discretization at hand, $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0},Q)$ is the discrete additive process noise, Q is the process noise covariance matrix, and $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0},R)$ is the discrete additive measurement noise with the measurement noise covariance matrix R. With this, it is not necessary that $\Delta t_y = \Delta t_{\mathrm{ekf}}$, i.e., the measurement update rate can be different from the propagation rate. For example the forward Euler discrete propagation function reads $\boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{x}_k + \Delta t_{\mathrm{ekf}} \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$, where Δt_{ekf} is the discretization step size.

Based on this, the EKF equations read

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{F}(\hat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}_{k-1}), \tag{8.3a}$$

$$P_k^- = A_{k-1} P_{k-1}^+ A_{k-1}^\top + G Q_{k-1} G^\top, \tag{8.3b}$$

which is known as the prediction, and

$$K_k = P_k^- C_k^\top (C_k P_k^- C_k^\top + R_k)^{-1}, \tag{8.3c}$$

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + K_k(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-)), \tag{8.3d}$$

$$P_k^+ = (I - K_k C_k) P_k^- (I - K_k C_k)^\top + K_k R_k K_k^\top, \tag{8.3e}$$

which is known as the correction, and where $k \in \mathbb{N}_0$. Furthermore, $\hat{\boldsymbol{x}}_k^-$ is the *a priori* state estimate, i.e., before any measurement is taken into account, $\hat{\boldsymbol{x}}_k^+$ is the *a posteriori* state estimate, i.e., the estimate including the current measurement. It is also important to note that the correction step is only performed if a measurement becomes available, i.e., $\boldsymbol{y}_k$ is not necessarily defined for all $k \in \mathbb{N}$ but is assumed to be an integer multiple of the base sample time Δt_{ekf}, i.e., $\Delta t_y = m \Delta t_{\mathrm{ekf}},\ m \in \mathbb{N}$. Furthermore,

$$P_k^- = E[(\boldsymbol{x} - \hat{\boldsymbol{x}}_k^-)(\boldsymbol{x} - \hat{\boldsymbol{x}}_k^-)^\top] \tag{8.4}$$

is the state error covariance of the *a priori* estimate, and

$$P_k^+ = E[(\boldsymbol{x} - \hat{\boldsymbol{x}}_k^+)(\boldsymbol{x} - \hat{\boldsymbol{x}}_k^+)^\top] \tag{8.5}$$

is the *a posteriori* state error covariance. The filter is initialized with $\hat{\boldsymbol{x}}_0^+ = E[\hat{\boldsymbol{x}}_0]$ and $P_0^+ = E[(\boldsymbol{x} - \hat{\boldsymbol{x}}_0)(\boldsymbol{x} - \hat{\boldsymbol{x}}_0)^\top]$. The quantity

$$\boldsymbol{\varepsilon}_k^- = \boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-) \tag{8.6}$$

is called the innovation as illustrated in Fig. 8.1. Finally,

$$A_{k-1} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}(\hat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}_{k-1}), \quad C_k = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}(\hat{\boldsymbol{x}}_k^-), \tag{8.7}$$

are the Jacobian of the discrete propagation and measurement function, respectively. In other words, the prediction and correction of the error covariance is based on the linearization of the discrete propagation function and the measurement model around the current best estimate and the previous input (for A_{k-1}).

8.2. Iterated Extended Kalman Filter

The EKF propagates the error covariances based on a linearization of the discrete propagation function and the measurement model around the current best estimate and the

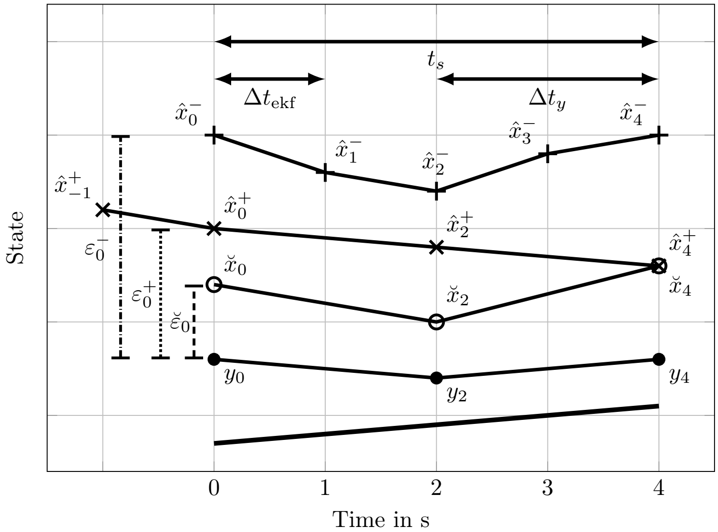


Figure 8.1.: Qualitative illustration of EKF and smoother quantities with *a priori* state $\hat{x}_k^-$ (plus), *a posteriori* state $\hat{x}_k^+$ (cross), smoothed state $\breve{x}_k$ (circle), and measurement y_k (filled dots), respective innovation ε_k^-, ε_k^+, $\breve{\varepsilon}_k$ (for $k = 0$), and real state (unmarked line), EKF sample time Δt_{ekf}, measurement sample time $\Delta t_y = 2\Delta t_{\mathrm{ekf}}$, and smoothing window $t_s = 2\Delta t_y$.

previous input. However, for highly nonlinear systems the linearization error can become relatively large. This motivates the use of a higher order extension of the EKF [see 141, Sec. 13.3.1]. After performing the measurement update (8.3c)-(8.3e), we get an improved estimate of the state, i.e., $\hat{\boldsymbol{x}}_k^+$. This estimate can be further improved by re-evaluating the measurement update with this new estimate. This procedure can be repeated as often as desired, typically, until the improvement of the new state estimate is negligible. It turns out that this procedure is equivalent to applying the Gauss-Newton method to the maximization problem of finding the maximum likelihood *a posteriori* state estimate for the given measurement making the iterated Extended Kalman Filter (IEKF) a simple but effective higher order extension to the EKF [13]. The measurement update is then given by

$$K_{k,i} = P_k^- C_{k,i}^\top (C_{k,i} P_k^- C_{k,i}^\top + R_k)^{-1} \tag{8.8}$$

$$\hat{\boldsymbol{x}}_{k,i+1}^+ = \hat{\boldsymbol{x}}_k^- + K_{k,i}(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k,i}^+) - C_{k,i}(\hat{\boldsymbol{x}}_k^- - \hat{\boldsymbol{x}}_{k,i}^+)) \tag{8.9}$$

$$P_{k,i+1}^+ = (I - K_{k,i} C_{k,i}) P_k^- (I - K_{k,i} C_{k,i})^\top + K_{k,i} R_k K_{k,i}^\top \tag{8.10}$$

for $i = 1, \ldots, N_{\mathrm{iekf}}$, where N_{iekf} is the number of measurement update iterations and $\hat{\boldsymbol{x}}_{k,1} = \hat{\boldsymbol{x}}_k^-$, $P_{k,1}^+ = P_k^-$. Note that for $N_{\mathrm{iekf}} = 1$ the IEKF reduces to the EKF.

8.3. Extended Kalman Smoother

The EKF in (8.3) provides state estimates up to and including (for the *a posteriori* estimate) measurements at time $t = t_k$. In principle, an estimate at $t = t_k$ can be further improved if future measurements at times $t > t_k$ were known. This information is, in fact, available, e.g., after an experiment/simulation has been conducted or (in a running experiment) for past estimates. This is known as a smoothing problem. To obtain a smoothed estimate, a second EKF is run backwards in time to calculate estimates based on future measurements w.r.t. a certain point in time. The forward and backwards estimates are then combined to yield a smoothed estimate. There are three types of smoothing, namely

1. fixed-point smoothing, where k is fixed and $N_y = k + 1, \dots$ is ongoing.
2. fixed-lag smoothing, where $k \in \mathbb{N}_0$ is ongoing and $N_y = k + L$, where L is the constant lag.
3. fixed-interval smoothing, where $k = 0, \dots, N_I$, with the interval length N_I.

In other words, the fixed-point smoother improves a single estimate for a fixed point in time while more and more measurements become available. The fixed-lag smoother utilizes measurements that are available in a time window of constant length to improve the estimate at the beginning of this window. The fixed-interval smoother improves every estimate in a window of constant length. Note that the fixed-lag and fixed-interval smoother are very similar and may be combined to yield improved estimates for every time instant within a fixed-lag interval.

The overall smoothed estimate relies on computing the optimal combination of the forward and backward filter estimates using the so-called fusion equations (see also [12, Sec. 12.10.4])

$$\breve{P}_k = \left[(P_k^f)^{-1} + (P_k^b)^{-1}\right]^{-1}, \tag{8.11}$$

where P_k^f is the error covariance of the (forward) EKF, i.e. (8.3), P_k^b is the error covariance of the backward filter, and $\breve{P}_k$ is the error covariance of the smoothed estimate. From this it is also evident that $\breve{P}_k \leq P_k^f$ in the sense of a matrix inequality, i.e., the smoothed error covariance must always be equal to or less than the forward error covariance. The smoothed state estimate is computed using

$$\breve{\boldsymbol{x}}_k = K_k^f \hat{\boldsymbol{x}}_k^f + (I - K_k^f)\hat{\boldsymbol{x}}_k^b \tag{8.12}$$

$$K_k^f = P_k^b (P_k^f + P_k^b)^{-1}. \tag{8.13}$$

See Simon [141, Sec. 9.4.1] for an in-depth discussion and derivation.

In the following, we are concerned with the fixed-interval smoothing procedure that can also run online, i.e., at the current time instant k the smoothed estimate is computed using the previous N estimates and, the corresponding measurements within that interval. To this end, the backward filter essentially consists of a EKF that runs backwards in time, i.e., from $j = k$ to $j = k - N$. However, the backwards filter must be initialized with values that are independent of the solution of the forward filter [141, Sec. 9.4.1]. Therefore, the backward filter is initialized with $P_j^- = \infty$, which is not feasible in a computational sense. Instead, the backward filter needs to be formulated as an information filter, i.e., the error covariance is replaced by its inverse, which is called the information matrix, i.e., $S_k^- = (P_k^-)^{-1} = 0$ [95, Sec. 2.8.1]. Accordingly, an auxiliary information state can be defined as $\boldsymbol{s}_j^- = S_j^- \hat{\boldsymbol{x}}_j^-$ such that $\boldsymbol{s}_k = \boldsymbol{0}$. With this, the information filter is given by the correction step

$$\boldsymbol{s}_j^+ = \boldsymbol{s}_j^- + C_j^\top R_j^{-1} \boldsymbol{y}_j \tag{8.14a}$$

$$S_j^+ = S_j^- + C_j^\top R_j^{-1} C_j \tag{8.14b}$$

and the prediction step

$$K_j^b = S_j^+ G (G^\top S_j^+ G + Q_{j-1}^{-1})^{-1} \tag{8.14c}$$

$$\boldsymbol{s}_j^- = A_{j-1}^\top (I - K_j^b G^\top)(\boldsymbol{s}_j^+ - S_j^+ B_{j-1} u_{j-1}) \tag{8.14d}$$

$$S_j^- = A_{j-1}^\top (I - K_j^b G^\top) S_j^+ A_{j-1}^\top \tag{8.14e}$$

for $j = k, \dots, k - N + 1$ [see 106, Sec. 8.4]. In order to obtain a smoothed estimate, three passes are necessary, i.e., the forward EKF pass (8.3), the backward EKF pass (8.14), and the smoothing pass (8.11), which imposes a significant computational burden. A much more efficient alternative is given by the Rauch-Tung-Striebel (RTS) smoother [128] that combines the backward and smoothing pass into a set of recursive equations, which only require saving the forward filter estimates and the corresponding error covariances (see also [56, Chapter 5], [141, Sec. 9.4.2])

$$\breve{K}_j = P_j^+ A_j^\top \left(P_{j+1}^-\right)^{-1} \tag{8.15a}$$

$$\breve{\boldsymbol{x}}_j = \hat{\boldsymbol{x}}_j^+ + \breve{K}_j (\breve{\boldsymbol{x}}_{j+1} - \hat{\boldsymbol{x}}_{j+1}^-) \tag{8.15b}$$

$$\breve{P}_j = P_j^+ + \breve{K}_j (\breve{P}_{j+1} - P_{j+1}^-) \breve{K}_j^\top \tag{8.15c}$$

for $j = k - 1, \dots, k - N$, which is initialized with $\breve{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^+$, $\breve{P}_k = P_k^+$.

Remark 5 (Handling Different Sampling Times) *The RTS algorithm is usually based on the assumption that the measurements have the same sampling time as the (forward) EKF* (8.3)*, i.e., $\Delta t_{ekf} = \Delta t_y$. However, if Δt_y is an (integer) multiple of Δt_{ekf}, applying the RTS algorithm* (8.15) *to obtain a smoothed estimate for all time steps corresponding to the forward filter, i.e., for each $t_k = k\Delta t_{ekf}, \quad k \in \mathbb{N}$ is not obvious since the measurements are only available at $t_y = i\Delta t_{ekf} = mk\Delta t_{ekf}$, where m is the (integer) multiple. This can be seen, e.g., in Fig. 8.1 for $k = 4$, where no measurement and, therefore, also no a posteriori state estimate is available, which is needed in* (8.15b)*. It is therefore the most straight-forward solution to use the a priori state estimate $\hat{\boldsymbol{x}}_{k|k-1} = \hat{\boldsymbol{x}}_k^-$ and covariance P_k^- of the forward filter* *as a proxy for* $\hat{\boldsymbol{x}}_{k+1}^-$ *and* P_k^+ *in* (8.15b) *and* (8.15a)-(8.15c)*, respectively. This corresponds to interpolating the smoothed solution between the available measurements as mentioned in Rauch et al. [128].*

8.4. Adaptive Noise Covariance Estimation

In the EKF equations (8.3), the principal DOFs are given by the process noise covariance matrix Q_k and the measurement noise covariance matrix R_k. Usually, the measurement noise covariance matrix R_k is known and depends on the used sensor. Moreover, it can be assumed that the measurement noise covariance matrix is constant over time. Determining the process noise covariance matrix Q_k, on the other hand, is not as straightforward and usually requires intricate knowledge of the system. In the following, we will show how the process noise covariance matrix Q_k can be estimated online in the EKF based on the innovations sequence that is the principal source of new information. An adaptive process is motivated by the fact that the system dynamics can change over time, e.g., due to aging of the system or due to changes in the environment. In the Kalman gain (8.3c), the denominator yields the theoretical covariance of the innovations matrix, i.e.,

$$E[\boldsymbol{\varepsilon}_k^-(\boldsymbol{\varepsilon}_k^-)^\top] = C_k P_k^- C_k^\top + R_k. \tag{8.16}$$

In this section, we will make use of the unbiased estimate of the innovations covariance matrix, i.e., the sampled innovations covariance matrix $\hat{\Sigma}_{k|N}^-$ see, e.g., [103, Sec. 3.3.2] that can be calculated with

$$E[\boldsymbol{\varepsilon}_k^-(\boldsymbol{\varepsilon}_k^-)^\top] \approx \hat{\Sigma}_{k|N}^- = \frac{1}{N} \sum_{i=k-N+1}^{k} (\boldsymbol{\varepsilon}_i^- - \overline{\boldsymbol{\varepsilon}})(\boldsymbol{\varepsilon}_i^- - \overline{\boldsymbol{\varepsilon}})^\top, \tag{8.17}$$

where $\overline{\boldsymbol{\varepsilon}}$ is the mean of the innovations sequence, and N is the number of samples in the window. Making use of the fact that the innovations sequence has zero mean for a correctly working EKF (see [108, Sec. 5], [54, Sec. 4.4.1]), (8.17) can be simplified to

$$\hat{\Sigma}_{k|N}^- = \frac{1}{N} \sum_{i=k-N+1}^{k} \boldsymbol{\varepsilon}_i^-(\boldsymbol{\varepsilon}_i^-)^\top. \tag{8.18}$$

Note that the sample innovations covariance matrix can be updated iteratively, i.e.,

$$\hat{\Sigma}_{k|N}^- = \hat{\Sigma}_{k-1|N}^- + \frac{1}{N}\left(\boldsymbol{\varepsilon}_k^-(\boldsymbol{\varepsilon}_k^-)^\top - \boldsymbol{\varepsilon}_{k-N+1}^-(\boldsymbol{\varepsilon}_{k-N+1}^-)^\top\right), \tag{8.19}$$

for $k \in \mathbb{N}_0$ and where $\hat{\Sigma}_{-1|N}^-$ is the initial sampled innovations covariance matrix typically chosen to be $\hat{\Sigma}_{-1|N}^- = \boldsymbol{\varepsilon}_0^-(\boldsymbol{\varepsilon}_0^-)^\top$. This is essentially a moving average filter for the innovations covariance matrix. Alternatively, an exponentially-weighted moving average filter can be used, i.e.,

$$\hat{\Sigma}_{k|N}^- = \alpha \boldsymbol{\varepsilon}_k^-(\boldsymbol{\varepsilon}_k^-)^\top + (1-\alpha)\hat{\Sigma}_{k-1|N}^-, \tag{8.20}$$

Note that the latter takes into account all observations $\boldsymbol{\varepsilon}_k,\ k \in \mathbb{N}_0$ while the former only takes into account the last N observations $\boldsymbol{\varepsilon}_{k-N+1}, \dots, \boldsymbol{\varepsilon}_k$. However, the exponentially-weighted moving average has the advantage that no past observations need to be stored in memory. The moving average and exponentially-weighted moving average have the same average age of the samples (see [113, Sec. 2.7]) when

$$\alpha = \frac{2}{N+1}. \tag{8.21}$$

In order to make use of (8.18) with regard to process noise covariance estimation, (8.2) is rearranged and the time index shifted, so that

$$G\boldsymbol{w}_{k-1} = \boldsymbol{x}_k - \boldsymbol{F}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}), \tag{8.22}$$

and, in view of (8.3a) and (8.3d), we can construct an estimation of the process noise as

$$\begin{aligned} G\hat{\boldsymbol{w}}_{k-1} &= \hat{\boldsymbol{x}}_k^+ - \boldsymbol{F}(\hat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}_{k-1}) && (8.23a)\\ &= \hat{\boldsymbol{x}}_k^+ - \hat{\boldsymbol{x}}_k^- && (8.23b)\\ &= K_k(y_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-)) && (8.23c)\\ &= K_k\boldsymbol{\varepsilon}_k^- && (8.23d) \end{aligned}$$

such that $\hat{\boldsymbol{w}}_{k-1} = G^\dagger K_k \boldsymbol{\varepsilon}_k^-$, where $G^\dagger = (G^\top G)^{-1}G^\top$ is the left pseudo-inverse of G. With this and taking into account (8.17), the process noise covariance matrix can be estimated with

$$\begin{aligned} \hat{Q}_{k-1} = E[\hat{\boldsymbol{w}}_{k-1}\hat{\boldsymbol{w}}_{k-1}^\top] &= E[G^\dagger K_k \boldsymbol{\varepsilon}_k^- (\boldsymbol{\varepsilon}_k^-)^\top K_k^\top (G^\dagger)^\top] && (8.24a)\\ &= G^\dagger K_k E[\boldsymbol{\varepsilon}_k^- (\boldsymbol{\varepsilon}_k^-)^\top] K_k^\top (G^\dagger)^\top && (8.24b)\\ &= G^\dagger K_k \hat{\Sigma}_{k|N}^- K_k^\top (G^\dagger)^\top. && (8.24c) \end{aligned}$$

If the process noise covariance matrix is known *a priori*, an estimate of the measurement noise covariance matrix can be obtained by rearranging (8.16), i.e.,

$$\hat{R}_k = \hat{\Sigma}_{k|N}^- - C_k P_k^- C_k^\top, \tag{8.25}$$

however, this estimate does not guarantee that $\hat{R}_k$ is positive definite. Therefore, it can be shown that

$$\hat{R}_k = \hat{\Sigma}_{k|N}^+ + C_k P_k^+ C_k^\top, \tag{8.26}$$

yields better estimates, where $\hat{\Sigma}_{k|N}^+ = {}^1\!/\!{}_N \sum_{i=k-N+1}^{k} \boldsymbol{\varepsilon}_i^+ (\boldsymbol{\varepsilon}_i^+)^\top$ is the sample residual covariance matrix, and $\boldsymbol{\varepsilon}_i^+ = \boldsymbol{y}_i - \boldsymbol{h}(\hat{\boldsymbol{x}}_i^+)$ is the residual [110, Sec 3.2]. Another alternative to (8.19) and (8.20) is to make use of the Kalman smoother for the calculation of the sampled innovations covariance matrix [54]. This is motivated by the fact that, in principle, if the sample innovations covariance matrix is not updated iteratively using (8.19) a fixed-length memory is required to store the innovations. Therefore, a further improvement in accuracy can be obtained by utilizing the smoothed innovations in (8.18), i.e.,

$$\hat{\Sigma}^{\breve{}}_{k|N} = \frac{1}{N} \sum_{i=k-N+1}^{k} \breve{\boldsymbol{\varepsilon}}_i \breve{\boldsymbol{\varepsilon}}_i^{\top} = \frac{1}{N} \sum_{i=k-N+1}^{k} (\boldsymbol{y}_i - \boldsymbol{h}(\breve{\boldsymbol{x}}_i))(\boldsymbol{y}_i - \boldsymbol{h}(\breve{\boldsymbol{x}}_i))^{\top}, \qquad (8.27)$$

see also Fig. 8.1.

Remark 6 (Simultaneous Q **and** R **estimation)** *It is important to note that the process and measurement noise covariance matrices are not independent of each other. Therefore, the adaptive estimation techniques presented in this section are not suitable for simultaneous estimation of both the process and measurement noise covariance matrix see Mohamed and Schwarz [110], Fraser [54, Sec. 3.2], and Maybeck [106, Sec. 10.7]. Also note that similar results can be obtained by deriving the adaptive estimation of the process and measurement noise covariance matrices by maximizing the maximum likelihood function of the respective unknown covariance matrix given the measurement sequence [see 106].*

Part III.

Simulative Studies

9. Dynamic Positioning

In this chapter a dynamic positioning problem is developed using a nonlinear feedback linearization according to Sec. 6.1 with control allocation according to Sec. 6.2 and extended with an AIEKF-based state and disturbance estimation according to Chapter 8. To this end, the nonlinear feedback controller calculates the control input $\boldsymbol{\tau}_{\mathrm{fb}}$ that stabilizes the origin of the inertial frame and its poles are chosen such that the control effort does not exceed their respective maximum values. For the combined state and disturbance estimation, the dynamic vessel model (5.74) is extended to take into account disturbances acting on the ship. The disturbances are usually modeled in the inertial frame, where they are assumed to be constant or slowly time varying with perturbations driven by a Gaussian white noise random process. In the following, we will distinguish between two different disturbance and measurement models. The first model is called the lumped disturbance model that represents all disturbances using a single lumped disturbance vector $\boldsymbol{d}$. The second model extends the lumped model using the wind model (5.56) to explicitly take into account knowledge of the wind effects and which enables a distinction between wind disturbances and any other disturbances arising from different sources. Here, it is assumed that the apparent wind velocity and direction can be measured using an anemometer. The estimated disturbance is taken into account in the high level controller by means of a feedforward control that compensates for the disturbance. Finally, a simulative validation of the proposed concepts is presented and discussed.

9.1. Problem Formulation

In the following, the individual parts of the dynamic positioning problem are presented and applied to the MSV, namely, the underlying model is discussed, followed by a synthesis of the feedback linearization controller. Subsequently, the control allocation problem is defined, finally, the state and disturbance estimation problem is formulated using the AIEKF. The latter is presented for two different measurement models, namely, the lumped disturbance model, where the vessel's pose is measured, and the combined lumped and wind disturbance model that explicitly takes into account the wind when apparent wind direction and speed is measured additionally to the pose. All parameters regarding the simulation, the controller, the control allocation, and the estimator are summarized in Tab. 9.2 and Tab. 9.3 for the first and second estimator and measurement model, respectively.

9.1.1. Vessel Model

The vessel model used in this simulative study is the Cybership II taken from Lindegaard [97], Skjetne [142] since it provides a fully parameterized, nonlinear 3DOF maneuvering model in the form (5.74) with a propulsion model that consists of two propeller-rudder systems and a bow thruster, i.e., the vector of control surfaces is given by $\boldsymbol{u} = [\omega_{\mathrm{pr,l}}, \delta_{\mathrm{pr,l}}, \omega_{\mathrm{pr,r}}, \delta_{\mathrm{pr,r}}, \omega_{\mathrm{tt}}]^\top$, where $\omega_{\mathrm{pr,l}}$ and $\omega_{\mathrm{pr,r}}$ are the rotational speeds of the left and right propeller, respectively, $\delta_{\mathrm{pr,l}}$ and $\delta_{\mathrm{pr,r}}$ are the rudder angles of the left and right rudder, respectively, and ω_{tt} is the rotational speed of the bow (tunnel) thruster. The individual models of the thrusters are given in Sec. 5.2.4. All relevant geometric, hydrodynamic, and propulsion parameters of the Cybership model are summarized in Tab. 9.1.

Group	Symbol	Value	Symbol	Value	Symbol	Value
General	L_{oa}	1.2	m	23.8		
	B	0.5	J_{zz}	1.76		
	x_g	0.046				
Added Mass	$X_{\dot{u}}$	−2.0				
	$Y_{\dot{v}}$	−10.0	$Y_{\dot{r}}$	0.0		
	$N_{\dot{v}}$	0.0	$N_{\dot{r}}$	−1.0		
Damping	X_u	−0.72	$X_{\lvert u\rvert u}$	−1.33	X_{uuu}	−5.87
	Y_v	−0.89	$Y_{\lvert v\rvert v}$	−36.47	$Y_{\lvert v\rvert r}$	0.08
	Y_r	0.11	$Y_{\lvert r\rvert v}$	−0.81	$Y_{\lvert r\rvert r}$	−3.45
	N_v	0.03	$N_{\lvert v\rvert v}$	3.96	$N_{\lvert v\rvert r}$	0.08
	N_r	−1.9	$N_{\lvert r\rvert v}$	0.13	$N_{\lvert r\rvert r}$	−0.75
Propeller-Rudder System(s)	$l_{x,\mathrm{pr}}$	−0.549	$T^+_{\lvert\omega_{\mathrm{pr}}\rvert\omega_{\mathrm{pr}}}$	3.65e−3	$L^+_{\delta_{\mathrm{pr}}}$	6.43
	$l_{y,\mathrm{pr}}$	±0.078	$T^+_{\lvert\omega_{\mathrm{pr}}\rvert u}$	1.52e−4	$L^+_{\lvert\delta_{\mathrm{pr}}\rvert\delta_{\mathrm{pr}}}$	5.84
	k_u	0.5	$T^-_{\lvert\omega_{\mathrm{pr}}\rvert\omega_{\mathrm{pr}}}$	5.10e−3	$L^-_{\delta_{\mathrm{pr}}}$	3.20
	d_{pr}	0.06	$T^-_{\lvert\omega\rvert_{\mathrm{pr}}u}$	4.56e−2	$L^-_{\lvert\delta_{\mathrm{pr}}\rvert\delta_{\mathrm{pr}}}$	2.34
	ω^+_{pr}	33	ω^-_{pr}	−33	$\dot{\omega}^{\max}_{\mathrm{pr}}$	15
	δ^+_{pr}	$35\,\frac{\pi}{180}$	δ^-_{pr}	$-35\,\frac{\pi}{180}$	$\dot{\delta}^{\max}_{\mathrm{pr}}$	$7\,\frac{\pi}{180}$
Bow (Tunnel) Thruster	$l_{x,\mathrm{tt}}$	0.466	$T_{\lvert\omega_{\mathrm{tt}}\rvert\omega_{\mathrm{tt}}}$	1.57e−4		
	$l_{y,\mathrm{tt}}$	0				
	ω^+_{tt}	83	ω^-_{tt}	−83	$\dot{\omega}^{\max}_{\mathrm{tt}}$	25

Table 9.1.: Parameters of the Cybership II MSV in (unscaled) SI units according to Lindegaard [97] and Skjetne [142]. Note that there's two propeller rudder systems, one on each side of the ship.

9.1.2. Nonlinear Feedback Controller

The dynamic positioning problem is solved based on a nonlinear feedback linearization as presented in Sec. 6.1. The task of dynamic positioning usually requires but is not limited to a static reference trajectory of the form

$$\boldsymbol{\eta}^*(t) = \begin{bmatrix} x^*(t) \\ y^*(t) \\ \psi^*(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{9.1}$$

such that the higher order derivatives of the reference trajectory (6.13), i.e.,

$$\boldsymbol{\xi}^* = \begin{bmatrix} x^* \\ \dot{x}^* \\ \vdots \\ x^{*(r_1-1)} \end{bmatrix} = \mathbf{0}, \quad \boldsymbol{\Upsilon}^* = \begin{bmatrix} y^* \\ \dot{y}^* \\ \vdots \\ y^{*(r_2-1)} \end{bmatrix} = \mathbf{0}, \quad \boldsymbol{\Psi}^* = \begin{bmatrix} \psi^* \\ \dot{\psi}^* \\ \vdots \\ \psi^{*(r_3-1)} \end{bmatrix} = \mathbf{0}, \tag{9.2}$$

for $r_1 = r_2 = r_3 = 2$, which holds without loss of generality, i.e., is not limited to the origin of the NED frame. The poles of the stabilizing linear controller

$$\boldsymbol{v} = \begin{bmatrix} x^{*(r_1)} - \boldsymbol{p}_1^\top (\boldsymbol{\Phi}_1(\boldsymbol{x}) - \boldsymbol{\xi}^*) \\ y^{*(r_2)} - \boldsymbol{p}_2^\top (\boldsymbol{\Phi}_2(\boldsymbol{x}) - \boldsymbol{\Upsilon}^*) \\ \psi^{*(r_3)} - \boldsymbol{p}_3^\top (\boldsymbol{\Phi}_3(\boldsymbol{x}) - \boldsymbol{\Psi}^*) \end{bmatrix}, \tag{9.3}$$

are chosen such that the force limits of the Cybership II vessel are not exceeded [142, Sec. 5.2.1]. Since the feedback linearization decouples the individual DOFs, the pole placement is directly linked to the transient behavior of the respective DOF. Thus, a slight preference of the heading dynamics over the surge and sway dynamics is expressed by choosing the corresponding poles further to the left in the complex plane compared with the poles of the decoupled surge and sway dynamics which, in principle, should result in a faster convergence of the heading angle.

9.1.3. Control Allocation

The control allocation problem is solved using the linearization of the propulsion model of the Cybership II in every iteration, which results in a quadratic optimization problem as discussed in Sec. 6.2. However, as discussed in Härkegård [67, Sec. 7.2.3], linearization of the nonlinear propulsion model can result in erroneous behavior of the control allocation when the relationship between any control surface and the resulting thrust is non-monotonic. In this particular case, the solution of the underlying quadratic optimization problem may not reflect the actual optimal solution. As a safeguard, we keep track of the accuracy of the linearization and, in case it fails to provide a sensible output, the control allocation is solved using the nonlinear propulsion model instead using

`SNOPT`. The input weighting matrix is set to $W_\tau = \operatorname{diag}(1, 10, 10)$, which is motivated by the fact that it is more important to achieve the commanded sway force and the yaw moment than the surge force. Furthermore, the controls weighting matrix is chosen as $W_{\boldsymbol{\tau}_\mathrm{c}^*} = \operatorname{diag}(1, 4, 1, 4, 2)$, which is motivated by the fact that the rudders should be kept in place as much as possible.

	Name	Symbol	Value
Simulation	Model	–	Cybership II, see Tab.9.1
	Initial state	$\hat{\boldsymbol{x}}_0$	$[-1,\ -1,\ 30\frac{\pi}{180},\ 0,\ 0,\ 0]^\top$
	Final time	T	35
	True wind speed	$V_{\mathrm{w,t}}$	2
	True wind angle of attack	$\beta_{\mathrm{w,t}}$	$-\frac{\pi}{2}$
	Pose measurement noise standard deviation	$\sqrt{R_{\mathrm{c},\eta}}$	1e−2 diag$(5,\ 5,\ 5)$
Controller	Control law	–	(6.11), (6.12)
	Desired pose	$\boldsymbol{\eta}^*$	$[0,\ 0,\ 0]^\top$
	Desired north position poles	λ^*_x	$-[0.2,\ 0.2]$
	Desired east position poles	λ^*_y	$-[0.2,\ 0.2]$
	Desired heading poles	λ^*_ψ	$-[0.6,\ 0.7]$
	Relative degree	$[r_1,\ r_2,\ r_3]$	$[2,\ 2,\ 2]$
Control Allocation	Input weighting matrix	W_τ	diag$(1,\ 10,\ 10)$
	Controls weighting matrix	W_u	diag$(1,\ 4,\ 1,\ 4,\ 2)$
	Initial controls	$\boldsymbol{u}_0$	$[15,\ 0.1\frac{\pi}{180},\ 15,\ 0.1\frac{\pi}{180},\ -10]^\top$
	Reference Values	$\boldsymbol{u}_{\mathrm{ref},k}$	$[\omega_{\mathrm{pr,l},k-1},\ 0,\ \omega_{\mathrm{pr,r},k-1},\ 0,\ \omega_{\mathrm{tt},k-1}]^\top$
	Control allocation weight	γ	1e4
Estimator	Model	–	Lumped, see (9.4)
	Discretization time[a]	Δt_{ekf}	1e−2
	Discrete propagation function	$\boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k)$	Euler forward
	Measurement sample time	Δt_y	$5\Delta t_{\mathrm{ekf}}$
	Initial state error standard deviation	$\sqrt{P_{0,x}}$	1e−3 diag$(1,\ 1,\ 1,\ 1,\ 1,\ 5)$
	Initial disturbance error standard deviation	$\sqrt{P_{0,\mathrm{d}}}$	1e−3diag$(5,\ 5,\ 5)$
	Initial state estimate	$\hat{\boldsymbol{x}}^+_{-1}$	$\mathcal{N}(\hat{\boldsymbol{x}}_0,\ P_{0,x})$
	Initial disturbance estimate	$\hat{\boldsymbol{d}}^+_{-1}$	$\mathcal{N}(\boldsymbol{g}(\hat{\boldsymbol{x}}_0, \boldsymbol{u}_0),\ P_{0,\mathrm{d}})$
	Process noise gain	G_ν	M^{-1}
	Disturbance process noise gain	G_{d}	diag$(1,\ 1,\ 1)$
	Process noise standard deviation	$\sqrt{Q_{c,\nu}}$	1e−2 diag$(1, 1, 2.5)$
	Disturbance process noise standard deviation	$\sqrt{Q_{c,\mathrm{d}}}$	1e−2 diag$(5,\ 5,\ 0.5)$
	Pose measurement noise standard deviation	$\sqrt{R_{\mathrm{c},\eta}}$	1e−2 diag$(10,\ 10,\ 8.83)$
	Adaptation method	Exponentially-weighted moving average	
	Adaptation time window[b]	t_{s}	1
	Number of measurement update iterations	N_{iekf}	2

Table 9.2.: Scenario parameters for the lumped disturbance estimation using the Cybership model with a nonlinear feedback controller and an adaptive EKF. All units are SI. All covariance matrices are given in their square root form, i.e., they are represented using their respective standard deviations and, thus, preserve the unit of the underlying quantity.

9.1.4. State and Disturbance Estimation

The AIEKF discussed in Sec. 8 is used to estimate the state vector $\boldsymbol{x}$ and all disturbances. To estimate the disturbance, all disturbances are lumped into the vector $\boldsymbol{d} = [d_X,\ d_Y,\ d_N]^\top$, which is expressed in the NED frame where it is usually assumed to be time invariant or slowly changing [47]. Therefore, the disturbance acting on the ship is transformed to the body frame using $\boldsymbol{\tau}_\mathrm{d} = R_{z,\psi}^\top \boldsymbol{d}$, which is added to the nominal model [50]. In the following, this approach is discussed in detail. Furthermore, an extension is presented when wind measurements are available that distinguishes disturbances caused by wind and all other disturbances are lumped into the vector $\boldsymbol{d}$. In both these approaches the resulting estimate for the disturbance vector $\boldsymbol{d}$ or $\boldsymbol{\tau}_\mathrm{d}$ is used to compensate the disturbance by adding it to the feedback control input $\boldsymbol{\tau}_\mathrm{fb}$, i.e., $\boldsymbol{\tau}_\mathrm{c}^* = \boldsymbol{\tau}_\mathrm{fb} + \hat{\boldsymbol{\tau}}_\mathrm{d}$, where $\hat{\boldsymbol{\tau}}_\mathrm{d} = -R_{z,\hat{\psi}}^\top \hat{\boldsymbol{d}}$. The high level controller does not include any integral action, which, in turn, results in an offset from the desired steady state. In both approaches, we make use of the adaptive measurement noise covariance estimation technique to improve the performance. Additionally, the iterated EKF variant is used, which results in a higher-order EKF. The EKF sample time is chosen as $\Delta t_\mathrm{ekf} = 0.01\,\mathrm{s}$ and the measurement sample time is chosen as $\Delta t_\mathrm{meas} = 5\Delta t_\mathrm{ekf}$. The discrete propagation function is based on the forward Euler method and initial estimator values are chosen based on the true values with initial errors added based on the respective initial error covariance matrices. The initial measurement noise covariance matrices are initialized with a larger deviation from their true values to emphasize the convergence of the adaptive measurement noise covariance estimation technique.

Lumped Disturbance Model

In the lumped disturbance model, the vessel model (5.74) is extended with the lumped disturbance vector $\boldsymbol{d}$ and the EKF model is represented mathematically using

$$\dot{\boldsymbol{\eta}} = R_{z,\psi}\boldsymbol{\nu}, \tag{9.4a}$$

$$\dot{\boldsymbol{\nu}} = -M^{-1}(C(\boldsymbol{\nu}) + D(\boldsymbol{\nu}))\boldsymbol{\nu} + M^{-1}\boldsymbol{\tau}_\mathrm{c} + M^{-1}R_{z,\psi}^\top \boldsymbol{d} + G_\nu \boldsymbol{w}_\nu, \qquad \boldsymbol{w}_\nu \sim \mathcal{N}(\boldsymbol{0}, Q_{\mathrm{c},\nu}), \tag{9.4b}$$

$$\dot{\boldsymbol{d}} = G_\mathrm{d}\boldsymbol{w}_\mathrm{d}, \qquad \boldsymbol{w}_\mathrm{d} \sim \mathcal{N}(\boldsymbol{0}, Q_\mathrm{c,d}), \tag{9.4c}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}_\mathrm{ext}) + \boldsymbol{v}_\eta = \boldsymbol{\eta} + \boldsymbol{v}_\eta, \qquad \boldsymbol{v}_\eta \sim \mathcal{N}(\boldsymbol{0}, R_{\mathrm{c},\eta}). \tag{9.4d}$$

Therein the process noise weighting matrices G_ν and G_d are chosen based on [35]. The extended observer model (9.4) can be compactly written as

$$\dot{\boldsymbol{x}}_\mathrm{ext} = \boldsymbol{f}_\mathrm{ext}(\boldsymbol{x}_\mathrm{ext}) + B_{\tau,\mathrm{ext}}\boldsymbol{u} + G\boldsymbol{w}_\mathrm{ext}, \qquad \boldsymbol{w}_\mathrm{ext} \sim \mathcal{N}(\boldsymbol{0}, Q_\mathrm{c,ext}) \tag{9.5a}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}_\mathrm{ext}) + \boldsymbol{v}_\eta, \qquad \boldsymbol{v}_\eta \sim \mathcal{N}(\boldsymbol{0}, R_{\mathrm{c},\eta}), \tag{9.5b}$$

where $\boldsymbol{x}_\mathrm{ext} = [\boldsymbol{x}^\top,\ \boldsymbol{d}^\top]^\top$ is the observer state, $\boldsymbol{w}_\mathrm{ext} = [\boldsymbol{w}_\nu^\top,\ \boldsymbol{w}_\mathrm{d}^\top]^\top$ is the process noise, $Q_\mathrm{c,ext} = \mathrm{diag}(Q_{c,\nu},\ Q_{c,\mathrm{d}})$ is the process noise covariance matrix of the extended system, $\boldsymbol{v}_\eta$

	Name	Symbol	Value
Simulation	Model	–	Cybership II, see Tab.9.1
	Initial state	$\hat{\boldsymbol{x}}_0$	$[-1,\ -1,\ 30\frac{\pi}{180},\ 0,\ 0,\ 0]^\top$
	Final time	T	35
	True wind speed	$V_{\mathrm{w,t}}$	2
	True wind angle of attack	$\beta_{\mathrm{w,t}}$	$-\frac{\pi}{2}$
	Pose measurement standard deviation	$\sqrt{R_{\mathrm{c},\eta}}$	$1\mathrm{e}{-2}\,\mathrm{diag}(5,\ 5,\ 5)$
	Wind measurement standard deviation	$\sqrt{R_{\mathrm{c,w}}}$	$1\mathrm{e}{-2}\,\mathrm{diag}(10,\ 5)$
Estimator	Model	Combined lumped and wind, see (9.10)	
	Discretization time	Δt_{ekf}	$1\mathrm{e}{-2}$
	Discrete propagation function	$\boldsymbol{F}(\boldsymbol{x}_k,\ \boldsymbol{u}_k)$	Euler forward
	Measurement sample time	Δt_y	$5\Delta t_{\mathrm{ekf}}$
	Initial state error standard deviation	$\sqrt{P_{0,x}}$	$1\mathrm{e}{-3}\,\mathrm{diag}(1,\ 1,\ 1,\ 1,\ 1,\ 5)$
	Initial disturbance error standard deviation	$\sqrt{P_{0,\mathrm{d}}}$	$1\mathrm{e}{-4}\mathrm{diag}(5,\ 5,\ 5)$
	Initial wind error standard deviation	$\sqrt{P_{0,\mathrm{w}}}$	$1\mathrm{e}{-3}\mathrm{diag}(5,\ 5)$
	Initial state estimate	$\hat{\boldsymbol{x}}_{-1}^{+}$	$\mathcal{N}(\hat{\boldsymbol{x}}_0,\ P_{0,x})$
	Initial disturbance estimate	$\hat{\boldsymbol{d}}_{-1}^{+}$	$\mathcal{N}(\boldsymbol{0},\ P_{0,\mathrm{d}})$
	Initial wind estimate	$\hat{\boldsymbol{x}}_{-1,\mathrm{w}}^{+}$	$\mathcal{N}([V_{\mathrm{w,t}},\ \beta_{\mathrm{w,t}}]^\top,\ P_{0,\mathrm{d}})$
	Process noise gain	G_ν	M^{-1}
	Disturbance process noise gain	G_{d}	$\mathrm{diag}(1,\ 1,\ 1)$
	Wind process noise gain	G_{w}	$\mathrm{diag}(1,\ 1)$
	Process noise standard deviation	$\sqrt{Q_{c,\nu}}$	$1\mathrm{e}{-2}\,\mathrm{diag}(1, 1, 2.5)$
	Disturbance process noise standard deviation	$\sqrt{Q_{c,\mathrm{d}}}$	$1\mathrm{e}{-2}\,\mathrm{diag}(5,\ 5,\ 0.5)$
	Wind process noise standard deviation	$\sqrt{Q_{c,\mathrm{w}}}$	$1\mathrm{e}{-2}\,\mathrm{diag}(5,\ 5)$
	Pose measurement noise standard deviation	$\sqrt{R_{\mathrm{c},\eta}}$	$1\mathrm{e}{-2}\,\mathrm{diag}(10,\ 10,\ 8.83)$
	Wind measurement noise standard deviation	$\sqrt{R_{\mathrm{c,w}}}$	$1\mathrm{e}{-2}\,\mathrm{diag}(10, 5)$
	Adaptation method	Exponentially-weighted moving average	
	Adaptation time window	t_{s}	1
	Number of measurement update iterations	N_{iekf}	2

Table 9.3.: Scenario parameters for the combined lumped and wind disturbance estimation using the Cybership II model with a nonlinear feedback controller and an adaptive EKF. All units are SI. All covariance matrices are given in their square root form, i.e., they are represented using their respective standard deviations and, thus, preserve the unit of the underlying quantity. Controller and control allocation parameters are identical to the lumped estimation scenario described in Tab. 9.2.

is the measurement noise, and $R_{\mathrm{c},\eta}$ is the corresponding measurement noise covariance matrix. Furthermore,

$$\boldsymbol{f}_{\mathrm{ext}}(\boldsymbol{x}_{\mathrm{ext}}) = \left[\begin{array}{c} \boldsymbol{f}(\boldsymbol{x}) \\ \hdashline \boldsymbol{0}^{(3\times1)} \end{array}\right] + \underbrace{\left[\begin{array}{c} \boldsymbol{0}^{(3\times1)} \\ M^{-1}R_{z,\psi}^{\top}\boldsymbol{d} \\ \hdashline \boldsymbol{0}^{(3\times1)} \end{array}\right]}_{\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}_{\mathrm{ext}})},\ B_{\tau,\mathrm{ext}} = \left[\begin{array}{c} B_\tau \\ \hdashline 0^{(3\times3)} \end{array}\right],\ G = \begin{bmatrix} 0^{(3\times6)} \\ \mathrm{diag}(G_\nu,\ G_{\mathrm{d}}) \end{bmatrix} \tag{9.6}$$

are the drift vector field with disturbance model $\boldsymbol{f}_\mathrm{d}(\boldsymbol{x}_\mathrm{ext})$, the input matrix with

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} R_{z,\psi}\boldsymbol{\nu} \\ -M^{-1}(C(\boldsymbol{\nu}) + D(\boldsymbol{\nu}))\boldsymbol{\nu} \end{bmatrix}, \quad B_\tau = \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix}, \tag{9.7}$$

defined in (5.75), and the process noise gain matrix of the extended system, respectively. The latter are chosen based on [35].

The observability of this system can be analyzed locally, i.e., based on the rank of the Jacobian of the observability map[1], where

$$\boldsymbol{q}(\boldsymbol{x}_\mathrm{ext}) = \begin{bmatrix} L^0_{\boldsymbol{f}_\mathrm{ext}} \boldsymbol{h}(\boldsymbol{x}_\mathrm{ext}) \\ L^1_{\boldsymbol{f}_\mathrm{ext}} \boldsymbol{h}(\boldsymbol{x}_\mathrm{ext}) \\ L^2_{\boldsymbol{f}_\mathrm{ext}} \boldsymbol{h}(\boldsymbol{x}_\mathrm{ext}) \end{bmatrix} = \begin{bmatrix} L^0_{\boldsymbol{f}_\mathrm{ext}} \boldsymbol{\eta}(t) \\ L^1_{\boldsymbol{f}_\mathrm{ext}} \boldsymbol{\eta}(t) \\ L^2_{\boldsymbol{f}_\mathrm{ext}} \boldsymbol{\eta}(t) \end{bmatrix} \in \mathbb{R}^9, \quad O(\boldsymbol{x}_\mathrm{ext}) = \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_\mathrm{ext}} \in \mathbb{R}^{9\times 9} \tag{9.8}$$

such that

$$\begin{aligned} \det O(\boldsymbol{x}_\mathrm{ext}) = \big[&(X_{\dot{u}} - m)\,(J_{zz}\,Y_{\dot{v}} + m^2\,{x_g}^2 - N_{\dot{r}}\,Y_{\dot{v}} - J_{zz}\,m \\ &+ N_{\dot{r}}\,m + {N_{\dot{v}}}^2 - 2\,N_{\dot{v}}\,m\,x_g)\big]^{-1}, \end{aligned} \tag{9.9}$$

is nonzero $\forall t \in \mathbb{R}$. Hence (9.6) is locally observable. Note that the (local) observability is not required by the EKF.

Since the EKF relies on Jacobian information of the observer model, the Jacobian of the extended drift vector field $\boldsymbol{f}_\mathrm{ext}(\boldsymbol{x}_\mathrm{ext})$ is needed, which can be found in Appendix A.3.

Combined Lumped and Wind Disturbance Model

Alternatively, the wind model as described in (5.56) can be incorporated into the disturbance estimator model as

$$\dot{\boldsymbol{\eta}} = R_{z,\psi}\boldsymbol{\nu}, \tag{9.10a}$$

$$\dot{\boldsymbol{\nu}} = -M^{-1}(C(\boldsymbol{\nu}) + D(\boldsymbol{\nu}))\boldsymbol{\nu} + M^{-1}\boldsymbol{\tau}_\mathrm{c} + M^{-1}\boldsymbol{\tau}_\mathrm{w}(\boldsymbol{x}, \boldsymbol{x}_\mathrm{w}) + M^{-1}R^\top_{z,\psi}\boldsymbol{d} + G_\nu \boldsymbol{w}_\nu, \tag{9.10b}$$

$$\dot{\boldsymbol{d}} = G_\mathrm{d}\boldsymbol{w}_\mathrm{d}, \tag{9.10c}$$

$$\dot{\boldsymbol{x}}_\mathrm{w} = G_\mathrm{w}\boldsymbol{w}_\mathrm{w}, \tag{9.10d}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}_\mathrm{ext}) = \boldsymbol{\eta} + \boldsymbol{v}_\eta \tag{9.10e}$$

$$\boldsymbol{y}_\mathrm{w} = \begin{bmatrix} V_\mathrm{w,a} \\ \gamma_\mathrm{w,a} \end{bmatrix} = \boldsymbol{h}_\mathrm{w}(\boldsymbol{x}_\mathrm{ext}) + \boldsymbol{v}_\mathrm{w} \tag{9.10f}$$

[1] Note that for nonlinear multiple input multiple output systems, the choice of the observability map is not unique but rather a DOF see, e.g., [see 77].

with

$$\boldsymbol{h}_{\mathrm{w}}(\boldsymbol{x}_{\mathrm{ext}}) = \begin{bmatrix} \sqrt{V_{\mathrm{w,t}}^2 + u^2 + v^2 + 2V_{\mathrm{w,t}}(u\cos(\beta_{\mathrm{w,t}} - \psi) + v\sin(\beta_{\mathrm{w,t}} - \psi))} \\ \mathrm{atan2}(\dot{y} + V_{\mathrm{w,t}}\sin(\beta_{\mathrm{w,t}}),\ \dot{x} + V_{\mathrm{w,t}}\cos(\beta_{\mathrm{w,t}})) - \psi \end{bmatrix}, \tag{9.11}$$

and $\dot{x} = \cos(\psi)u - \sin(\psi)v$, and $\dot{y} = \sin(\psi)u + \cos(\psi)v$, i.e., it is assumed that the apparent wind speed and direction can be measured, e.g., using an anemometer. Furthermore, $\boldsymbol{x}_{\mathrm{w}} = [V_{\mathrm{w,t}},\ \beta_{\mathrm{w,t}}]^\top$, i.e., the true respective wind speed and direction are combined in the additional wind observer state such that $\boldsymbol{x}_{\mathrm{ext}} = [\boldsymbol{x}^\top,\ \boldsymbol{d}^\top,\ \boldsymbol{x}_{\mathrm{w}}^\top]^\top$ is the total observer state. Additionally, $\boldsymbol{w}_\nu \sim \mathcal{N}(\boldsymbol{0}, Q_{\mathrm{c},\nu})$, $\boldsymbol{w}_{\mathrm{d}} \sim \mathcal{N}(\boldsymbol{0}, Q_{\mathrm{c,d}})$, $\boldsymbol{w}_{\mathrm{w}} \sim \mathcal{N}(\boldsymbol{0}, Q_{\mathrm{c,w}})$, $\boldsymbol{v}_\eta \sim \mathcal{N}(\boldsymbol{0}, R_{\mathrm{c},\eta})$, $\boldsymbol{v}_{\mathrm{w}} \sim \mathcal{N}(\boldsymbol{0}, R_{\mathrm{c,w}})$, and recall that the wind force is calculated using (5.56), i.e.,

$$\boldsymbol{\tau}_{\mathrm{w}}(\boldsymbol{x}, \boldsymbol{x}_{\mathrm{w}}) = \frac{1}{2}\rho_{\mathrm{a}} V_{\mathrm{w,a}}^2 \begin{bmatrix} -c_x\cos(\gamma_{\mathrm{w,a}})A_{\mathrm{t}} \\ -c_y\sin(\gamma_{\mathrm{w,a}})A_{\mathrm{l}} \\ c_n\sin(2\gamma_{\mathrm{w,a}})A_{\mathrm{l}}L_{\mathrm{oa}} \end{bmatrix}. \tag{9.12}$$

Similar to the previous section, (9.10) can be compactly written in the form (9.5), i.e.,

$$\dot{\boldsymbol{x}}_{\mathrm{ext}} = \boldsymbol{f}_{\mathrm{ext}}(\boldsymbol{x}_{\mathrm{ext}}) + B_{\tau,\mathrm{ext}}\boldsymbol{u} + G\boldsymbol{w}_{\mathrm{ext}}, \qquad \boldsymbol{w}_{\mathrm{ext}} \sim \mathcal{N}(\boldsymbol{0}, Q_{\mathrm{c,ext}}), \tag{9.13a}$$

$$\boldsymbol{y}_{\mathrm{ext}} = \boldsymbol{h}_{\mathrm{ext}}(\boldsymbol{x}_{\mathrm{ext}}) + \boldsymbol{v}_{\mathrm{ext}} \qquad \boldsymbol{v}_{\mathrm{ext}} \sim \mathcal{N}(\boldsymbol{0}, R_{\mathrm{c,ext}}) \tag{9.13b}$$

where $\boldsymbol{w}_{\mathrm{ext}} = [\boldsymbol{w}_\nu^\top,\ \boldsymbol{w}_{\mathrm{d}}^\top,\ \boldsymbol{w}_{\mathrm{w}}^\top]^\top$ is the process noise, $Q_{\mathrm{c,ext}} = \mathrm{diag}(Q_{\mathrm{c},\nu},\ Q_{\mathrm{c,d}},\ Q_{\mathrm{c,w}})$ is the process noise covariance matrix, $R_{\mathrm{c,ext}} = \mathrm{diag}(R_{\mathrm{c},\eta},\ R_{\mathrm{c,w}})$ is the measurement noise covariance matrix of the extended system, respectively. Moreover, $\boldsymbol{y}_{\mathrm{ext}} = [\boldsymbol{y}^\top,\ \boldsymbol{y}_{\mathrm{w}}^\top]^\top$ is the extended measurement vector, $\boldsymbol{h}_{\mathrm{ext}}(\boldsymbol{x}_{\mathrm{ext}}) = [\boldsymbol{h}^\top(\boldsymbol{x}_{\mathrm{ext}}),\ \boldsymbol{h}_{\mathrm{w}}^\top(\boldsymbol{x}_{\mathrm{ext}})]^\top$ is the extended measurement function, and

$$\boldsymbol{f}_{\mathrm{ext}}(\boldsymbol{x}_{\mathrm{ext}}) = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}) \\ \hdashline \boldsymbol{0}^{(5\times1)} \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{0}^{(3\times1)} \\ M^{-1}(\boldsymbol{\tau}_{\mathrm{w}}(\boldsymbol{x}, \boldsymbol{x}_{\mathrm{w}}) + R_{z,\psi}^\top\boldsymbol{d}) \\ \hdashline \boldsymbol{0}^{(5\times1)} \end{bmatrix}}_{\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}_{\mathrm{ext}})},\ B_{\tau,\mathrm{ext}} = \begin{bmatrix} B_\tau \\ \hdashline 0^{(5\times3)} \end{bmatrix}, \tag{9.14}$$

and

$$G = \begin{bmatrix} 0^{(3\times8)}, \\ \mathrm{diag}(G_\nu,\ G_{\mathrm{d}},\ G_{\mathrm{w}}) \end{bmatrix} \tag{9.15}$$

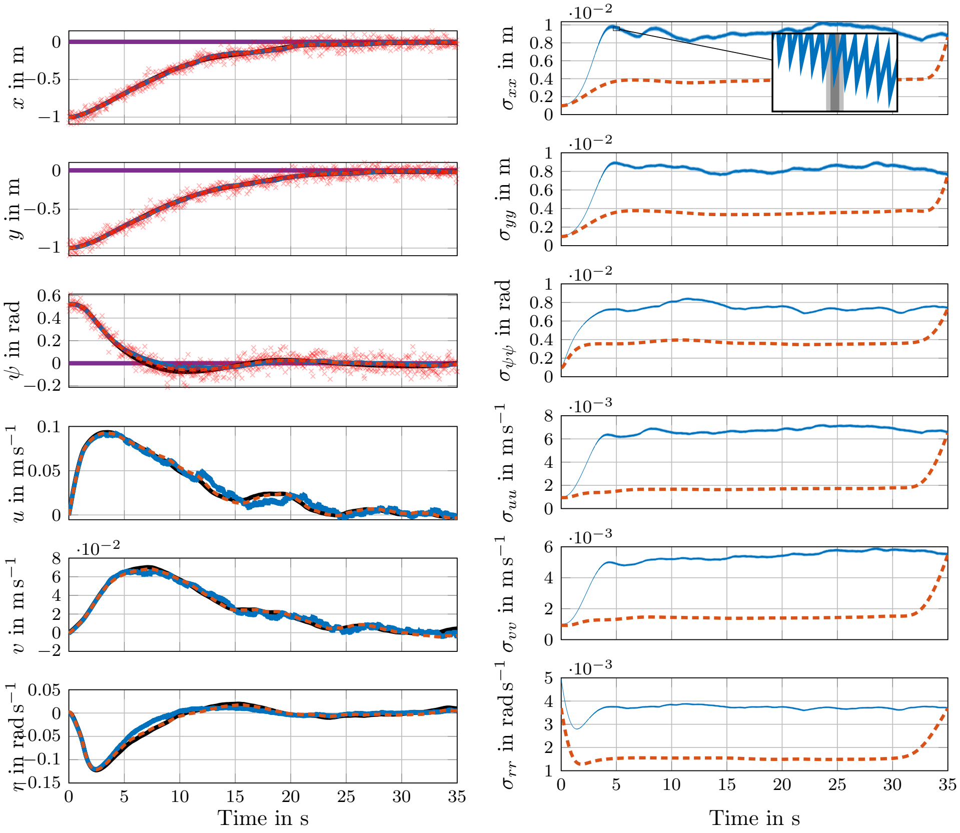
are the drift vector field with disturbance model $\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}_{\mathrm{ext}})$, the input matrix, and the process noise gain matrix of the extended system, respectively.

In principle, the observability of this system can be analyzed locally, i.e., based on the rank of the Jacobian of the observability map, i.e.,

$$\boldsymbol{q}(\boldsymbol{x}_{\mathrm{ext}}) = \left[\begin{array}{c} L^0_{\boldsymbol{f}_{\mathrm{ext}}} \boldsymbol{h}(\boldsymbol{x}_{\mathrm{ext}}) \\ L^1_{\boldsymbol{f}_{\mathrm{ext}}} \boldsymbol{h}(\boldsymbol{x}_{\mathrm{ext}}) \\ L^2_{\boldsymbol{f}_{\mathrm{ext}}} \boldsymbol{h}(\boldsymbol{x}_{\mathrm{ext}}) \\ \hdashline L^0_{\boldsymbol{f}_{\mathrm{ext}}} \boldsymbol{h}_{\mathrm{w}}(\boldsymbol{x}_{\mathrm{ext}}) \end{array}\right] = \left[\begin{array}{c} L^0_{\boldsymbol{f}_{\mathrm{ext}}} \boldsymbol{\eta}(t) \\ L^1_{\boldsymbol{f}_{\mathrm{ext}}} \boldsymbol{\eta}(t) \\ L^2_{\boldsymbol{f}_{\mathrm{ext}}} \boldsymbol{\eta}(t) \\ \hdashline L^0_{\boldsymbol{f}_{\mathrm{ext}}} V_{\mathrm{w,a}}(t) \\ L^0_{\boldsymbol{f}_{\mathrm{ext}}} \gamma_{\mathrm{w,a}}(t) \end{array}\right] \in \mathbb{R}^{11}, \; O(\boldsymbol{x}_{\mathrm{ext}}) = \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{\mathrm{ext}}} \in \mathbb{R}^{11\times 11} \tag{9.16}$$

where $\det O(\boldsymbol{x}_{\mathrm{ext}}) \neq 0, \quad \forall t \in \mathbb{R}$ is required. However, due to the highly nonlinear relationship between the relative and the absolute wind quantities in the measurement function $\boldsymbol{h}_{\mathrm{w}}$, this expression cannot be analyzed in constructive manner. A simpler analysis is given, e.g., using the structural observability, a necessary condition for local observability, of the system but is omitted in this case [8]. Instead, we assume that the system is observable, since, as a rule of thumb, for any additional estimated state (in this case $V_{\mathrm{w,t}}$, $\beta_{\mathrm{w,t}}$) an additional measurement (in this case $V_{\mathrm{w,a}}$, $\gamma_{\mathrm{w,a}}$) is needed REF, and the system's observability without wind measurements is proven in the previous section.

Analogously to the lumped disturbance model, the Jacobian matrices of the extended model are needed for the EKF iterations and are presented in Appendix A.3.



(a) Target pose $\boldsymbol{\eta}^*$ (violet), simulated state $\boldsymbol{x}$ (black), estimated state $\hat{\boldsymbol{x}}$ (blue), and smoothed state $\check{\boldsymbol{x}}$ (dashed orange) with noisy measurements $\boldsymbol{y}$ (red crosses).

(b) Square root of diagonal elements of state error covariance matrix P_x (blue) and corresponding smoothed values $\check{P}_x$ (dashed orange).

Figure 9.1.: Vessel states $\boldsymbol{x}$ (left) and corresponding estimation error standard deviation P_x (right) for the lumped estimation with disturbance compensation for the scenario in Tab. 9.2.

9.2. Results and Discussion

The following sections describe the performance of the dynamic positioning controller with control allocation using the lumped disturbance model and the combined lumped and wind disturbance model, respectively. The former combines all forces and model uncertainties into the lumped disturbance vector $\boldsymbol{d}$ and the latter distinguishes between disturbances caused by wind and all other disturbances and uncertainties. This combined approach also makes use of wind measurements and an underlying model for the wind force.

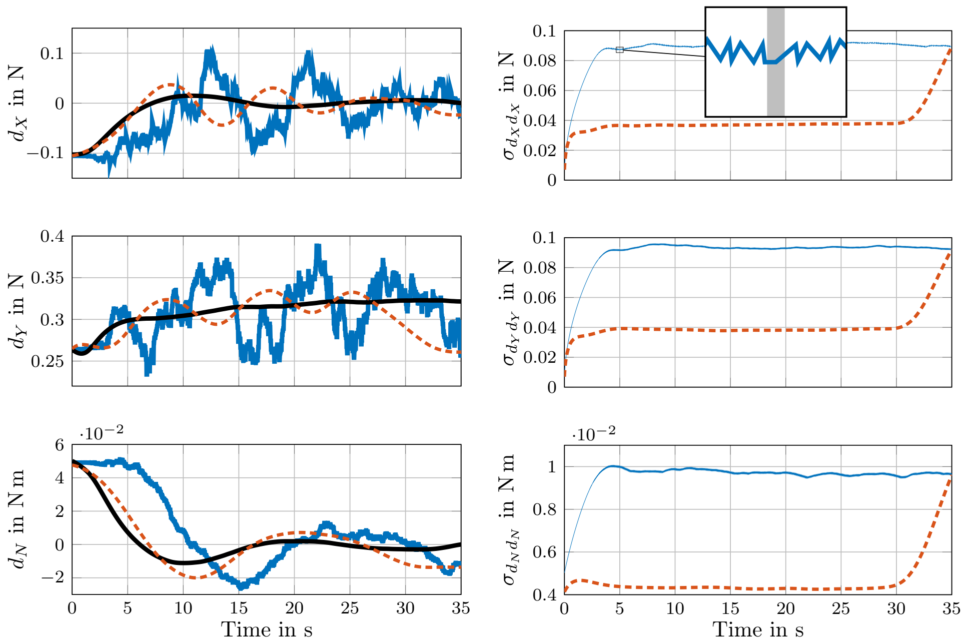
9.2.1. Lumped Disturbance Model

The primary control goal of stabilizing the origin of the pose, i.e., dynamic positioning is achieved by the proposed EKF-based controller. In Fig. 9.1a the vessel states are depicted along with the corresponding square root of the diagonal elements of the state error covariance matrix P_x in Fig. 9.1b. In the former figure the true (black) and estimated (blue) states are depicted with the desired pose (violet) and measurements (red). As a post-processing step, the RTS smoother is applied to verify its performance and the smoothed estimates are also shown (dashed orange). As can be seen, the primary goal of $\boldsymbol{\eta}^* = \mathbf{0}$ is achieved by the nonlinear controller after about 25 s. The vessel's heading converges slightly faster after about 10 s, which is due to the fact that the corresponding poles of the stabilizing controller (6.11) are chosen further to the left in the complex plane compared with the poles of the position controller, see Tab. 9.2. Note that this behavior is desired in this case, since the correct heading angle is deemed more important compared with the position of the vessel. However, this behavior can easily be changed by choosing different poles for the position and/or heading.

Furthermore, the state error standard deviation (and covariance) converges to nearly steady state, which corresponds to the steady state solution $P_{x,\infty}$ of the algebraic matrix Riccati equation [141, Sec. 7.3] calculated using the linearization of the dynamic model around the origin $\boldsymbol{\eta} = \mathbf{0}$. This also implies that the proposed adaptation process according to Sec. 8.4 of the measurement noise covariance matrix R_c has converged. Since the fundamental sample time of the EKF is different from the measurement sample time, ripples in the state error standard deviation are observed, which is magnified for σ_{xx} in Fig. 9.1b at $t = 5$ s. This is because, in general, the state error covariance increases in between measurements because the prediction error covariance P_k^- is propagated to the next time instant with, essentially, infinite measurement error covariance. More specifically, the matrix Riccati equation (in continuous time) is expressed mathematically using

$$\dot{P} = AP + PA^\top + GQG^\top - PCR^{-1}C^\top P, \qquad P(t_0) = P_0, \tag{9.17}$$

with A calculated based on (8.7). In between measurements, the last term vanishes such that this differential equation is driven by the positive semidefinite matrix $GQG^\top$ and, therefore,



(a) True disturbance $\boldsymbol{d}$ (black), estimated disturbance $\hat{\boldsymbol{d}}$ (blue), and smoothed disturbance $\breve{\boldsymbol{d}}$ (orange).

(b) Square root of diagonal elements of disturbance error covariance matrix P_d (blue) and corresponding smoothed values (orange).

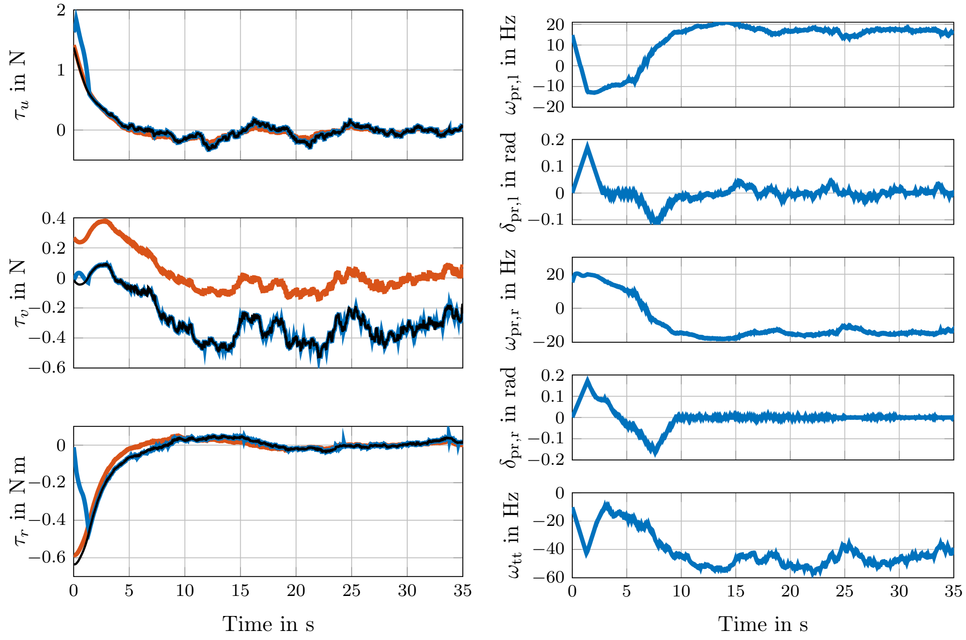
Figure 9.2.: Disturbances $\boldsymbol{d}$ acting on the vessel (left) and corresponding diagonal elements of the estimation error standard deviation $\sqrt{P_{\mathrm{d}}}$ (right) for the lumped estimation with disturbance compensation for the scenario in Tab. 9.2.

increases over time. As soon as a measurement is available, the state error covariance is reduced. Additionally, the RTS smoother is applied to the EKF estimates to further reduce the state error covariance as a post-processing step. Typical behavior is observed, i.e., it holds that $\breve{P} \leq P \; \forall t \in \mathbb{R}$ and $\breve{P}(T) = P(T)$ since there is no additional measurement available for the smoother to increase the estimate at the end of the simulation.

Apart from the vessel states, the lumped estimation model (9.4) is used to estimate the lumped disturbance expressed in the NED frame, i.e., $\boldsymbol{d} = R_{z,\psi}\boldsymbol{\tau}_{\mathrm{d}}$, which is shown in Fig. 9.2a, where the simulated (black), estimated (blue), and smoothed (dashed orange) values are depicted. In Fig. 9.2b the corresponding standard deviation is shown similarly as the vessel states error standard deviation. As expected, the disturbance estimate $\hat{\boldsymbol{d}}$ is less accurate compared with the vessel state estimates, which can be attributed to the erroneous disturbance model, i.e., $\dot{\boldsymbol{d}} = G_{\mathrm{d}}\boldsymbol{w}_{\mathrm{d}}$, which does not reflect the true dynamics of the simulated disturbance (black) in Fig. 9.2a. Other approaches [47, Sec.11.3.6],[35] employ a first order Markov model

$$\dot{\boldsymbol{d}} = -T^{-1}\boldsymbol{d} + G_{\mathrm{d}}\boldsymbol{w}_{\mathrm{d}}, \tag{9.18}$$

to reflect the time varying behavior, but this requires additional tuning of the added



(a) Nominal commanded controller input $\boldsymbol{\tau}_{\mathrm{fb}}$ (orange) calculated by the nonlinear controller, total commanded controller input $\boldsymbol{\tau}_{\mathrm{c}}^{*} = \boldsymbol{\tau}_{\mathrm{fb}} - R_{z,\hat{\psi}}^{\top}\hat{\boldsymbol{d}}$ (black), and realized input after control allocation $\boldsymbol{\tau}_{\mathrm{c}} = \boldsymbol{g}(\hat{\boldsymbol{x}}, \boldsymbol{u}^{*})$ (blue).

(b) Optimal control surfaces $\boldsymbol{u}^{*}$ (blue) as the solution of (6.24) of the Cybership II propulsion model.

Figure 9.3.: Control input $\boldsymbol{\tau}_{\mathrm{c}}$ (left) and control surfaces $\boldsymbol{u}$ (right) of for the scenario in Tab. 9.2.

DOF, namely, the time decay matrix T that includes the time constants of the assumed (exponential) decay. Changes in the surge and sway disturbance d_X, d_Y, respectively, are estimated with a sufficiently small time delay and accuracy compared with the simulated value. In the case of the yaw moment, a certain time delay is observed, which, in this case, is still sufficiently small to allow the disturbance compensation to take the desired effect. Note, however, that the delay disappears using the smoothed values of the disturbance estimate and the error covariance significantly improves. This also motivates the use of the RTS smoother in the context of parameter estimation, which essentially employs the same dynamical model as the disturbance model, i.e., no change over time but, in that case, the estimation model is accurate and enables the on- and offline parameter estimation [52].

In Fig. 9.3a the control inputs are depicted. Therein, the nonlinear controller input $\boldsymbol{\tau}_{\mathrm{fb}}$ is depicted (orange) along with the total desired input $\boldsymbol{\tau}_{\mathrm{c}}^{*} = \boldsymbol{\tau}_{\mathrm{fb}} - R_{z,\hat{\psi}}^{\top}\hat{\boldsymbol{d}}$ (black), i.e., the sum of the nonlinear controller input and the (negative) estimated disturbance, and the realized input after the control allocation, i.e., $\boldsymbol{\tau}_{\mathrm{c}} = \boldsymbol{g}(\hat{\boldsymbol{x}}, \boldsymbol{u}^{*})$, where, $\boldsymbol{g}(\cdot)$ represents the nonlinear propulsion model of the Cybership II, and $\boldsymbol{u}^{*}$ is set of optimal control inputs as the solution of (6.24), which are visualized in Fig. 9.3b. In that particular figure, it can

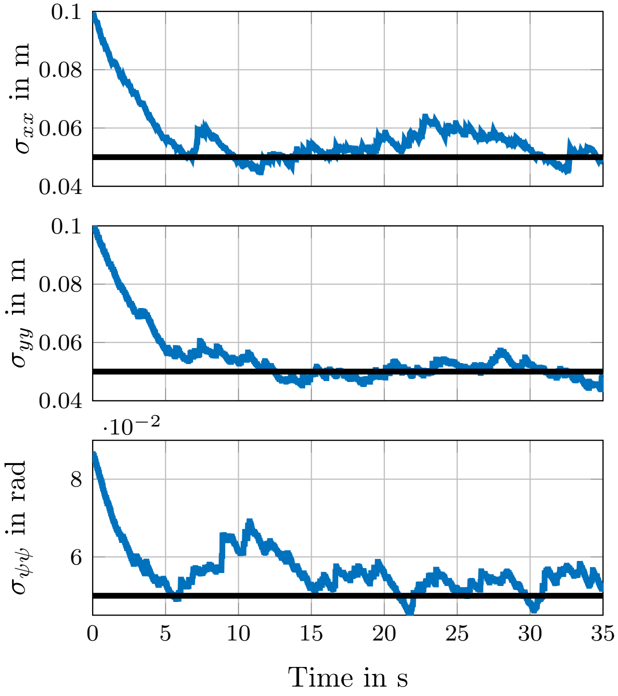
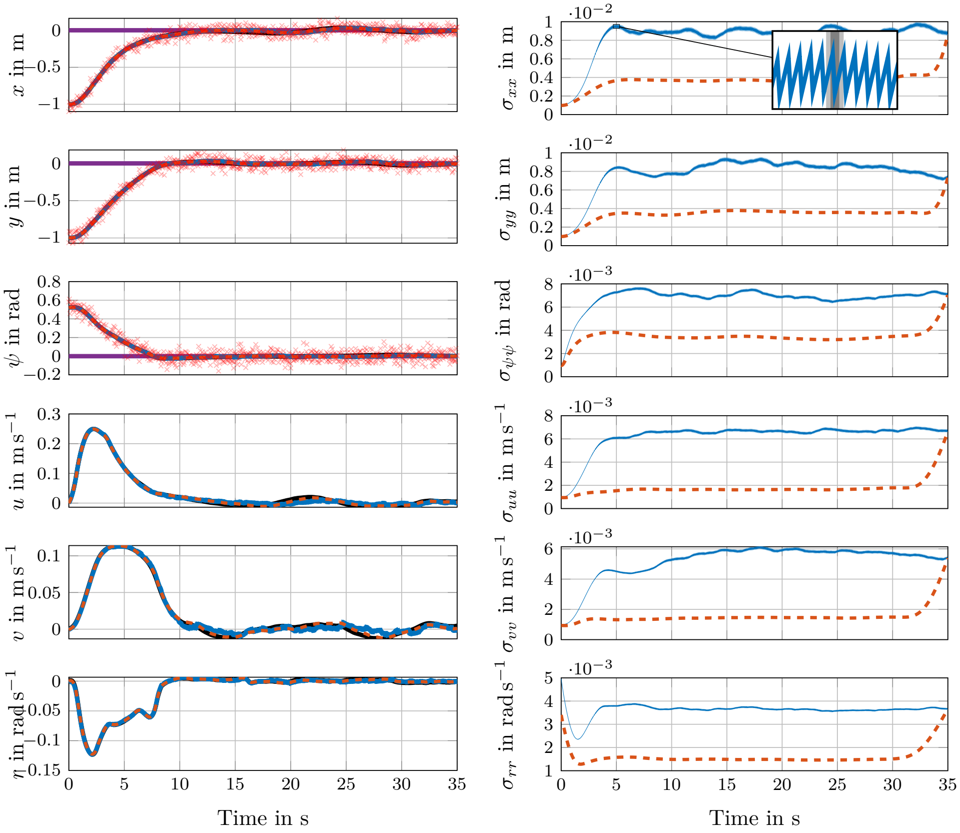


Figure 9.4.: Square root of diagonal elements of the pose measurement noise covariance matrix $R_{\mathrm{c},eta}$ that is adaptively adjusted using an EWMA for the lumped estimation with disturbance compensation for the scenario in Tab. 9.2.

be seen that both rudder angles $\delta_{\mathrm{pr,l}}$, $\delta_{\mathrm{pr,r}}$ are kept at zero if possible, which validates the weighting of the control surface weighting matrix W_u in Tab. 9.2. Moreover, the total commanded input in Fig. 9.3a shows a significant deviation from the commanded virtual input calculated by the nonlinear controller in the sway force, which is due to the true wind angle of attack $\beta_{\mathrm{w,t}} = -\pi/2$, i.e., the wind is coming from the west, pressing the vessel to its starboard side in the desired pose $\boldsymbol{\eta}^* = \mathbf{0}$. This force is counteracted by the disturbance compensation and, in view of the control surfaces in Fig. 9.3b, realized mainly by means of the bow thruster, which generates a negative sway force for $\omega_{\mathrm{tt}} < 0$ and also a negative yaw moment. This yaw moment is counteracted by the left and right main propellers of the propulsion, where the left propeller is used to generate a forward thrust and the right propeller is used to generate a (similar) backward thrust, which, in combination with (approximately) zero rudder angles, results in a positive yaw moment induced by the main propellers. This behavior can be changed, e.g., by assigning different reference values $\boldsymbol{u}_{\mathrm{ref}}$ and corresponding weights to the individual control surfaces in the control surface weighting matrix W_u in Tab. 9.2 as now, the desired rudder angles are zero. In principle, a sway force can be induced by the main propellers with nonzero rudders, but this is not desired in this particular example. In that case, the resulting yaw moment would then be counteracted by the bow thruster.

In Fig. 9.4 the square root of the diagonal elements of the measurement noise covariance matrix R_c are depicted, which is calculated using the EWMA filter from Sec. 8.4 and, more specifically, using (8.26). Therein, the true measurement noise standard deviation is depicted (black) with the estimated value (blue), see also Tab. 9.2. The estimated value is initialized with a relatively large error and converges to the true respective value in approximately 5 s. This is also about the time when the EKF state and disturbance error covariance matrices converge to nearly steady state.



(a) Target pose $\boldsymbol{\eta}^*$ (violet), simulated state $\boldsymbol{x}$ (black), estimated state $\hat{\boldsymbol{x}}$ (blue), and smoothed state $\check{\boldsymbol{x}}$ (dashed orange) with noisy measurements $\boldsymbol{y}$ (red crosses).

(b) Square root of diagonal elements of state error covariance matrix P_x (blue) and corresponding smoothed values $\check{P}_x$ (dashed orange).

Figure 9.5.: Vessel states (left) and corresponding estimation error standard deviation (right) for the combined lumped and wind estimation with disturbance compensation for the scenario in Tab. 9.3.

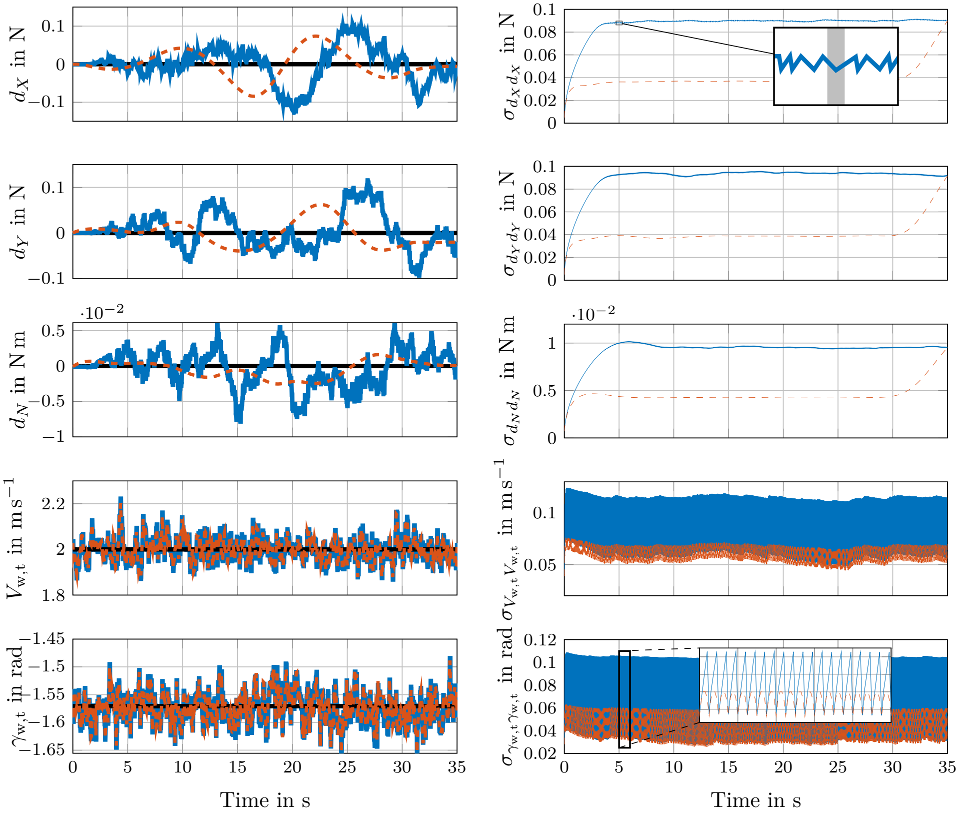
9.2.2. Combined Lumped and Wind Disturbance Model

The combined estimation model (9.10) shows similar qualitative behavior as the lumped estimation model (9.4). In particular, the main controller goal to stabilize the vessel at the target pose is achieved, see Fig. 9.5a. The main difference lies in the time in that the pose converges to the target value, which, in this case, takes about 10 s compared to 25 s for the lumped estimation model, even though both controllers operate with the same poles, see Tab. 9.2 and Tab. 9.3. The state error standard deviation is depicted in Fig. 9.5b also converge to steady state values and reach almost identical values compared with the lumped estimation model.

The faster convergence to the desired pose compared with the lumped estimation approach can be attributed the fact that the wind model (5.56) is included in the combined approach, which enables the observer to estimate the true wind speed and direction accurately, see Fig. 9.6a, which can be translated into the approximate wind force using the wind model to counteract the actual wind disturbance. The additional estimates $\hat{\boldsymbol{x}}_{\mathrm{w}}$ of the true wind speed and angle are very accurate and the corresponding error standard deviation approximately approaches the simulated measurement noise added to the apparent wind values, see Tab. 9.3. Note that the error standard deviation of the true wind speed and angle oscillate rapidly such that the plot appears to be a wide band. For this reason, the segment where $t \in [5\,\mathrm{s}, 6\,\mathrm{s}]$ is magnified in Fig. 9.6b for $\sigma_{\gamma_{\mathrm{w,t}}\gamma_{\mathrm{w,t}}}$. The lumped disturbance estimate $\hat{\boldsymbol{d}}$ in Fig. 9.6a oscillates around the true additional disturbance, which, in this case, is zero.

The control input $\boldsymbol{\tau}_{\mathrm{c}}$ and the control surfaces $\boldsymbol{u}$ are depicted in Fig. 9.7. Therein, the total commanded input is much more aggressive in magnitude compared with the lumped approach, which can be attributed to the inclusion of the wind model to the disturbance estimation. This aggressive behavior also explains the faster convergence to the desired pose. The control surfaces also show similar qualitative behavior compared with the lumped estimation approach. However, in the first 10 s the control allocation only partly achieves the commanded sway force and yaw moment but, afterwards, converges to the commanded inputs. The more aggressive behavior is also reflected in the use of the control surfaces, which is particularly visible in the bow thruster rotational speed ω_{tt} and the larger magnitudes of the rudders.

In Fig. 9.8 the square root of the diagonal elements of the measurement noise covariance matrices $R_{\mathrm{c},\eta}$ and $R_{\mathrm{c,w}}$ are depicted, which are calculated using the EWMA filter from Sec. 8.4 and, more specifically, using (8.26). Therein, the true measurement noise standard deviation is depicted (black) with the estimated value (blue), see also Tab. 9.3. The estimated values are initialized with a relatively large error and converges to the true respective value in approximately 5 s. While the pose measurement standard deviation approach the true values, the wind measurement standard deviation is slightly underestimated in both the speed and the direction measurement, which is attributed to the highly nonlinear relationship between the measured apparent and the true wind speed and direction. This nonlinearity might be poorly approximated using the Jacobian matrix that is used by the EKF.

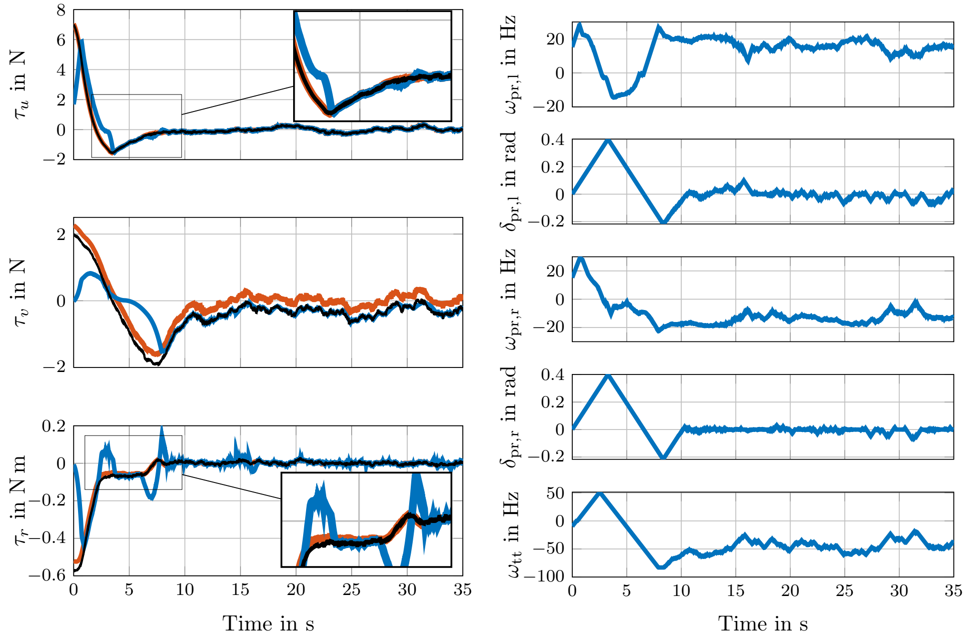


(a) True disturbance $\boldsymbol{d}$ and wind $\boldsymbol{x}_{\mathrm{w}}$ (black), estimated disturbance $\hat{\boldsymbol{d}}$ and wind $\hat{\boldsymbol{x}}_{\mathrm{w}}$ (blue), and smoothed disturbance $\check{\boldsymbol{d}}$ and wind $\check{\boldsymbol{x}}_{\mathrm{w}}$ (orange).

(b) Square root of diagonal elements of disturbance error covariance matrix P_d and wind error covariance matrix P_{w} (blue), respectively, and corresponding respective smoothed values $\check{P}_{\mathrm{d}}$, $\check{P}_{\mathrm{w}}$ (orange).

Figure 9.6.: Disturbances $\boldsymbol{d}$ and wind[a] $\boldsymbol{x}_{\mathrm{w}}$ acting on the vessel (left) and corresponding respective square root of diagonal elements of the estimation error covariance matrices P_{d}, P_{w} (right) for the combined lumped and wind estimation with disturbance compensation for the scenario in Tab. 9.3.

[a] Note that since the apparent wind is measured and not the true wind the measurements of the wind are not depicted in this plot to avoid confusion.



(a) Nonlinear control input $\boldsymbol{\tau}_{\text{fb}}$ (orange), total control input $\boldsymbol{\tau}_{\text{c}}^* = \boldsymbol{\tau}_{\text{fb}} - \boldsymbol{\tau}_{\text{w}}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}_{\text{w}}) - R_{z,\hat{\psi}}^\top \hat{\boldsymbol{d}}$ (black), and realized input after control allocation $\boldsymbol{\tau}_{\text{c}} = \boldsymbol{g}(\hat{\boldsymbol{x}}, \boldsymbol{u}^*)$ (blue).

(b) Optimal control surfaces $\boldsymbol{u}^*$ (blue) as the solution of (6.24) of the Cybership II propulsion model.

Figure 9.7.: Control input $\boldsymbol{\tau}_{\text{c}}$ (left) and control surfaces $\boldsymbol{u}$ (right) of for the combined lumped and wind estimation with disturbance compensation for the scenario in Tab. 9.3.

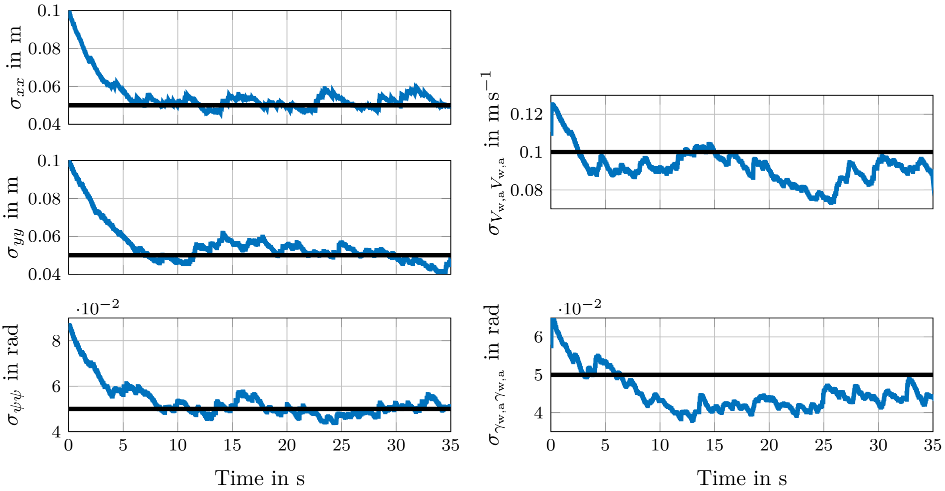


Figure 9.8.: Square root of diagonal elements of the pose and wind measurement noise covariance matrix $R_{\text{c},\eta}$ (left), $R_{\text{c,w}}$ (right), respectively, that is adaptively adjusted using an EWMA for the combined lumped and wind estimation with disturbance compensation the scenario in Tab. 9.3.

10. Predictive Path Following

In this chapter a path-following problem for a mariner class vessel [29] is solved in a confined environment in the Kiel Canal, one of the most frequented canals in the world, where AIS data is used to obtain obstacle information. To this end, a nonlinear MPC controller according to Sec. 6.3 is proposed, where the reference path is given by piecewise linear segments. In this simulative study, the timing law is chosen such that realistic behavior of the vessel is realized by taking into account the vessel's service (surge) speed u_0 and its distance to current line segment, thereby achieving slow progress on the path if the vessel is far away from the current line segment and fast progress if the vessel is close to the current line segment [123]. Furthermore, the MPC takes into account obstacles using the dual collision avoidance constraints as presented in Sec. 7 in combination with frustum-, occlusion-, and backface culling, as well as input bound and rate constraints. An EKF according to Chapter 8 is used to reconstruct the vessel's state based on pose measurements. The presented approach is illustrated by means of a simulative study that realizes the path following in a confined environment in the Kiel Canal. Obstacle data is obtained from real AIS data.

10.1. Problem Formulation

In the following section the individual segments of the control loop as depicted in Fig. 2.1 are discussed for the predictive path following task. To this end, the underlying vessel model and its parameterization is introduced. Furthermore, the path following MPC controller is presented, which gives a brief overview of the theory and concepts of predictive path following. Additionally, dual collision avoidance constraints are used to avoid obstacles in the environment and the particular. Lastly, the EKF observer model is presented where we assume that noisy measurements of the vessel's pose are available. From these measurements, the vessel state is reconstructed. In this example, no disturbance estimation and/or compensation is used, but wind is added as a disturbance based on a Weibull distributed wind speed and a normally distributed wind direction. All relevant parameters regarding the scenario, controller, and observer are summarized in Tab. 10.2.

	Symbol	Value	Symbol	Value	Symbol	Value	Symbol	Value
General	L_{oa}	160.93	m	798e−5				
	B	23.17	J_{zz}	39.2e−5				
	x_g	−0.023						
Added Mass	$X_{\dot{u}}$	−42e−5						
	$Y_{\dot{v}}$	−748e−5	$Y_{\dot{r}}$	−9.34e−5				
	$N_{\dot{v}}$	4.65e−5	$N_{\dot{r}}$	−43.8e−5				
Damping	X_u	−184e−5	X_{uu}	−110e−5	$X_{v\delta}$	93e−5	X_{uuu}	−215e−5
			X_{vv}	−899e−5	X_{rr}	18e−5	$X_{uv\delta}$	93e−5
			$X_{\delta\delta}$	−95e−5	X_{rv}	798e−5	$X_{u\delta\delta}$	−190e−5
	Y_0	−4e−5	Y_{0u}	−8e−5			Y_{0uu}	−4e−5
	Y_v	−1160e−5	Y_{vu}	−1160e−5	Y_{vvr}	15 356e−5	Y_{vvv}	−8078e−5
	Y_r	−499e−5	Y_{ru}	−499e−5	$Y_{uu\delta}$	278e−5	$Y_{v\delta\delta}$	−4e5
	Y_δ	278e−5	$Y_{u\delta}$	556e−5	$Y_{\delta\delta\delta}$	−90e−5	$Y_{vv\delta}$	1190e−5
	N_0	3e−5	N_{0u}	6e−5			N_{0uu}	3e−5
	N_v	−264e−5	N_{vu}	−264e−5	N_{vvr}	−5483e−5	N_{vvv}	1636e−5
	N_r	−166e−5	N_{ru}	−166e−5	$N_{uu\delta}$	−139e−5	$N_{v\delta\delta}$	13e−5
	N_δ	−139e−5	$N_{u\delta}$	−278e−5	$N_{\delta\delta\delta}$	45e−5	$N_{vv\delta}$	−489e−5

Table 10.1.: Parameters of the Mariner Class MSV in dimensionless units according to [29]. Note that there's only one rudder to steer the ship.

10.1.1. Vessel Model

A mariner class model is used in this simulative study, which constitutes a 3DOF model as described in (5.72) with the exception that the velocity-dependent forces acting on the ship are fitted to a third-order Taylor polynomial, which consists of numerous coefficients given in Tab. 10.1. In total, the mathematical model is expressed using

$$\dot{\boldsymbol{\eta}} = R_{z,\psi}([u_0,\ 0,\ 0]^\top + \tilde{\boldsymbol{\nu}}), \qquad t > t_0, \qquad \boldsymbol{\eta}(t_0) = \boldsymbol{\eta}_0 \tag{10.1a}$$

$$\dot{\boldsymbol{\nu}} = M^{-1}\boldsymbol{\tau}(\tilde{\boldsymbol{\nu}}, \delta), \qquad t > t_0, \qquad \boldsymbol{\nu}(t_0) = \boldsymbol{\nu}_0, \tag{10.1b}$$

where $\tilde{\boldsymbol{\nu}} = [\Delta u,\ v,\ r]^\top$ and $\Delta u = u - u_0$ is the constant surge speed u_0. This model is, in essence, an autopilot model with only one control surface, namely, the rudder angle $u = \delta$ that is subject to the constraints

$$\mathcal{U} = \{\delta \in \mathbb{R} | -30^\circ \leq \delta \leq 30^\circ,\ |\dot{\delta}| \leq 5\,^\circ\,\mathrm{s}^{-1}\}. \tag{10.2}$$

In the simulation, a disturbance is acting on the vessel that is calculated based on (5.56) with true wind velocity $V_{\text{w,t}} \sim \mathcal{W}(2,1)$ and true wind angle of attack $\beta_{\text{w,t}} \sim \mathcal{N}(\pi, 10\frac{\pi}{180})$. Noise is added to the pose measurement such that $\boldsymbol{v}_\eta \sim \mathcal{N}(\mathbf{0}, R_{\text{c},\eta})$ with $R_{\text{c},\eta} = \text{diag}(3.87^2,\ 3.87^2,\ 0.32^2)$.

10.1.2. Model Predictive Controller

Path following is distinguished from trajectory tracking in that no time information is given a priori for the former, whereas the latter includes this information of when the system should be where in the state space in the target trajectory. In a sense, path following introduces another DOF to the control problem, namely, the path parameter θ, whose time evolution is given by a user-defined ODE of the form

$$\dot{\theta} = q(\boldsymbol{x}, \theta, v), \qquad t > t_0, \quad \theta(t_0) = \theta_0 \tag{10.3}$$

with the timing law $q : \mathbb{R}^{n_x} \times \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ that dictates the time evolution of the path parameter and, indirectly, of the predicted path points [39]. Output path following problems aim to achieve

$$\lim_{t \to \infty} \boldsymbol{h}(\boldsymbol{x}(t)) - \boldsymbol{p}(\theta) = \boldsymbol{0}, \tag{10.4}$$

where $\mathcal{P} = \{\boldsymbol{y} \in \mathbb{R}^{n_y} | \theta \in [0, \theta^+] \mapsto \boldsymbol{p}(\theta)\}$ is the path defined in the output space, and $\boldsymbol{p}(\theta)$ is the path parameterization. Path following problems can be tackled using nonlinear control theory, where, e.g., a transverse normal form, i.e., a variant of the Brunovsky canonical form introduced in Chapter 6.1, is used to stabilize the tangential and transverse error dynamics, respectively [10, 114, 115, 116]. Recently, predictive path following controllers have been developed that combine the concepts of path following with the benefits of optimization-based control approaches to take into account state and input constraints [40, 39]. This combined approach can also be further improved with differentially flat systems to transform the path into corresponding desired states [43, 17]. In the context of predictive path-following control, it is common to formulate the timing law (10.3) as an integrator chain $\theta^{(r_\theta)} = v$ of length r_θ with virtual input v that is treated as a design parameter and added as an additional decision variable to the MPC problem [17]. This particular choice also simplifies the stability analysis of the closed-loop system and allows for an intuitive connection to the transverse normal form [42] that is especially useful for controller synthesis.

In this example, a nonlinear MPC control problem (6.74) is used to solve this task with running cost

$$l(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{s}) = ||\boldsymbol{e}||_2^2 + ||\boldsymbol{s}||_S^2, \tag{10.5}$$

where

$$\boldsymbol{e} = \boldsymbol{p}^{\mathcal{V}} - \boldsymbol{p}(\theta) \tag{10.6}$$

is the error between the current position of the vessel and the reference path

$$\boldsymbol{p}(\theta) = \boldsymbol{p}_i^{\mathrm{wp}} + \alpha(\theta) \left(\boldsymbol{p}_{i+1}^{\mathrm{wp}} - \boldsymbol{p}_i^{\mathrm{wp}}\right) \tag{10.7}$$

	Name	**Symbol**	**Value**
Simulation	Model	–	Mariner Class, see Tab.10.1
	Initial state	$\hat{\boldsymbol{\eta}}_0$	$[54.36658°\text{N}, 10.13914°\text{E}, -75\frac{\pi}{180}]^\top$
		$\hat{\boldsymbol{\nu}}_0$	$[0, 0, 0]^\top$
	Final time	T	400
	True wind speed	$V_{\mathrm{w,t}}$	$\mathcal{W}(2, 1)$
	True wind angle of attack	$\beta_{\mathrm{w,t}}$	$\mathcal{N}(\pi, 10\frac{\pi}{180})$
	Pose measurement noise standard deviation	$\sqrt{R_{\mathrm{c},\boldsymbol{\eta}}}$	diag(3.87, 3.87, 0.32)
Controller	Control law	–	
	Prediction Horizon	t_{hor}	32
	Discretization Steps	N	13
	Solver	–	`SNOPT`
	Discrete propagation function	$\boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k)$	Trapezoidal
	Slack Weighting Matrix	S	100diag(8, 8, 8)
	Waypoints	$\boldsymbol{p}_1^{\mathrm{wp}}$	[54.36631°N, 10.14026°E]
		$\boldsymbol{p}_2^{\mathrm{wp}}$	[54.36677°N, 10.13753°E]
		$\boldsymbol{p}_3^{\mathrm{wp}}$	[54.36825°N, 10.13468°E]
		$\boldsymbol{p}_4^{\mathrm{wp}}$	[54.36847°N, 10.12809°E]
		$\boldsymbol{p}_5^{\mathrm{wp}}$	[54.36870°N, 10.12417°E]
	Tuning Parameter	σ	1
Obstacle Constraints	Dual constraints	–	(7.56)
	Safety Distance	d_{safe}	3
	Occlusion Culling Solid Angle Threshold	Ω	15
	View Frustum Angle	–	$45\frac{\pi}{180}$
	View Frustum Far Plane Offset	–	200
	View Frustum Side Plane Offset	–	50
Estimator	Model	–	(10.11)
	Discretization time	t_{ekf}	5e−2
	Discrete propagation function	$\boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k)$	Euler forward
	Measurement sample time	Δt_y	Δt_{ekf}
	Initial state error standard deviation	$\sqrt{P_0}$	10 diag(1, 1, 1, 1, 1, 1)
	Initial state estimate	$\hat{\boldsymbol{x}}_{-1}^{+}$	$\mathcal{N}(\hat{\boldsymbol{x}}_0, P_0)$
	Process noise gain	G	diag(1, 1, 1, 1, 1, 1)
	Process noise standard deviation	$\sqrt{Q_c}$	1e−2diag(1, 1, 1, 1, 1, 1)
	Pose measurement noise standard deviation	$\sqrt{R_{\mathrm{c},\boldsymbol{\eta}}}$	diag(3.87, 3.87, 0.32)
	Adaptation method	–	none
	Adaptation time window	–	n.A.
	Number of measurement update iterations	N_{iekf}	1

Table 10.2.: Scenario parameters for the path following scenario the Mariner model with a model predictive controller and an EKF. All units are SI. All covariance matrices are given in their square root form, i.e., they are represented using their respective standard deviations and, thus, preserve the unit of the underlying quantity.

at the i-th iteration, i.e., a piece-wise linear path between the waypoints $\boldsymbol{p}_i^{\mathrm{wp}}$ and $\boldsymbol{p}_{i+1}^{\mathrm{wp}}$, and

$$\alpha(\theta) = \frac{\theta - \sum_{j=1}^{i-1} ||\boldsymbol{p}_{j+1}^{\mathrm{wp}} - \boldsymbol{p}_j^{\mathrm{wp}}||_2}{||\boldsymbol{p}_{i+1}^{\mathrm{wp}} - \boldsymbol{p}_i^{\mathrm{wp}}||_2} \tag{10.8}$$

achieves that $\theta \in [0,\ \sum_i ||\boldsymbol{p}_{i+1}^{\mathrm{wp}} - \boldsymbol{p}_i^{\mathrm{wp}}||_2]$ is parameterized as the arc length of the entire reference path. Based on this, the time evolution of the path is dictated using the timing law based on [123]

$$\dot{\theta} = u_0 \left[1 - \sigma \tanh\left(\frac{\boldsymbol{e}^\top \boldsymbol{e}}{L_{\mathrm{oa}}^2}\right)\right], \qquad t > 0, \quad \theta(t_i) = \theta_i, \tag{10.9}$$

where L_{oa} is the length of the vessel, see Tab. 10.1, and $\sigma \in [0,\ 1]$ is a tuning parameter. The initial state θ_i for each iteration is initialized using the cross-track error calculated based on $\hat{\boldsymbol{x}}_i$. Note that the timing law (10.9) is a nonlinear function of the state $\boldsymbol{x}$ and the path parameter θ that is added to the MPC problem as an additional constraint. Furthermore, we point out that the path-following error (10.6) is only defined using the respective positions, whereas it is more common to define it in the entire state or output space. However, the proposed combination of the timing law and the running cost yields a good performance and also avoids extending the vector of decision variables with the virtual input v and the additional states representing the time derivatives up to order r_θ of the path parameter θ.

Collision avoidance is realized using the dual approach as presented in Chapter 7 for all different collision avoidance constraint methods in the full body case. For a concise overview of the different constraints, recall Tab. 7.1. To this end, the simulation is repeated using the proposed scenario for each of the different collision avoidance constraints, i.e., indicator, distance, and signed distance according to [161] and proposed signed distance constraints. For the indicator constraint case, the obstacles are artificially enlarged prior to adding them to the OCP to take into account the safety distance since, in that case, the safety distance can not be included directly. The safety distance is set to $d_{\mathrm{safe}} = 3\,\mathrm{m}$, the ship length is $L_{\mathrm{oa}} = 169.93\,\mathrm{m}$, and the MPC problem is solved on a receding horizon with $N = 13$ steps with a horizon length of $t_{\mathrm{hor}} = 32\,\mathrm{s}$. The slack variable s is used to ensure feasibility of the MPC problem and is penalized in the cost function (10.5) with the weighting matrix $S = \mathrm{diag}(800,\ 800,\ 800)$, see Maciejowski and Kerrigan [102] for more information on soft constrained MPC.

The dynamical equations (6.74c) and the integral in (6.74a) are discretized in time using the trapezoidal scheme, i.e., the discrete propagation function (6.71) is given by

$$\boldsymbol{F}(\boldsymbol{x}_{\mathrm{ext},k}, \boldsymbol{x}_{\mathrm{ext},k+1}, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}) = \boldsymbol{x}_k + \frac{\Delta t_{\mathrm{mpc}}}{2}\left(\boldsymbol{f}_{\mathrm{ext}}(\boldsymbol{x}_{\mathrm{ext},k}, \boldsymbol{u}_k) + \boldsymbol{f}_{\mathrm{ext}}(\boldsymbol{x}_{\mathrm{ext},k+1}, \boldsymbol{u}_{k+1})\right), \tag{10.10}$$

where $\boldsymbol{x}_{\mathrm{ext}} = [\boldsymbol{x}^\top,\ \theta]^\top$ is the augmented state vector, $\boldsymbol{f}_{\mathrm{ext}} = [\boldsymbol{f}^\top,\ q]^\top$ is the augmented right-hand-side. The input is interpolated as a piece-wise linear continuous function.

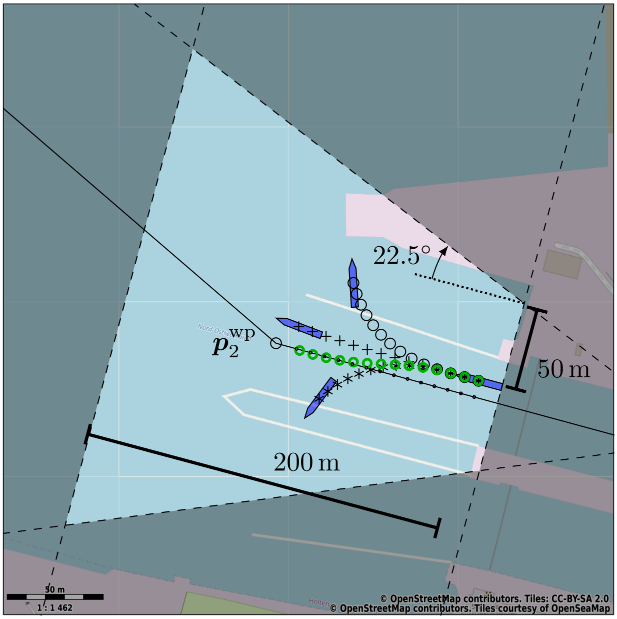

22.5°
$\boldsymbol{p}_2^{\mathrm{wp}}$
50 m
200 m
50 m
1 : 1 462
© OpenStreetMap contributors. Tiles: CC-BY-SA 2.0
© OpenStreetMap contributors. Tiles courtesy of OpenSeaMap

Figure 10.1.: Illustration of the possible predicted vessel states calculated using (7.57) (black crosses, circles, and asterisks, respectively) for the first iteration with view frustum, optimal path points (small black dots), and optimal vessel positions (green circles) for the first MPC iteration.

This results in a total of $N(n_x + n_\theta)$ defect constraints, where n_θ is the dimension of the path parameter θ and $n_\theta = 1$ in this case. Furthermore, the problem consists of $n_{w_x} = (N+1)(n_x + n_\theta)$ state decision variables, $n_{w_u} = (N+1)n_u$ input decision variables. Since all other constraints are evaluated at the discrete time steps, $4(N+1)$ (linear) input inequality constraints resulting from the feasible set (10.2), and $N+1$ (nonlinear) collision avoidance (state inequality) constraints, $2(N+1)K$ consistency (state equality) constraints, and $(N+1)(L+\check{L})K$ non-negativity (linear inequality) constraints resulting from the collision avoidance constraints are added to the problem, where K is the number of obstacles considered in the current iteration.

To improve efficiency, the culling procedure according to Sec. 7.5 is used to reduce the number of collision avoidance constraints and dual decision variables that are considered in the MPC problem. To this end, the view frustum far clipping plane is set at a distance of 200 m from the body coordinate origin of the controlled vessel. The corresponding view angle of the view frustum is set to 45° with the left and right clipping planes offset by 50 m, see Fig. 10.1. Occluders are chosen based on a solid angle threshold of $\Omega_{\min} = 15\,\mathrm{sr}$. Furthermore, the predicted vessel's positions needed to evaluate the culling conditions for every predicted time step are estimated using (7.57), which is depicted in Fig. 10.1 for the first iteration along with the optimal predicted vessel positions (green circles), i.e., the solution of the MPC problem.

10.1.3. State Estimation

A standard EKF as discussed in Sec. 8 is used to estimate the state of the controlled vessel based on noisy pose measurements. In this context a standard EKF signifies an estimator without adaptive covariance estimation and without iterative measurement update. The EKF operates with a sampling time of $\Delta t_{\mathrm{ekf}} =$ and measurement sample time $\Delta t_{\mathrm{meas}} = \Delta t_{\mathrm{ekf}}$ and applies the Euler forward discrete propagation function. The observer model is expressed mathematically as

$$\begin{aligned}
\dot{\boldsymbol{\eta}} &= R_{z,\psi}([u_0,\ 0,\ 0]^\top + \tilde{\boldsymbol{\nu}}) + G_\eta \boldsymbol{w}_\eta, & \boldsymbol{w}_\eta &\sim \mathcal{N}(\mathbf{0}, Q_{\mathrm{c},\eta}), && (10.11\mathrm{a})\\
\dot{\boldsymbol{\nu}} &= M^{-1}\boldsymbol{\tau}(\tilde{\boldsymbol{\nu}}, \delta) + G_\nu \boldsymbol{w}_\nu, & \boldsymbol{w}_\nu &\sim \mathcal{N}(\mathbf{0}, Q_{\mathrm{c},\nu}), && (10.11\mathrm{b})\\
\boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v}_\eta = \boldsymbol{\eta} + \boldsymbol{v}_\eta, & \boldsymbol{v}_\eta &\sim \mathcal{N}(\mathbf{0}, R_{\mathrm{c},\eta}). && (10.11\mathrm{c})
\end{aligned}$$

which can be depicted more compactly using

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, u) + G\boldsymbol{w}, & \boldsymbol{w} &\sim \mathcal{N}(\mathbf{0}, Q_{\mathrm{c}}), && (10.12\mathrm{a})\\
\boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v}_\eta, & \boldsymbol{v}_\eta &\sim \mathcal{N}(\mathbf{0}, R_{\mathrm{c},\eta}). && (10.12\mathrm{b})
\end{aligned}$$

Therein, $\boldsymbol{x} = [x,\ y,\ \psi,\ u,\ v,\ r]^\top$ is the observer state, $\boldsymbol{w}$ is the process noise, $Q_{\mathrm{c}} = \mathrm{diag}(Q_{\mathrm{c},\eta}, Q_{\mathrm{c},\nu})$ is the process noise covariance matrix, $G = \mathrm{diag}(G_\eta,\ G_\nu)$ is the process noise gain matrix. The latter is chosen to be a unit matrix, which corresponds to the assumption that every noise component affects the system equally. Furthermore, $\boldsymbol{v}_\eta$ in (10.12) is the measurement noise and $R_{\mathrm{c},\eta}$ is the corresponding measurement noise covariance matrix. The initial state estimates differ significantly from the true state to illustrate the convergence of the EKF. The observer pose measurement noise covariance matrix $R_{\mathrm{c},\eta}$ is equal to the true noise covariance, which makes the adaptive noise covariance estimation obsolete. Furthermore, no iterative measurement update is used in the EKF correction step.

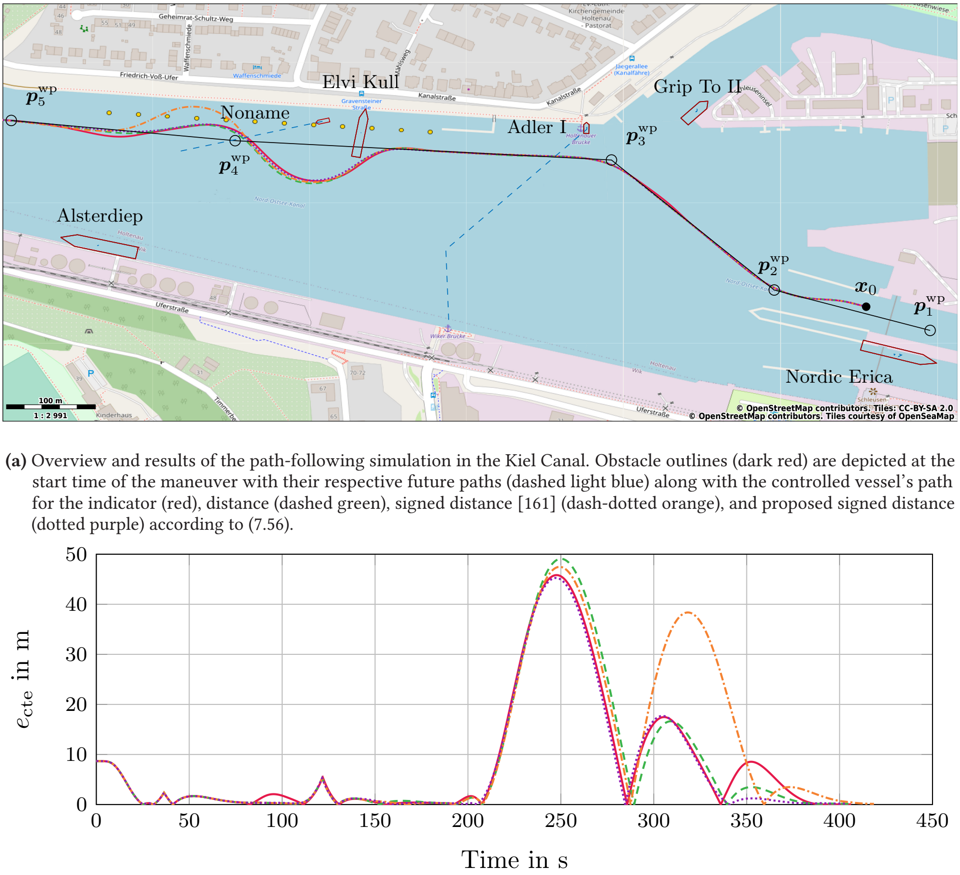
10.2. Results and Discussion

The predictive controller realizes the path following with the controlled vessel with a varying RMS cross track error $e_{\mathrm{rms}} = \sqrt{\frac{1}{N}\sum_i^N e_{\mathrm{cte}}^2} = \{14.46\,\mathrm{m},\ 15.47\,\mathrm{m},\ 17.78\,\mathrm{m},\ 18.04\,\mathrm{m}\}$ for the indicator, distance, signed distance according to [161], and proposed signed distance constraints, respectively, see Fig. 10.2. Therein, the cross track error is defined as

$$e_{\mathrm{cte}} = \cos(\psi_i^{\mathrm{wp}})(y(t) - y_i^{\mathrm{wp}}) - \sin(\psi_i^{\mathrm{wp}})(x(t) - x_i^{\mathrm{wp}}), \tag{10.13}$$

where

$$\psi_i^{\mathrm{wp}} = \mathrm{atan2}(y_{i+1}^{\mathrm{wp}} - y_i^{\mathrm{wp}},\ x_{i+1}^{\mathrm{wp}} - x_i^{\mathrm{wp}}) \tag{10.14}$$

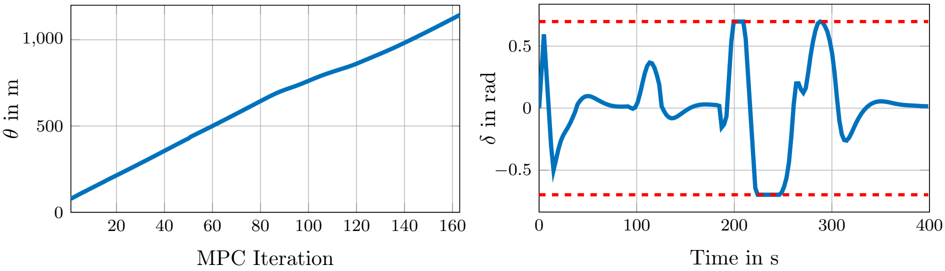


(a) Overview and results of the path-following simulation in the Kiel Canal. Obstacle outlines (dark red) are depicted at the start time of the maneuver with their respective future paths (dashed light blue) along with the controlled vessel's path for the indicator (red), distance (dashed green), signed distance [161] (dash-dotted orange), and proposed signed distance (dotted purple) according to (7.56).

(b) Cross-track error of simulation using indicator (red), distance (dashed green), signed distance according to [161] (dash-dotted orange), and signed distance (dotted purple) constraints according to (7.56) along with their respective root-mean-square errors of $e_{\mathrm{rms}} = \{14.46\,\mathrm{m},\ 15.47\,\mathrm{m},\ 17.78\,\mathrm{m},\ 14.13\,\mathrm{m}\}$.

Figure 10.2.: Path following performance in the north-east-plane (above) and corresponding cross track error (below) for the scenario summarized in Tab. 10.2.

is the desired heading angle of the vessel at waypoint $i = 1, \ldots, 5$. The cross track error is depicted in Fig. 10.2b. In Fig. 10.2a the solutions are depicted for the starting time in the north-east-plane with a map of the Kiel Canal and the relevant obstacles during the maneuver. The (true) future trajectories of the obstacles are depicted in dashed light blue. Note that the predicted future positions of the obstacles are estimated for each MPC iteration based on their respective currently observed heading angle and average observed speed within the last minute of AIS data transmission if that particular vessel lies in the simulated radar range of the controlled vessel for that amount of time. Otherwise, the speed is estimated based on the most recent transmitted data transmitted using AIS. The actual extrapolation of future positions is done using a linear prediction for the horizon



(a) Path parameter that is strictly monotonically increasing, i.e., $\dot{\theta} > 0, \ \forall t$ and which is parameterized by the arc length of the path.

(b) Rudder angle (blue) with actor limits (dashed red).

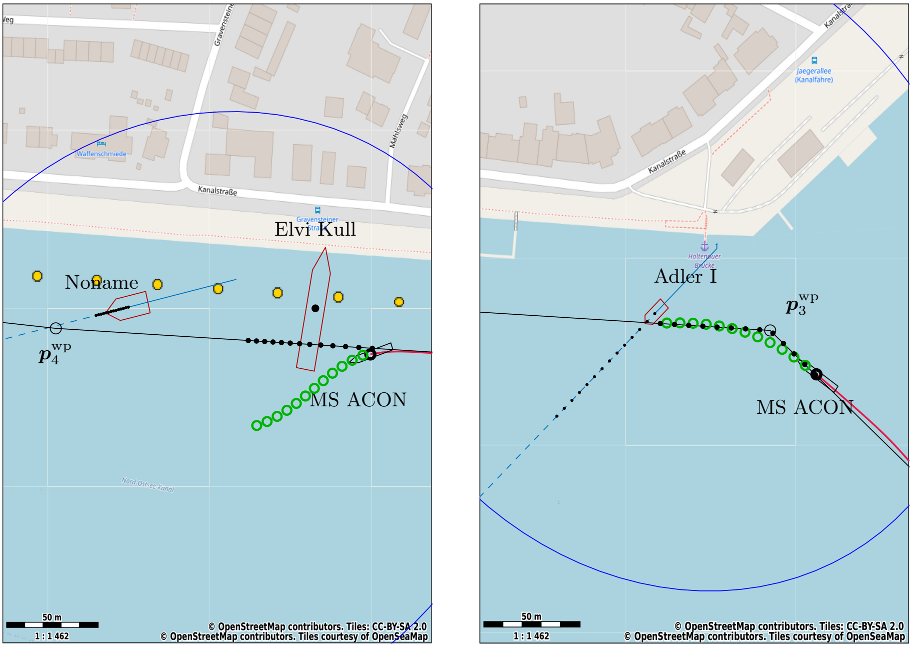
Figure 10.3.: Time evolution of path parameter (left) and rudder angle (right) for the scenario summarized in Tab. 10.2 for the indicator approach.

	Without culling	With culling	Reduction in %
$\bar{n}_{w_\lambda}$	133.79	60.08	55.09
$\bar{n}_{w_\mu}$	133.79	38.72	71.06
$\bar{n}_{w_\lambda} + \bar{n}_{w_\mu}$	267.58	98.80	63.08
$\bar{K}$	1.44	1.16	19.44

Table 10.3.: Comparison of average (taken over all iterations where obstacles are present) number of dual decision variables associated with the controlled vessel $\bar{n}_{w_\lambda}$ and with the obstacles $\bar{n}_{w_\mu}$ and number of obstacles per MPC iteration $\bar{K}$ before and after the culling procedure for the indicator constraint simulation.

time. The rudder input that realizes the depicted trajectory is shown in Fig. 10.3 for the indicator approach along with the time evolution of the path parameter θ. As can be seen, the path parameter is strictly monotonically increasing, i.e., $\dot{\theta} > 0, \ \forall t$ or, in other words, the reference path points $\boldsymbol{p}(\theta)$ always progress in the direction of the path.

The collision avoidance task is also achieved by the predictive controller with slightly different trajectories depending on the specific implementation of the collision avoidance constraints. In this context, it is always assumed that the obstacles are non-cooperative, i.e., they do not actively try to avoid a collision with the controlled vessel. Also note that we do not take into account the COLREGs in this simulation but rather consider each (possible) encounter as a last minute maneuver to avoid collision at any cost. The vessel "Elvi Kull" is seen to block the path of the controlled vessel, which results in a collision avoidance maneuver where the cross track error increases as seen in Fig. 10.2b at approximately $t = 210$ s. The only major difference of trajectories w.r.t. the collision avoidance constraint implementation is seen for the signed distance constraint according to [161] where the controlled vessel needs to evade a collision with the obstacle "Noname" and is seen to pass that particular obstacle with a larger cross track error than for the other implementations.

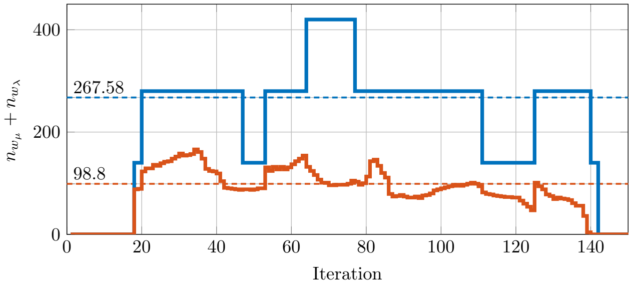

Kanalstraße
Gravensteiner
Mühlsweg
Waffenschmiede
Elvi Kull
Noname
MS ACON
$\boldsymbol{p}_4^{\mathrm{wp}}$
Nord-Ostsee-Kanal
50 m
1 : 1 462
© OpenStreetMap contributors. Tiles: CC-BY-SA 2.0
© OpenStreetMap contributors. Tiles courtesy of OpenSeaMap
Jaegerallee (Kanalfähre)
Holtenauer Brücke
Adler I
$\boldsymbol{p}_3^{\mathrm{wp}}$
MS ACON
50 m
1 : 1 462
© OpenStreetMap contributors. Tiles: CC-BY-SA 2.0
© OpenStreetMap contributors. Tiles courtesy of OpenSeaMap

(a) Evading maneuver for the obstacle "Elvi Kull". The further the predicted controlled vessel's position (green circles) deviates from the reference path (black line) the slower the predicted path points (bold black dots) progress, which is due to (10.9).

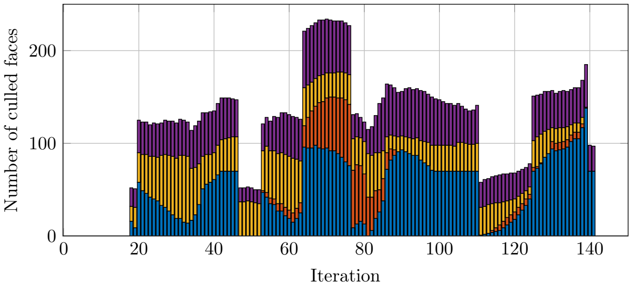
(b) Transition from the first line segment defined by $(\boldsymbol{p}_1, \boldsymbol{p}_2)$ to the next line segment with the obstacle "Adler I" that crosses the path.

Figure 10.4.: Snapshot of MPC iteration $i = 85$ (left) and $i = 45$ (right) for the indicator simulation with predicted controlled vessel positions (green circles), radar radius (dark blue), reference path points (bold black dots), and obstacles (dark red) with obstacles' past path (light blue), predicted positions (black dots), and future path (dashed light blue).

The performance of the culling procedure is depicted for the indicator approach in Fig. 10.5, which shows the reduction of the number of decision variables considered in the MPC problem. In Fig. 10.5a the total number of dual decision variables without culling is shown in blue and the number of dual decision variables after the culling procedure is shown in orange. In Tab. 10.3 the average number of dual decision variables associated with the controlled vessel $\bar{n}_{w_\lambda}$ and with the obstacles $\bar{n}_{w_\mu}$ and the number of obstacles $\bar{K}$ per MPC iteration before and after the culling procedure is shown together with the overall reduction in decision variables and obstacles. The culling procedure is seen to reduce the number of decision variables by approximately 63.08 % and the number of obstacles by 19.44 %. Figure 10.5b shows the number of culled dual decision variables for each of the four culling conditions namely, the frustum- (blue), occlusion- (red), obstacle backface (yellow), and controlled vessel backface (purple) condition. The majority of culled decision variables is due to the backface conditions.



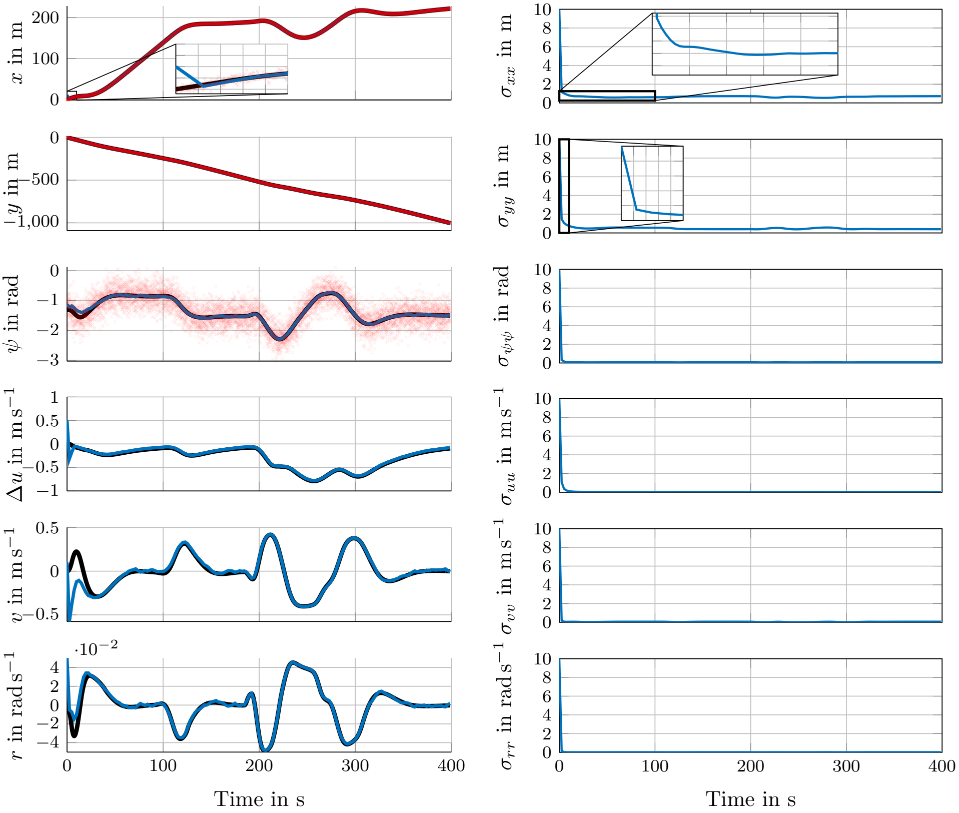
(a) Total number of dual decision variables before (blue) and after (orange) the culling procedure or each iteration with averages (taken over all iterations where obstacles are present) (dashed blue and orange, respectively).



(b) Number of faces culled by frustum- (blue), occlusion- (red), obstacle backface (yellow), and controlled object backface (purple) condition.

Figure 10.5.: Performance of the culling procedure for the indicator approach.

The EKF is initialized with a large initial state error and converges to the true state as seen in Fig. 10.6.



(a) Target pose $\boldsymbol{\eta}^*$ (violet), simulated state $\boldsymbol{x}$ (black), estimated state $\hat{\boldsymbol{x}}$ (blue), and noisy measurements $\boldsymbol{y}$ (red crosses).

(b) Square root of diagonal elements of state error covariance matrix P (blue).

Figure 10.6.: Vessel states $\boldsymbol{x}$ (left) and corresponding estimation error standard deviation P (right) for the predictive path following with full body indicator obstacle constraints for the scenario in Tab. 10.2.

Part IV.

Conclusion and Perspectives for Future Research

11. Conclusions

In this thesis the mathematical models of MSVs with 3DOF by means of Newton's and Euler's second law of motion are derived rigorously. Application of these laws results in a coupled ODE describing the rigid body dynamics of the vessel. Additional forces and moments are taken into account and separated into acceleration- and velocity-dependent forces and moments as well as environmental and controlled forces. Investigation of the acceleration-dependent forces and moments by means of a Taylor series approximation motivate the notion of an added mass matrix that is included to the rigid body mass matrix. Analogously, velocity-dependent forces and moments result in a damping matrix that is a sum of linear and nonlinear damping terms. Furthermore, the analysis of the velocity-dependent terms results in the notion of a so-called added mass Coriolis matrix that is included in the rigid body Coriolis matrix. A link between the classical Abkowitz model and the second-order modulus model is established by means of this Taylor series approximation of the nonlinear damping terms. In this work, focus is put on on wind forces as environmental disturbances and moments acting on the vessel. To this end, the relationship between the MSVs geometry, its relative motion through the wind and the resulting forces and moments that can be approximated mathematically as a function of the relative quantities and the ships geometric dimensions is analyzed. Forces and moments due to the propulsion system are also modeled. To this end, the notion of effectors and actuators is introduced, which map the actual control surfaces such as, e.g., propeller revolutions per minute and rudder angles and their respective locations w.r.t. the CG of the vessel to the resulting forces and moments. Two of the most widely-used effectors are introduced and modeled, namely, propeller-rudder systems and tunnel thrusters. Furthermore, the conditions for course stability of the resulting nonlinear model are derived.

A nonlinear controller is proposed that is used to achieve a dynamic positioning task. It is based on the feedback linearization of the nonlinear 3DOF vessel model that consists of a state feedback law, which linearizes and decouples the nonlinear system dynamics and enables the use of a linear controller. The latter is designed using pole placement of the resulting linear system. In this context the property of differential flatness is discussed and it is shown that the 3DOF vessel model is differentially flat with the pose as a flat output, which is a side effect of the system being feedback linearizable.

Control allocation plays an important part in the control loop especially in dynamic positioning problems. In this thesis, the concept of control allocation is introduced that is used to link commanded generalized forces to actual control inputs. To this end, focus is put on time invariant actor models, i.e., it is assumed that underlying low-level controllers are used to achieve the commanded set points of the control allocation. With this, the control allocation problem consists of a constrained static optimization problem that can be solved using numerical methods such as, e.g., interior-point methods. Therefore, a general purpose interior-point SQP line search algorithm is proposed to highlight the complexity involved in solving these types of problems.

An alternative high-level control concept is proposed which is based on the optimization-based approach. To this end, a nonlinear MPC problem is formulated that is solved at each sampling time to minimize a user-defined cost function subject to the nonlinear system dynamics input and state constraints. The latter constraints are primarily important and used achieve collision avoidance with other vessels and obstacles. To this end, a dual approach is proposed to incorporate collision avoidance constraints that are able to represent the obstacles' and controlled vessel's geometry by means of convex polyhedra. This makes the proposed method especially suited for applications in confined areas. The resulting dynamic optimization problem is solved on a receding horizon using a direct simultaneous method, which transcribes the continuous-time OCP into a large finite-dimensional nonlinear programming problem. The latter, in turn, is solved using state-of-the-art software packages such as, e.g., SNOPT. To reduce complexity, a culling method is proposed that is able to reduce the number of constraints and dual decision variables associated with the collision avoidance constraints in the resulting NLP problem.

A fundamental part of the control loop is the reconstruction of the vessels state, which is used in both of the high-level control schemes. To this end, the EKF, the de-facto standard for nonlinear state estimation, is summarized and several extensions to the standard EKF are discussed such as an iterated EKF, which re-iterates the measurement update step of the EKF to improve the accuracy of the state estimate and, essentially, constitutes a higher-order filter estimate. Furthermore, we enhance the IEKF with an adaptive strategy to estimate diagonal elements of the process and measurement noise covariance matrices based on the filter innovation sequence, which results in an AIEKF. To this end, the sampled covariance matrix of the innovations sequence is approximated using either a moving average filter or a recursive exponentially-weighted moving average filter. The former approximation can be further improved by, e.g., making use of the RTS smoother, which improves the accuracy of the innovation sequence for the moving average filter.

The proposed high-level controllers, the control allocation and the proposed EKF and its extensions are evaluated in two simulation studies. The first simulative study illustrates a dynamic positioning problem of the Cybership II, which is modeled with a nonlinear 3DOF model and includes a propulsion model that consists of two propeller-rudder systems and a bow tunnel thruster. To this end, the proposed nonlinear feedback linearization is used in conjunction with the control allocation to achieve stabilize the vessel's pose at the origin. Furthermore, the proposed AIEKF is used to estimate the vessel's state and disturbances acting on the vessel. The latter estimate is then used to compensate for the disturbances in the control loop. In this context, two different observer models are used, namely, a lumped model and a combined lumped and wind disturbance model, where the former assumes that all disturbances acting on the vessel can be modeled as one time invariant quantity, and the latter separates the disturbances into forces and moments caused by wind and other forces. In this context, the wind direction and speed are assumed to be measurable and an approximate wind model is added to the estimator. The second simulative study is based on a nonlinear autopilot model that is used to perform a path-following task in the Kiel Bay Area using AIS data to obtain obstacle information. This study makes use of the MPC controller with dual collision avoidance constraints and culling to reduce the complexity of the resulting NLP problem.

12. Perspectives for Future Research

The focus on this thesis lies on 3DOF surface vessel models, which neglect, e.g., wave-induced motions of the vessel. However, the concepts presented in this thesis can be applied to 6DOF models as well where extensions to the observer model are necessary such as, e.g., estimating the amplitude and frequency of the wave-induced motions [50, 126]. Furthermore, the actor models presented in this thesis are tailored to the specific thruster configuration of the Cybership II, which consists of two propeller-rudder systems and a bow tunnel thruster. Therefore, analysis can be extended to include other types of effectors, such as azimuth thrusters, and Voith-Schneider propellers, which are especially complex. Additionally, the presented control algorithms assume that the model and propulsion parameters are known. However, in practice, this is not the case, and the parameters have to be estimated online or identified offline using methods presented in, e.g., Fossen et al. [52], Wirtensohn et al. [157], Skjetne et al. [143], Skjetne [142].

In this thesis, a control allocation is used, i.e., the explicit actor models are taken into account in the static optimization problem as is typical, e.g., in aircraft applications [66, 67, 68]. Alternatively, a thrust allocation can be applied, where only the geometric arrangement of the effectors is considered as part of the optimization problem and the actual control surfaces are extracted separately from the solution of the optimization problem [142, Sec. B.1.4], [89, 88]. On the one hand, this latter approach simplifies the complexity of the thrust allocation problem but, on the other hand, shifts part of the complexity downstream of the control loop. However, this approach may be suitable to avoid the pitfall of non-monotonicity that results in occasional erroneous solutions [67]. Furthermore, the actor dynamics can be taken into account, which reflects the fact that the actuators are not able to change their set points instantaneously. When this is taken into account, the control allocation problem is represented as an OCP [21], i.e., a dynamic optimization problem.

The proposed predictive dual collision avoidance scheme proves to be a promising solution for confined environments and is used in the context of, e.g., autonomous parking [160]. The combination with the proposed culling techniques enables a more efficient solution of the resulting OCP. However, the culling conditions need to be evaluated based on predictions of the other obstacles and the controlled vessel's movement both of which are subject to uncertainties. This uncertainty becomes more and more prevalent for increasing prediction horizons. Therefore, the culling conditions are evaluated in a very conservative manner to avoid collisions, i.e., only if all conditions agree that an obstacle face fulfills all conditions for all predicted time steps is it culled from the OCP. A remedy to this problem is to evaluate the proposed culling conditions in each iteration of the numerical procedure instead of evaluating it based on predictions [57]. However, this comes with the drawback of increasing the computational complexity of the numerical procedure and raises the question of the applicability of this approach.

Furthermore, estimation of the vessel states and disturbances is investigated using an AIEKF. This approach can be extended to estimate vessel parameters, as shown in, e.g., Fossen et al. [52], where time invariance of the parameters is added in form of an ODE to the observer model similar to the disturbance estimation approach in this thesis. Furthermore, the AIEKF could be employed to estimate wave-induced motion of the vessel, as shown in Popov et al. [126], where the observer model is extended with a second order system and the wave frequency in each DOF. This information is especially relevant in higher seas. In the context of real world applicability, sensor fusion, sensor placement, faulty measurements, and missing measurements are important topics to be investigated [148, 11, 12]. The proposed observer models can also be included using the UKF, which copes with nonlinear models using the unscented transformation instead of successive linearization [152], which might improve the estimation accuracy in the context of parameter estimation since this inherently induces nonlinearities to any dynamic model.

The simulative studies show promising results for both the predictive path following and the nonlinear dynamic positioning control designs. In the dynamic positioning problem, the fact that tunnel thrusters are only effective for low speeds, say, around five knots for a passenger ferry of normal size is ignored [21]. Furthermore, the effect of wave-induced motions of the vessel is neglected, which is especially relevant in higher seas [50, 158, 126]. A very important extension to the proposed control designs is to take into account the COLREGs, which are not considered in this thesis. Different approaches to tackle this problem in the context of optimal control usually involves altering the cost function appropriately and introducing a module that detects situations in which the cost function must be altered to achieve the desired COLREGs-compliant maneuver [37]. Moreover, an interesting next step is the real world application of the proposed schemes, which comes with a variety of different challenges.

A. Appendix

This chapter provides additional information that is not part of the main part of the thesis. In particular, two example applications for the interior-point SQP line search method proposed in Sec. 6.2.2 are presented and compared with `fmincon`, a state-of-the-art general purpose numerical solver embedded in `MATLAB`. Furthermore, an extensive derivation of the proof used in Chapter 7 is provided. Lastly, further insight is given into the Jacobian matrices needed by the AIEKF proposed in Chapter 8 and used in the simulative studies in Chapter 9.

A.1. Example Problems for the Numerical Solution of NLPs

In this section we present two example problems that are solved with the algorithm presented in Sec. 6.2.2. To this end, the solution of a hanging chain is presented that represents a relatively small static optimization problem with linear cost function, nonlinear equality, and linear equality constraints [19]. Furthermore a solution to an energy-optimal dynamic optimization problem is presented that calculates a swing-up of an inverted pendulum on a cart. The latter represents a direct simultaneous approach that discretizes the dynamic quantities in time, which results in a medium-scale static optimization problem with nonlinear cost and equality constraints, and linear inequality constraints.

A.1.1. Static Optimization: Hanging Chain

The hanging chain problem consists of multiple linked segments that are fixed at the first and second suspension points of the chain $\boldsymbol{p}_0 = [\bar{x}_0,\ \bar{y}_0]^\top$ and $\boldsymbol{p}_{n_s} = [\bar{x}_{n_s},\ \bar{y}_{n_s}]^\top$, respectively. The goal of this optimization is to minimize the potential energy

$$E_{\text{pot}}(\boldsymbol{w}) = \sum_{i=1}^{n_l} L_i \frac{y_i + y_{i-1}}{2} \tag{A.1}$$

of the chain, where n_s is the number of segments, and L_i is the length of the i-the chain segment. See also Fig. A.1 for an illustration of the variables. Since the chain is rigid, the length of the individual segments is fixed, which is expressed using equality constraints, i.e.,

$$(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 - L_i^2 = 0, \qquad i = 1, \ldots, n_s. \tag{A.2}$$

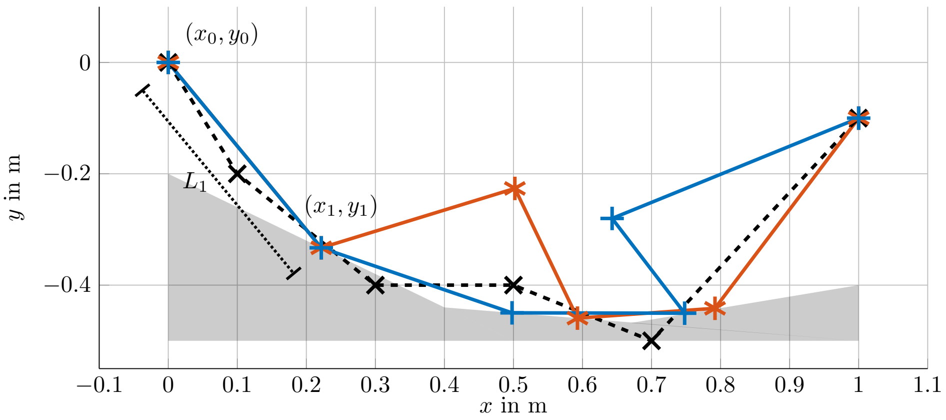


Figure A.1.: Example solution of the hanging chain problem with initial guess (dashed black), `fmincon`'s solution (orange), where $c(\boldsymbol{x}^*) = -0.4346$, proposed algorithm's solution (blue) with $c(\boldsymbol{x}^*) = -0.4456$, and the inequality constraints modeling the floor (shaded gray).

Furthermore, if there is a floor, the solution must ensure that the chain is above or on the floor, i.e.,

$$m_j x_i + b_j - y_i \leq 0, \qquad i = 1, \dots, n_s, \quad j = 1, \dots, n_f, \tag{A.3}$$

where m_j is the slope and b_j the ordinate offset of the j-th floor segment and n_f is the number of floor segments. In principle x_0, y_0, x_{n_s}, y_{n_s} can be excluded from the optimization problem since they are fixed to their respective suspension points $\boldsymbol{p}_0$, $\boldsymbol{p}_{n_s}$. With this, the optimization problem is expressed mathematically as

$$c(\boldsymbol{w}) = E_{\text{pot}}(\boldsymbol{w}) \tag{A.4a}$$

s.t.

$$\boldsymbol{g}(\boldsymbol{w}) = \begin{bmatrix} (x_1 - x_0)^2 + (y_1 - y_0)^2 - L_1^2 \\ \vdots \\ (x_{n_s} - x_{n_s-1})^2 + (y_{n_s} - y_{n_s-1})^2 - L_{n_s}^2 \\ x_{n_s} - \bar{x}_{n_s} \\ y_{n_s} - \bar{y}_{n_s} \end{bmatrix} = \boldsymbol{0}, \tag{A.4b}$$

$$\boldsymbol{h}(\boldsymbol{w}) = \begin{bmatrix} m_1 x_1 + b_1 - y_1 \\ \vdots \\ m_1 x_{n_s} + b_1 - y_{n_s} \\ \vdots \\ m_{n_f} x_1 + b_{n_f} - y_1 \\ \vdots \\ m_{n_f} x_{n_s} + b_{n_f} - y_{n_s} \end{bmatrix} \leq \boldsymbol{0} \tag{A.4c}$$

with $\boldsymbol{w} = [x_1, \ldots, x_{n_s}, y_1, \ldots, y_{n_s}]^\top$. In Fig. A.1, the solution of this problem is shown for a chain with $n_s = 5$ segments and $n_f = 3$ floor segments. The initial guess is shown as a dashed black line. Note that this initial solution is not feasible, e.g., it does not take into account the fixed segment lengths or the floor constraints. The solution of `fmincon` is also depicted in Fig. A.1 as an orange line, and the solution of proposed algorithm as a blue line. As can be seen, the install guess is not feasible, i.e., the chain is below the floor and, thus, violates the inequality constraints. Furthermore, the chain links lengths violate the equality constraints that ensure the correct length. In this particular example, the proprietary algorithm achieves a slightly better solution than `fmincon`, i.e., $c(\boldsymbol{x}^*) = -0.4456$ vs. $c(\boldsymbol{x}^*) = -0.4346$ obtained by `fmincon`.

A.1.2. Dynamic Optimization: Inverted Pendulum on a Cart

In this example solve an optimal control swing-up problem of an inverted pendulum on a cart, i.e., we aim to find an input u to the underlying dynamical system such that the energy of the control effort is minimized. The system states $\boldsymbol{x} = [x_1, x_2, x_3, x_4]^\top = [s, v, \theta, \omega]^\top$ are the time-varying quantities and describe the cart position and velocity w.r.t. the y axis, and angle and angular velocity of the pole w.r.t. the z axis. Note that $\theta = \varphi - \pi$. The system dynamics are expressed in terms of these states and the input to the system u, which is the force applied to the cart, i.e.,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u), \qquad t > t_0, \quad \boldsymbol{x}(t_0) = \hat{\boldsymbol{x}}_0, \tag{A.5}$$

where $\hat{\boldsymbol{x}}_0$ is the initial state of the system, $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$, and $u(t) \in \mathbb{R}^{n_u}$ and

$$\boldsymbol{f}(\boldsymbol{x}, u) = \begin{bmatrix} x_2 \\ -\frac{\left(l^3\, m^2 + Jlm\right) \sin(x_3) {x_4}^2 + g\, l^2\, m^2 \cos(x_3) \sin(x_3) + \left(-b\, l^2 m - Jb\right) x_2 + \left(l^2 m + J\right) u}{l^2\, m^2 \cos(x_3)^2 - l^2\, m^2 + (-M\, l^2 - J) m - JM} \\ x_4 \\ \frac{l^2\, m^2 \cos(x_3) \sin(x_3) {x_4}^2 + \left(gl\, m^2 + Mglm\right) \sin(x_3) + (lmu - blm\, x_2) \cos(x_3)}{l^2\, m^2 \cos(x_3)^2 - l^2\, m^2 + (-M\, l^2 - J) m - JM} \end{bmatrix}. \tag{A.6}$$

In the following, we consider the energy-optimal swing-up OCP

$$\min_u J(u) = \int_0^{t_\mathrm{f}} \frac{1}{2} u^2(t) \mathrm{d}t \tag{A.7a}$$

s.t.

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u), \quad \boldsymbol{x}(0) = \hat{\boldsymbol{x}}_0, \quad \boldsymbol{x}(t_\mathrm{f}) = \boldsymbol{x}_\mathrm{f} \tag{A.7b}$$

$$u^- \leq u \leq u^+, \tag{A.7c}$$

where

$$\hat{\boldsymbol{x}}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top, \qquad \boldsymbol{x}_\mathrm{f} = \begin{bmatrix} -1 & 0 & \pi & 0 \end{bmatrix}^\top. \tag{A.8}$$

An approximate solution to this problem is obtained using a direct multiple shooting, where the system dynamics are discretized using a trapezoidal scheme, i.e.,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \frac{t_\mathrm{f}}{2N}\Big[\boldsymbol{f}(\boldsymbol{x}_k, u_k) + \boldsymbol{f}(\boldsymbol{x}_{k+1}, u_{k+1})\Big]. \tag{A.9}$$

We also need to approximate the integral, which we also achieve with the trapezoidal rule, i.e.,

$$\int_0^{t_\mathrm{f}} l(\boldsymbol{x}, u)\mathrm{d}t \approx \frac{t_\mathrm{f}}{2N}\sum_{k=0}^{N-1} l(\boldsymbol{x}_k, u_k) + l(\boldsymbol{x}_{k+1}, u_{k+1}) \tag{A.10}$$

Furthermore, we model the input $u(t)$ as a piece-wise linear function for each time step $\frac{t_\mathrm{f}}{N}$ which leads to an NLP

$$\min_{\boldsymbol{w}\in\mathbb{R}^{n_w}} \; c(\boldsymbol{w}) \tag{A.11}$$

$$\text{s.t.} \tag{A.12}$$

$$\boldsymbol{g}(\boldsymbol{w}) = \boldsymbol{0}, \tag{A.13}$$

$$\boldsymbol{h}(\boldsymbol{w}) \leq \boldsymbol{0}, \tag{A.14}$$

where $\boldsymbol{w} = [\boldsymbol{w}_x^\top, \boldsymbol{w}_u^\top]^\top = [\boldsymbol{x}_0^\top, \ldots, \boldsymbol{x}_N^\top, u_0, \ldots, u_N]^\top$ are the decision variables and N is the number of discretization steps. Furthermore,

$$\boldsymbol{g}(\boldsymbol{w}) = \begin{bmatrix} \boldsymbol{g}^\mathrm{bc}(\boldsymbol{w}) \\ \boldsymbol{g}^\mathrm{ode}(\boldsymbol{w}) \end{bmatrix}, \tag{A.15}$$

where

$$\boldsymbol{g}^\mathrm{bc}(\boldsymbol{w}) = \begin{bmatrix} \boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0 \\ \boldsymbol{x}_N - \boldsymbol{x}_\mathrm{f}. \end{bmatrix} \tag{A.16}$$

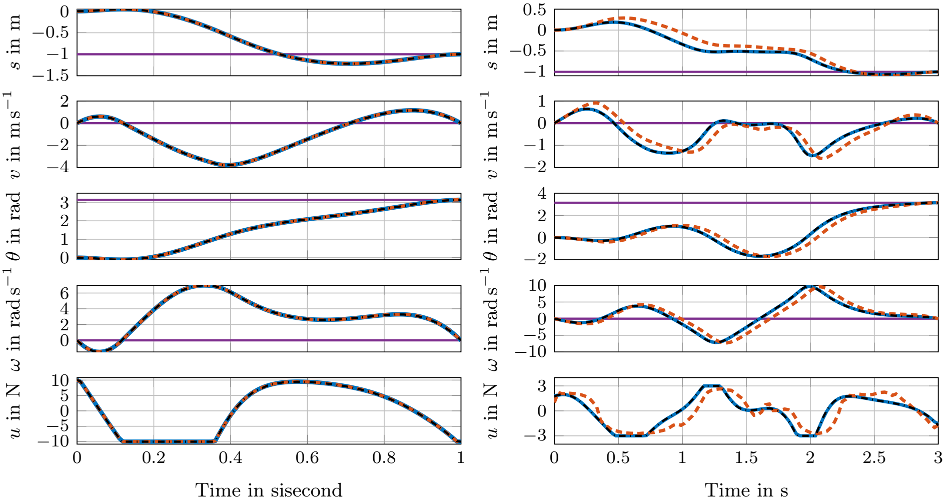
is the vector of boundary constraints, and

$$\boldsymbol{g}^\mathrm{ode}(\boldsymbol{w}) = \begin{bmatrix} \boldsymbol{g}_0^\mathrm{ode}(\boldsymbol{x}_1, \boldsymbol{x}_0, u_0, u_1) \\ \vdots \\ \boldsymbol{g}_{N-1}^\mathrm{ode}(\boldsymbol{x}_N, \boldsymbol{x}_{N-1}, u_{N-1}, u_N) \end{bmatrix}, \tag{A.17}$$

is the vector of ODE (defect) constraints, where

$$\boldsymbol{g}_k^\mathrm{ode}(\cdot) = -\boldsymbol{x}_{k+1} + \boldsymbol{x}_k + \frac{t_\mathrm{f}}{2N}\Big[\boldsymbol{f}(\boldsymbol{x}_k, u_k) + \boldsymbol{f}(\boldsymbol{x}_{k+1}, u_{k+1})\Big] \tag{A.18}$$

for $k = 0, \ldots, N-1$ realizes the trapezoidal integration of the ODE. The bound constraints



(a) Input constraints $u^+ = -u^- = 10\,\mathrm{N}$ with $c(\boldsymbol{x}^*) = 58.73$ (`fmincon`) and $c(\boldsymbol{x}^*) = 58.73$ (proposed algorithm).

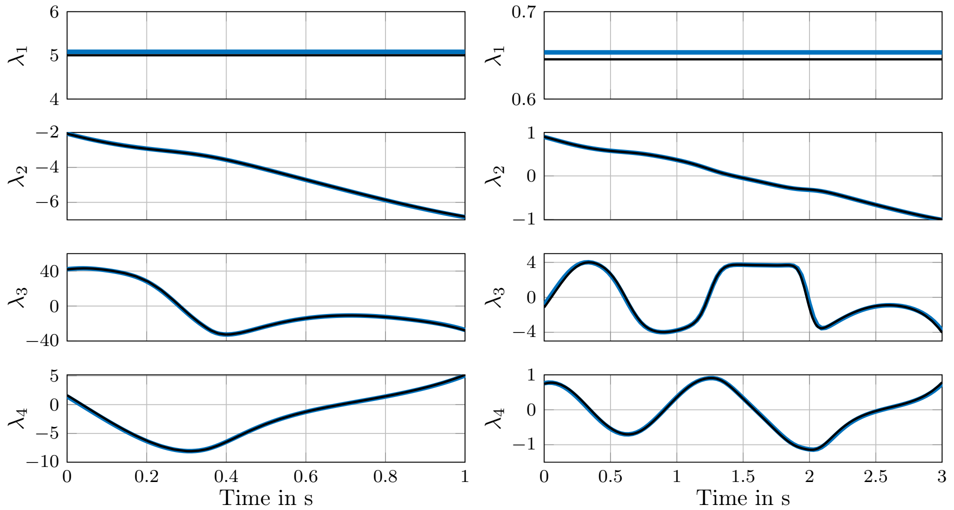
(b) Input constriants $u^+ = -u^- = 3\,\mathrm{N}$ with $c(\boldsymbol{x}^*) = 9.89$ (`fmincon`), $c(\boldsymbol{x}^*) = 9.33$ (proposed algorithm).

Figure A.2.: Solutions to the energy optimal swing-up problem of the inverted pendulum on a cart for $t_\mathrm{f} = 1\mathrm{s}$ (left) and $t_\mathrm{f} = 3\mathrm{s}$ (right) with `fmincon`'s solution (dashed orange), proposed algorithm's solution (blue), final values $\boldsymbol{x}_\mathrm{f}$ (violet), and `bvp4c`'s solution (dash-dotted black).

from the swing-up problem can be expressed in the general NLP form

$$\boldsymbol{h}(\boldsymbol{w}) = \begin{bmatrix} u_0 - u^+ \\ \vdots \\ u_N - u^+ \\ u^- - u_0 \\ \vdots \\ u^- - u_N \end{bmatrix}. \tag{A.19}$$

If we set $N = 100$, this leads to $n_g = 2n_y + Nn_y = 408$ (nonlinear) equality constraints and $n_h = 2N = 200$ (linear) inequality constraints with $n_w = (N+1)(n_y + n_u) = 505$ decision variables. As can be seen in Fig. A.2, the solutions of `fmincon` and the proposed algorithm coincide for $t_\mathrm{f} = 1\mathrm{s}$ but differ for $t_\mathrm{f} = 3\mathrm{s}$, where the proposed algorithm obtains a slightly lower cost function value (at the price of taking much more time to solve...). It is also not very surprising, that the energy is an order of magnitude larger if the final time is smaller, i.e., much more energy is needed for that maneuver. Furthermore, if you plot the Lagrange multipliers for the defect constraints over time, we get an estimate of the



(a) Final value for Lagrange multipliers $\boldsymbol{\lambda}(t_\mathrm{f}) = \boldsymbol{\mu} = [5,\ -6.81,\ -27.87,\ -5.17]^\top$.

(b) Final value for Lagrange multipliers $\boldsymbol{\lambda}(t_\mathrm{f}) = \boldsymbol{\mu} = [0.64,\ -1,\ -4,\ 0.78]^\top$.

Figure A.3.: Lagrange multipliers for the defect constraints using the proposed algorithm (blue) and `bvp4c`'s costate variables (black) for $t_\mathrm{f} = 1\mathrm{s}$ (left) and $t_\mathrm{f} = 3\mathrm{s}$ (right).

adjoint states $\boldsymbol{\lambda}$ that arise as part of the necessary optimality conditions of the dynamic optimization problem, the so-called Euler-Lagrange Equations (ELG)

$$\begin{aligned}
\dot{\boldsymbol{x}}^* &= \nabla_{\boldsymbol{\lambda}} \mathcal{H}(\boldsymbol{x}^*, u^*, \boldsymbol{\lambda}^*) & \boldsymbol{x}(0) &= \hat{\boldsymbol{x}}_0, \quad \boldsymbol{g}(t_\mathrm{f}, \boldsymbol{x}(t_\mathrm{f})) = \boldsymbol{0} & \text{(A.20)}\\
\dot{\boldsymbol{\lambda}}^* &= -\nabla_{\boldsymbol{x}} \mathcal{H}(\boldsymbol{x}^*, u^*, \boldsymbol{\lambda}^*) & \boldsymbol{\lambda}(t_\mathrm{f}) &= \nabla_{\boldsymbol{x}(t_\mathrm{f})} \psi(t_\mathrm{f}, \boldsymbol{x}^*(t_\mathrm{f}), \boldsymbol{\mu}), & \text{(A.21)}\\
u^* &= \operatorname{argmin}_v \{\mathcal{H}(\boldsymbol{x}, v, \boldsymbol{\lambda}) \ : \ u^- \leq v \leq u^+\}, & & & \text{(A.22)}
\end{aligned}$$

which must be fulfilled for a (local) optimizer u and where

$$\mathcal{H}(\boldsymbol{x}, u, \boldsymbol{\lambda}) = l(\boldsymbol{x}, u) + \boldsymbol{\lambda}^\top \boldsymbol{f}(\boldsymbol{x}, u) \tag{A.23}$$

is the Hamiltonian and

$$\psi(t_\mathrm{f}, \boldsymbol{x}(t_\mathrm{f}), \boldsymbol{\mu}) = \varphi(t_\mathrm{f}, \boldsymbol{x}(t_\mathrm{f})) + \boldsymbol{\mu}^\top \boldsymbol{g}(t_\mathrm{f}, \boldsymbol{x}(t_\mathrm{f})) \tag{A.24}$$

an auxiliary function. Therein, $\varphi(\cdot,\cdot)$ is the final cost (which is equal to zero in the above example). These first order necessary optimality conditions constitute a two point boundary value problem (TPBVP) and are hard to solve, especially in the presence of inequality constraints. However, solving for the optimal input with $l(\boldsymbol{x}, u) = \frac{1}{2}u^2$ for input-affine systems, i.e., $\boldsymbol{f}(\boldsymbol{x}, u) = \boldsymbol{f}_0(\boldsymbol{x}) + \boldsymbol{f}_1(\boldsymbol{x})u$ leads to

$$u = \begin{cases} u^-, & \text{if } -\boldsymbol{\lambda}^\top \boldsymbol{f}_1(\boldsymbol{x}) \leq u^- \\ -\boldsymbol{\lambda}^\top \boldsymbol{f}_1(\boldsymbol{x}) & \text{else} \\ u^+, & \text{if } -\boldsymbol{\lambda}^\top \boldsymbol{f}_1(\boldsymbol{x}) \geq u^+ \end{cases} \tag{A.25}$$

This has also been implemented in the example `MATLAB` file for sake of completeness. Essentially, solving the ELG using `bvp4c` is an *indirect* numerical method in that it solves the ELG equations instead of directly addressing the OCP. More specifically, `bvp4c` realizes an indirect collocation to solve the OCP and the optimal input can be expressed as $u = k(\boldsymbol{x}, \boldsymbol{\lambda})$ that is inserted to the ELG, which yields the TPBVP

$$\dot{\boldsymbol{x}}^* = \nabla_{\boldsymbol{\lambda}} \mathcal{H}(\boldsymbol{x}^*, k(\boldsymbol{x}^*, \boldsymbol{\lambda}^*), \boldsymbol{\lambda}^*) \qquad \boldsymbol{x}(0) = \hat{\boldsymbol{x}}_0, \quad \boldsymbol{g}(t_\mathrm{f}, \boldsymbol{x}(t_\mathrm{f})) = \boldsymbol{0} \tag{A.26}$$

$$\dot{\boldsymbol{\lambda}}^* = -\nabla_{\boldsymbol{x}} \mathcal{H}(\boldsymbol{x}^*, k(\boldsymbol{x}^*, \boldsymbol{\lambda}^*), \boldsymbol{\lambda}^*) \qquad \boldsymbol{\lambda}(t_\mathrm{f}) = \nabla_{\boldsymbol{x}(t_\mathrm{f})} \psi(t_\mathrm{f}, \boldsymbol{x}^*(t_\mathrm{f}), \boldsymbol{\mu}). \tag{A.27}$$

The unknown constant $\boldsymbol{\mu}$ can be determined by `bvp4c`. Alternatively, the TPBVP can be augmented by $\dot{\boldsymbol{\mu}} = \boldsymbol{0}$. Indirect methods are known to be highly-reliant on a good initial guess. The often-used linear interpolation between $\hat{\boldsymbol{x}}_0$ and $\boldsymbol{x}_\mathrm{f}$ for the individual states with all initial adjoint states zeros as an initial guess failed to result in convergence of the solver. Therefore, for the states, `fmincon`'s solution, which is not as good as the proposed algorithm's, has been passed as an initial guess to `bvp4c` with zero costates.

A.2. Proof of (7.11) using Farkas' Lemma

Expanding (7.10) with $I = I^\top$ gives

$$A^\top \boldsymbol{\mu} + \boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2 = \boldsymbol{0}, \tag{A.28a}$$

$$\boldsymbol{b}^\top \boldsymbol{\mu} + (\boldsymbol{p}^{\mathcal{V}})^\top \boldsymbol{\sigma}_1 - (\boldsymbol{p}^{\mathcal{V}})^\top \boldsymbol{\sigma}_2 < 0, \tag{A.28b}$$

$$\boldsymbol{\mu} \geq \boldsymbol{0},\ \boldsymbol{\sigma}_1 \geq \boldsymbol{0},\ \boldsymbol{\sigma}_2 \geq \boldsymbol{0}. \tag{A.28c}$$

Subsequenty, (A.28a) is solved for $\boldsymbol{\sigma}_2$, i.e.,

$$\boldsymbol{\sigma}_2 = \boldsymbol{\sigma}_1 + A^\top \boldsymbol{\mu},$$

which is inserted in (A.28b) to give

$$A^\top \boldsymbol{\mu} + \boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2 = \boldsymbol{0}, \tag{A.29a}$$

$$(\boldsymbol{b} - A\boldsymbol{p}^{\mathcal{V}})^\top \boldsymbol{\mu} < 0, \tag{A.29b}$$

$$\boldsymbol{\mu} \geq \boldsymbol{0},\ \boldsymbol{\sigma}_1 \geq \boldsymbol{0},\ \boldsymbol{\sigma}_2 \geq \boldsymbol{0}. \tag{A.29c}$$

Apparently, $\boldsymbol{z}(t) = \boldsymbol{\sigma}_2(t) - \boldsymbol{\sigma}_1(t) \in \mathbb{R}^n$ is an arbitrary vector and can thus be ignored. Multiplying (A.29b) by minus one gives the desired result

$$\boldsymbol{p}^{\mathcal{V}} \notin O \Leftrightarrow \exists \boldsymbol{\mu} \geq \boldsymbol{0} : (A\boldsymbol{p}^{\mathcal{V}} - \boldsymbol{b})^\top \boldsymbol{\mu} > 0. \tag{A.30}$$

A.3. EKF Jacobian Matrices for the Dynamic Positioning Simulation

In the following, the Jacobian matrices for the EKF in Sec. 9 are given as far as this is reasonably possible. If a Jacobian is not given explicitly here, it is generated using `CasADi`, a symbolic framework for automatic differentiation [5].

A.3.1. Lumped Disturbance Estimation Model

The analytical Jacobians for the lumped estimation model (9.4) are given by

$$\frac{\partial \boldsymbol{f}_{\text{ext}}}{\partial \boldsymbol{x}_{\text{ext}}} = \underbrace{\begin{bmatrix} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} & 0^{(6\times3)} \\ 0^{(3\times6)} & 0^{(3\times3)} \end{bmatrix}}_{\in\mathbb{R}^{9\times9}} + \begin{bmatrix} \frac{\partial \boldsymbol{f}_{\text{d}}}{\partial \boldsymbol{x}} & \frac{\partial \boldsymbol{f}_{\text{d}}}{\partial \boldsymbol{d}} \end{bmatrix} \tag{A.31}$$

with $\frac{\partial \boldsymbol{f}_{\text{d}}}{\partial \boldsymbol{x}}$, $\frac{\partial \boldsymbol{f}_{\text{d}}}{\partial \boldsymbol{d}}$ given by

$$\frac{\partial \boldsymbol{f}_{\text{d}}}{\partial \boldsymbol{x}} = \begin{bmatrix} 0^{(3\times2)} & \boldsymbol{0} & 0^{(3\times3)} \\ 0^{(3\times2)} & M^{-1}\frac{\partial R_{z,\psi}^{\top}}{\partial \psi}\boldsymbol{d} & 0^{(3\times3)} \\ 0^{(3\times2)} & \boldsymbol{0} & 0^{(3\times3)} \end{bmatrix} \in \mathbb{R}^{9\times6}, \qquad \frac{\partial \boldsymbol{f}_{\text{d}}}{\partial \boldsymbol{d}} = \begin{bmatrix} 0^{(3\times3)} \\ M^{-1}R_{z,\psi}^{\top} \\ 0^{(3\times3)} \end{bmatrix} \in \mathbb{R}^{9\times3}, \tag{A.32}$$

respectively. Furthermore the measurement function Jacobian and input Jacobian evaluate to

$$\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_{\text{ext}}} = \begin{bmatrix} I^{(3\times3)} & 0^{(3\times6)} \end{bmatrix}, \qquad \frac{\partial \boldsymbol{f}_{\text{ext}}}{\partial \boldsymbol{u}} = B_{\tau,\text{ext}}, \tag{A.33}$$

respectively.

A.3.2. Combined Lumped and Wind Disturbance Estimation Model

Analogously to the previous subsection the analytical Jacobians for the combined lumped and wind disturbance model (9.10) are given by

$$\frac{\partial \boldsymbol{f}_{\text{ext}}}{\partial \boldsymbol{x}_{\text{ext}}} = \underbrace{\begin{bmatrix} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} & 0^{(6\times5)} \\ 0^{(5\times6)} & 0^{(5\times5)} \end{bmatrix}}_{\in\mathbb{R}^{11\times11}} + \begin{bmatrix} \frac{\partial \boldsymbol{f}_{\text{d}}}{\partial \boldsymbol{x}} & \frac{\partial \boldsymbol{f}_{\text{d}}}{\partial \boldsymbol{d}} & \frac{\partial \boldsymbol{f}_{\text{d}}}{\partial \boldsymbol{x}_{\text{w}}} \end{bmatrix}. \tag{A.34}$$

Therein, $\frac{\partial f_{\mathrm{d}}}{\partial \boldsymbol{x}}$, $\frac{\partial f_{\mathrm{d}}}{\partial \boldsymbol{d}}$, $\frac{\partial f_{\mathrm{d}}}{\partial \boldsymbol{x}_{\mathrm{w}}}$ are very cumbersome expressions such that the explicit structure is not explicitly given here but generated using `CasADi` [5]. Lastly, the measurement and input Jacobians are formally given by

$$\frac{\partial \boldsymbol{h}_{\mathrm{ext}}}{\partial \boldsymbol{x}_{\mathrm{ext}}} = \begin{bmatrix} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_{\mathrm{ext}}} \\ \frac{\partial \boldsymbol{h}_{\mathrm{w}}}{\partial \boldsymbol{x}_{\mathrm{ext}}} \end{bmatrix}, \qquad \frac{\partial \boldsymbol{f}_{\mathrm{ext}}}{\partial \boldsymbol{u}} = B_{\tau,\mathrm{ext}}, \tag{A.35}$$

respectively, where

$$\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_{\mathrm{ext}}} = \begin{bmatrix} I^{(3\times3)} & 0^{(3\times8)} \end{bmatrix}, \qquad \frac{\partial \boldsymbol{h}_{\mathrm{w}}}{\partial \boldsymbol{x}_{\mathrm{ext}}} = \begin{bmatrix} \frac{\partial V_{\mathrm{w,a}}}{\partial \boldsymbol{x}_{\mathrm{ext}}} \\ \frac{\partial \gamma_{\mathrm{w,a}}}{\partial \boldsymbol{x}_{\mathrm{ext}}} \end{bmatrix}, \tag{A.36}$$

which is also are generated using `CasADi`.

A.4. On Numerical Methods for Solving Optimal Control Problems

To expand on Sec. 6.3.2 and Fig. 6.2, refer to Fig. A.4 that categorizes different mechanisms, which are involved in solving OCPs numerically, namely, solving ODEs numerically, solving NLPs numerically, handling inequality constraints (within an NLP), ensuring (NLP) convergence, providing a search direction, providing gradient information, solving implicit equations, and evaluating integral costs and constraints. Furthermore, it relates these mechanisms to the overarching concepts of numerical integration methods, indirect and direct methods, NLP methods, inequality constraints and root finding methods.

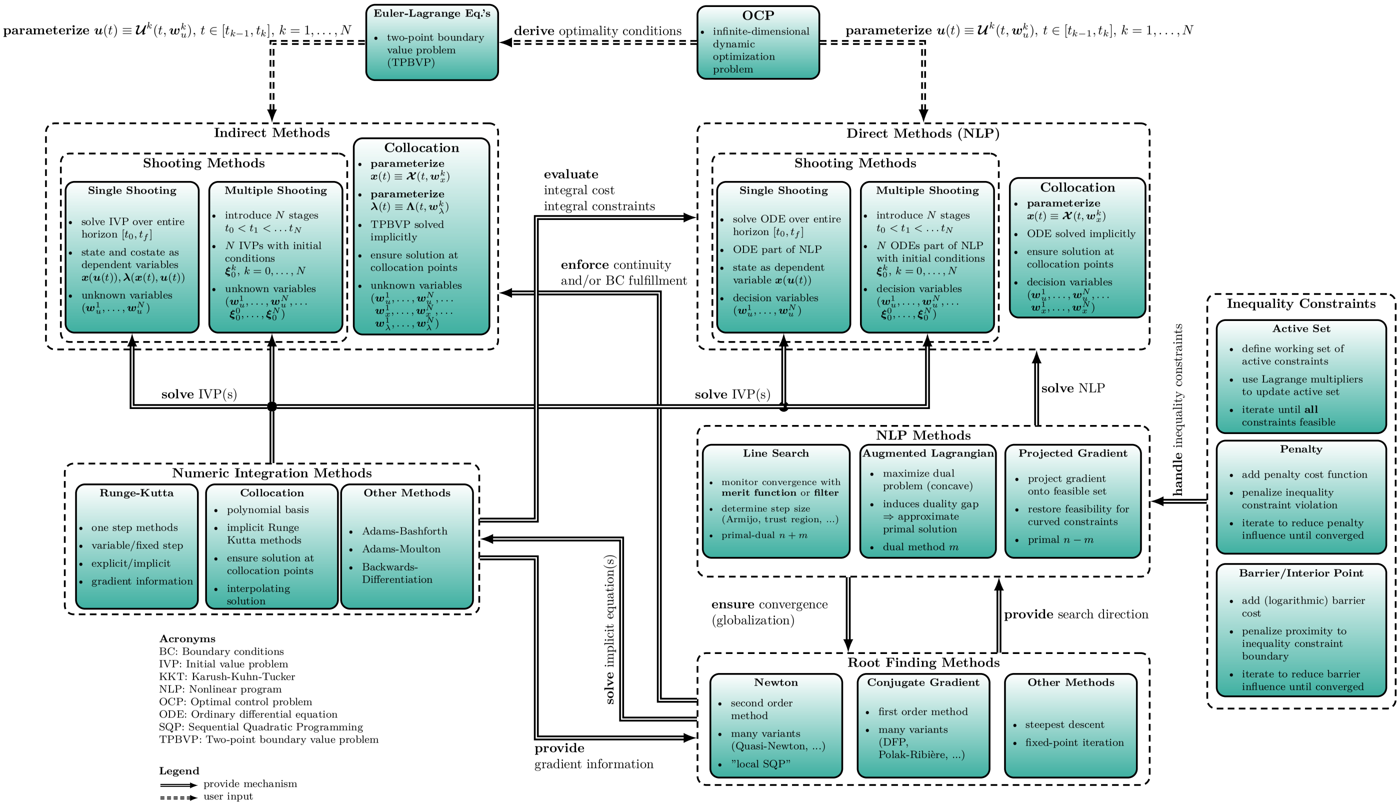

OCP
infinite-dimensional dynamic optimization problem
derive optimality conditions
Euler-Lagrange Eq.'s
two-point boundary value problem (TPBVP)
parameterize $u(t) \equiv \mathcal{U}^k(t, w_u^k)$, $t \in [t_{k-1}, t_k]$, $k = 1, \dots, N$
parameterize $u(t) \equiv \mathcal{U}^k(t, w_u^k)$, $t \in [t_{k-1}, t_k]$, $k = 1, \dots, N$
Indirect Methods
Shooting Methods
Single Shooting
solve IVP over entire horizon $[t_0, t_f]$
state and costate as dependent variables $x(u(t)), \lambda(x(t), u(t))$
unknown variables $(w_u^1, \dots, w_u^N)$
Multiple Shooting
introduce N stages $t_0 < t_1 < \dots t_N$
N IVPs with initial conditions ξ_0^k, $k = 0, \dots, N$
unknown variables $(w_u^1, \dots, w_u^N, \dots \xi_0^0, \dots, \xi_0^N)$
Collocation
parameterize $x(t) \equiv \mathcal{X}(t, w_x^k)$
parameterize $\lambda(t) \equiv \Lambda(t, w_\lambda^k)$
TPBVP solved implicitly
ensure solution at collocation points
unknown variables $(w_u^1, \dots, w_u^N, \dots w_x^1, \dots, w_x^N, \dots w_\lambda^1, \dots, w_\lambda^N)$
Direct Methods (NLP)
Shooting Methods
Single Shooting
solve ODE over entire horizon $[t_0, t_f]$
ODE part of NLP
state as dependent variable $x(u(t))$
decision variables $(w_u^1, \dots, w_u^N)$
Multiple Shooting
introduce N stages $t_0 < t_1 < \dots t_N$
N ODEs part of NLP with initial conditions ξ_0^k, $k = 0, \dots, N$
decision variables $(w_u^1, \dots, w_u^N, \dots \xi_0^0, \dots, \xi_0^N)$
Collocation
parameterize $x(t) \equiv \mathcal{X}(t, w_x^k)$
ODE solved implicitly
ensure solution at collocation points
decision variables $(w_u^1, \dots, w_u^N, \dots w_x^1, \dots, w_x^N)$
evaluate integral cost integral constraints
enforce continuity and/or BC fulfillment
solve IVP(s)
solve IVP(s)
solve NLP
Numeric Integration Methods
Runge-Kutta
one step methods
variable/fixed step
explicit/implicit
gradient information
Collocation
polynomial basis
implicit Runge Kutta methods
ensure solution at collocation points
interpolating solution
Other Methods
Adams-Bashforth
Adams-Moulton
Backwards-Differentiation
solve implicit equation(s)
provide gradient information
NLP Methods
Line Search
monitor convergence with merit function or filter
determine step size (Armijo, trust region, ...)
primal-dual $n + m$
Augmented Lagrangian
maximize dual problem (concave)
induces duality gap ⇒ approximate primal solution
dual method m
Projected Gradient
project gradient onto feasible set
restore feasibility for curved constraints
primal $n - m$
ensure convergence (globalization)
provide search direction
Root Finding Methods
Newton
second order method
many variants (Quasi-Newton, ...)
"local SQP"
Conjugate Gradient
first order method
many variants (DFP, Polak-Ribière, ...)
Other Methods
steepest descent
fixed-point iteration
handle inequality constraints
Inequality Constraints
Active Set
define working set of active constraints
use Lagrange multipliers to update active set
iterate until all constraints feasible
Penalty
add penalty cost function
penalize inequality constraint violation
iterate to reduce penalty influence until converged
Barrier/Interior Point
add (logarithmic) barrier cost
penalize proximity to inequality constraint boundary
iterate to reduce barrier influence until converged
Acronyms
BC: Boundary conditions
IVP: Initial value problem
KKT: Karush-Kuhn-Tucker
NLP: Nonlinear program
OCP: Optimal control problem
ODE: Ordinary differential equation
SQP: Sequential Quadratic Programming
TPBVP: Two-point boundary value problem
Legend
provide mechanism
user input

Figure A.4.: Disambiguation of and mechanisms for numerical OCP methods.

List of Figures

2.1. Schematic control loop structure with GNC of an MSV (dashed) with reference pose $\boldsymbol{\eta}^*$, feedback, commanded, and realized control forces $\boldsymbol{\tau}_{\mathrm{fb}}$, $\boldsymbol{\tau}_{\mathrm{c}}^*$, and $\boldsymbol{\tau}_{\mathrm{c}}$, respectively, optimal control surfaces $\boldsymbol{u}^*$, measurements $\boldsymbol{y}$, measurement noise $\boldsymbol{v}$, real and estimated states $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, respectively, and real and estimated disturbances $\boldsymbol{\tau}_{\mathrm{d}}$ and $\hat{\boldsymbol{\tau}}_{\mathrm{d}}$, respectively. 5

5.1. Schematic relations of the vessel's body-fixed frame $(0_b x_b y_b z_b)$ to the inertial frame $(0_i x_i y_i z_i)$ in the 2D plane, i.e., for $z = \dot{z} = 0$, and with $y_g = z_g = 0$, i.e., the CG lies midships along the x_b-axis. 18

5.2. A force $\boldsymbol{f}^p$ acting at point P induces a moment $\boldsymbol{m}_i^i = \boldsymbol{p}_i^{ip} \times \boldsymbol{f}_i^p = \boldsymbol{p}_i^{ib} \times \boldsymbol{f}_i^p + \boldsymbol{m}_i^b = \boldsymbol{p}_i^{ib} \times \boldsymbol{f}_i^p + \boldsymbol{p}_i^{bp} \times \boldsymbol{f}_i^p$ w.r.t. the inertial frame. Furthermore, it effects a translational motion of the MSV according to (5.11). 21

5.3. Relations of true and apparent wind w.r.t. the NED frame and the body frame, where $\beta_{\mathrm{w,t}}$, $\beta_{\mathrm{w,a}}$ is the angle of attack of the true and apparent wind in the NED frame, respectively, $\gamma_{\mathrm{w,t}}$, $\gamma_{\mathrm{w,a}}$ are the respective angles of attack of the true and apparent wind w.r.t. the body frame, β is the drift angle, $\boldsymbol{v}_{\mathrm{w,t}}$ is the true wind velocity $\boldsymbol{v}_{\mathrm{w,h}} = -[\dot{x},\ \dot{y}]^\top$ is the headwind velocity, and $\boldsymbol{v}_{\mathrm{w,a}}$ is the apparent wind velocity with $V_{\mathrm{w,t}} = ||\boldsymbol{v}_{\mathrm{w,t}}||_2$, $V_{\mathrm{w,a}} = ||\boldsymbol{v}_{\mathrm{w,a}}||_2$. 29

5.4. Example actuator configuration with $n_a = 3$ actuators, namely, a combined propeller-rudder system with rudder angle δ and two tunnel thrusters with actuator location $\boldsymbol{p}_b^{ba_j} = [l_{x,j},\ l_{y,j}]^\top$ shown for the bow tunnel thruster $\boldsymbol{T}_1$. . 31

6.1. Brunovsky canonical form of the nonlinear system (6.1) under the nonlinear control law (6.4) with new input $\boldsymbol{v}$, vectorial relative degree $\boldsymbol{r} = [r_1,\ r_2, \ldots, r_{n_y}]^\top$ and (fictitious) output $\boldsymbol{\lambda}(\boldsymbol{x})$. 40

6.2. Overview of solution approaches to OCPs. On the left, a structured overview is given. See also appendix A.4 for further insights into that topic. On the right, a Venn diagram is shown highlighting similarities. 58

6.3. Illustration of the MPC approach with the true state (solid red), estimated state (red pluses), predicted states (red crosses) with interpolated values (dashed red), applied control (solid blue), predicted inputs (blue circles) and piece-wise linear interpolation (dashed blue), prediction horizon t_{hor} and MPC sample time Δt_{mpc}. 61

7.1. Illustration of the base shape ˜ $\mathrm{V}(solidblue)andtheaffinetransformation\mathrm{V}(\boldsymbol{\eta})ofthecontrolledobject(hatchedblue)$. 64

7.2. Visualization of the underlying concepts of convex analysis, where $\mathcal{O}$ is defined as a square (red area) with outward normal vectors $\boldsymbol{a}_j$, $j = 1, \ldots, \breve{L}$ and $\breve{L} = 4$, $\mathcal{H}^-(\boldsymbol{a}_1, b_1)$ is a half-space (shaded gray area) such that $\mathcal{O} \subset \mathcal{H}^-(\boldsymbol{a}_1, b_1)$, $\mathcal{H}(\boldsymbol{a}_1, b_1)$ is the corresponding (supporting) hyperplane (dashed black line), and $\boldsymbol{s}_{\mathcal{O}}(\boldsymbol{a}_1)$ is a support point (green circle), which, in this case, is not unique. 65

7.3. Illustration of the three culling techniques. First, a frustum culling procedure is used (Fig. 7.3a). Subsequently, occlusion (shadow) culling is evaluated (Fig. 7.3b) and lastly, backface culling is applied to the scenario (Fig. 7.3c). Faces identified by any previous step are denoted using dashed gray lines while faces identified by the current step are indicated using dashed red lines. 78

8.1. Qualitative illustration of EKF and smoother quantities with *a priori* state $\hat{x}_k^-$ (plus), *a posteriori* state $\hat{x}_k^+$ (cross), smoothed state $\breve{x}_k$ (circle), and measurement y_k (filled dots), respective innovation ε_k^-, ε_k^+, $\breve{\varepsilon}_k$ (for $k = 0$), and real state (unmarked line), EKF sample time Δt_{ekf}, measurement sample time $\Delta t_y = 2\Delta t_{\mathrm{ekf}}$, and smoothing window $t_s = 2\Delta t_y$. 85

9.1. Vessel states $\boldsymbol{x}$ (left) and corresponding estimation error standard deviation P_x (right) for the lumped estimation with disturbance compensation for the scenario in Tab. 9.2. 103

9.2. Disturbances $\boldsymbol{d}$ acting on the vessel (left) and corresponding diagonal elements of the estimation error standard deviation $\sqrt{P_{\mathrm{d}}}$ (right) for the lumped estimation with disturbance compensation for the scenario in Tab. 9.2. . . . 105

9.3. Control input $\boldsymbol{\tau}_{\mathrm{c}}$ (left) and control surfaces $\boldsymbol{u}$ (right) of for the scenario in Tab. 9.2. 106

9.4. Square root of diagonal elements of the pose measurement noise covariance matrix $R_{\mathrm{c},eta}$ that is adaptively adjusted using an EWMA for the lumped estimation with disturbance compensation for the scenario in Tab. 9.2. . . . 107

9.5. Vessel states (left) and corresponding estimation error standard deviation (right) for the combined lumped and wind estimation with disturbance compensation for the scenario in Tab. 9.3. 109

9.6. Disturbances $\boldsymbol{d}$ and wind[1] $\boldsymbol{x}_{\mathrm{w}}$ acting on the vessel (left) and corresponding respective square root of diagonal elements of the estimation error covariance matrices P_{d}, P_{w} (right) for the combined lumped and wind estimation with disturbance compensation for the scenario in Tab. 9.3. 111

9.7. Control input $\boldsymbol{\tau}_{\mathrm{c}}$ (left) and control surfaces $\boldsymbol{u}$ (right) of for the combined lumped and wind estimation with disturbance compensation for the scenario in Tab. 9.3. 112

[1] Note that since the apparent wind is measured and not the true wind the measurements of the wind are not depicted in this plot to avoid confusion.

9.8. Square root of diagonal elements of the pose and wind measurement noise covariance matrix $R_{c,\eta}$ (left), $R_{c,w}$ (right), respectively, that is adaptively adjusted using an EWMA for the combined lumped and wind estimation with disturbance compensation the scenario in Tab. 9.3. 112

10.1. Illustration of the possible predicted vessel states calculated using (7.57) (black crosses, circles, and asterisks, respectively) for the first iteration with view frustum, optimal path points (small black dots), and optimal vessel positions (green circles) for the first MPC iteration. 118

10.2. Path following performance in the north-east-plane (above) and corresponding cross track error (below) for the scenario summarized in Tab. 10.2. 120

10.3. Time evolution of path parameter (left) and rudder angle (right) for the scenario summarized in Tab. 10.2 for the indicator approach. 121

10.4. Snapshot of MPC iteration $i = 85$ (left) and $i = 45$ (right) for the indicator simulation with predicted controlled vessel positions (green circles), radar radius (dark blue), reference path points (bold black dots), and obstacles (dark red) with obstacles' past path (light blue), predicted positions (black dots), and future path (dashed light blue). 122

10.5. Performance of the culling procedure for the indicator approach. 123

10.6. Vessel states $\boldsymbol{x}$ (left) and corresponding estimation error standard deviation P (right) for the predictive path following with full body indicator obstacle constraints for the scenario in Tab. 10.2. 124

A.1. Example solution of the hanging chain problem with initial guess (dashed black), `fmincon`'s solution (orange), where $c(\boldsymbol{x}^*) = -0.4346$, proposed algorithm's solution (blue) with $c(\boldsymbol{x}^*) = -0.4456$, and the inequality constraints modeling the floor (shaded gray). 134

A.2. Solutions to the energy optimal swing-up problem of the inverted pendulum on a cart for t_f = 1s (left) and t_f = 3s (right) with `fmincon`'s solution (dashed orange), proposed algorithm's solution (blue), final values $\boldsymbol{x}_f$ (violet), and `bvp4c`'s solution (dash-dotted black). 137

A.3. Lagrange multipliers for the defect constraints using the proposed algorithm (blue) and `bvp4c`'s costate variables (black) for t_f = 1s (left) and t_f = 3s (right). 138

A.4. Disambiguation of and mechanisms for numerical OCP methods. 142

List of Tables

2.1. Classification of obstacle representations, related methods and example applications for collision avoidance OCPs. 8

5.1. Coefficient domains and typical relations w.r.t. fundamental vessel parameters m, x_g, J_{zz}, i.e., with constant forward surge speed u_0 according to [1, Chap. 4]. 26

7.1. Summary of dual conditions/constraints of all three approaches for evaluating collision avoidance between $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$ such that[2] $\mathcal{V}(\boldsymbol{\eta}) \cap \mathcal{O} = \emptyset$. 67

9.1. Parameters of the Cybership II MSV in (unscaled) SI units according to Lindegaard [97] and Skjetne [142]. Note that there's two propeller rudder systems, one on each side of the ship. 94

9.2. Scenario parameters for the lumped disturbance estimation using the Cybership model with a nonlinear feedback controller and an adaptive EKF. All units are SI. All covariance matrices are given in their square root form, i.e., they are represented using their respective standard deviations and, thus, preserve the unit of the underlying quantity. 97

9.3. Scenario parameters for the combined lumped and wind disturbance estimation using the Cybership II model with a nonlinear feedback controller and an adaptive EKF. All units are SI. All covariance matrices are given in their square root form, i.e., they are represented using their respective standard deviations and, thus, preserve the unit of the underlying quantity. Controller and control allocation parameters are identical to the lumped estimation scenario described in Tab. 9.2. 99

10.1. Parameters of the Mariner Class MSV in dimensionless units according to [29]. Note that there's only one rudder to steer the ship. 114

10.2. Scenario parameters for the path following scenario the Mariner model with a model predictive controller and an EKF. All units are SI. All covariance matrices are given in their square root form, i.e., they are represented using their respective standard deviations and, thus, preserve the unit of the underlying quantity. 116

[2] Note that for the signed distance, this depends on d_{safe}. If $d_{\text{safe}} < 0$, overlapping of $\mathcal{V}(\boldsymbol{\eta})$ and $\mathcal{O}$ is considered as not colliding.

10.3. Comparison of average (taken over all iterations where obstacles are present) number of dual decision variables associated with the controlled vessel $\bar{n}_{w_\lambda}$ and with the obstacles $\bar{n}_{w_\mu}$ and number of obstacles per MPC iteration $\bar{K}$ before and after the culling procedure for the indicator constraint simulation. 121

References

[1] M. A. Abkowitz. Ship Hydrodynamics - Steering and Manoeuvrability. *Hydro- and Aerodynamics Laboratory, Hydrodynamics Section, Lyngby, Denmark, Report No. Hy-5, Lectures*, 1964.

[2] Jürgen Adamy. *Nichtlineare Systeme Und Regelungen*. Springer Vieweg, Berlin, 3rd edition, 2014. ISBN 978-3-662-55684-9.

[3] Sunil K. Agrawal and Heberrt Sira-Ramirez. *Differentially Flat Systems*. Number 6 in Automation and Control Engineering. CRC Press, Boca Baton, Florida, 1st edition, 2004.

[4] Aulia Siti Aisjah. An Analysis Nomoto Gain and Norbin Parameter on Ship Turning Maneuver. *IPTEK The Journal for Technology and Science*, 21(2), 2010. ISSN 0853-4098.

[5] Joel A. E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. CasADi: A Software Framework for Nonlinear Optimization and Optimal Control. *Mathematical Programming Computation*, 2018. ISSN 1867-2949.

[6] B. R. Andrievsky and I. B. Furtat. Disturbance Observers: Methods and Applications. I. Methods. *Automation and Remote Control*, 81(9):1563–1610, 2020. ISSN 0005-1179.

[7] B. R. Andrievsky and I. B. Furtat. Disturbance Observers: Methods and Applications. II. Applications. *Automation and Remote Control*, 81(10):1775–1818, 2020. ISSN 0005-1179.

[8] Marco Tulio Angulo, Andrea Aparicio, and Claude H. Moog. Structural Accessibility and Structural Observability of Nonlinear Networked Systems. *IEEE Trans. Netw. Sci. Eng.*, 7(3):1656–1666, 2020. ISSN 2327-4697, 2334-329X.

[9] Jens G. Balchen, Nils A. Jenssen, Eldar Mathisen, and Steinar Sælid. A Dynamic Positioning System Based on Kalman Filtering and Optimal Control. In *Modeling, Identification and Control: A Norwegian Research Bulletin*, volume 1, pages 135–163. Norwgian Society of Automatic Control, 1980. ISBN 1890-1328.

[10] A. Banaszuk and J. Hauser. Feedback Linearization of Transverse Dynamics for Periodic Orbits. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 2, pages 1639–1644, Lake Buena Vista, FL, USA, 1994. IEEE. ISBN 978-0-7803-1968-4.

[11] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-Multisensor Tracking: Principles and Techniques.* Yaakov Bar-Shalom, Storrs, Conn, 1995. ISBN 978-0-9648312-0-9.

[12] Yaakov Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, 2005. ISBN 9780471416555.

[13] Bradley M. Bell and Fredrick W. Cathey. The Iterated Kalman Filter Update as a Gauss-Newton Method. *IEEE Trans. Autom. Control.*, 38:294–297, 1993.

[14] Daniele Bertin, Sergio Bittanti, Silvano Meroni, and Sergio M. Savaresi. Dynamic Positioning of a "Single-Thruster" Vessel by Feedback Linearization. *IFAC Proceedings Volumes*, 33(21):275–280, 2000. ISSN 1474-6670.

[15] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming.* Society for Industrial and Applied Mathematics, 2nd edition, 2010. ISBN 978-0-89871-688-7. doi: 10.1137/1.9780898718577.

[16] Glenn Bitar, Bjørn-Olav H. Eriksen, Anastasios M. Lekkas, and Morten Breivik. Energy-Optimized Hybrid Collision Avoidance for ASVs. *2019 18th European Control Conference, ECC 2019*, pages 2522–2529, 2019.

[17] Martin Böck and Andreas Kugi. Real-time Nonlinear Model Predictive Path-Following Control of a Laboratory Tower Crane. *IEEE Transactions on Control Systems Technology*, 22(4):1461–1473, 2014. ISSN 1558-0865.

[18] Marc Bodson. Evaluation of Optimization Methods for Control Allocation. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 25(4), 2001.

[19] J. F. Bonnans, editor. *Numerical Optimization: Theoretical and Practical Aspects.* Universitext. Springer, Berlin, 2nd edition, 2006. ISBN 978-3-540-35445-1.

[20] Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004. ISBN 978-0-521-83378-3.

[21] Anke Brandner. *Optimale Allokation Für Voith-Schneider Propeller Im Rahmen Des Dynamischen Positionierens.* PhD thesis, Universität Ulm, 2014.

[22] Morten Breivik and Thor I. Fossen. Motion Control Concepts for Trajectory Tracking of Fully Actuated Ships. *7th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC)*, 2006.

[23] Pavol Brunovsky. A Classification of Linear Controllable Systems. *Kybernetika*, 6: 173–188, 1970.

[24] Richard H. Byrd, Jorge Nocedal, and Richard A. Waltz. Knitro: An Integrated Package for Nonlinear Optimization. In G. Di Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, pages 35–59. Springer US, Boston, MA, 2006. ISBN 978-0-387-30065-8.

[25] Walter Caharija, Kristin Ytterstad Pettersen, and Jan Tommy Gravdahl. Path Following of Underactuated Surface Vessels in Presence of Unknown Constant Environmental Forces: Preliminary Results. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 9(PART 1), 2013. ISSN 14746670.

[26] Sandra Carrillo and Juan Contreras. Obtaining First and Second Order Nomoto Models of a Fluvial Support Patrol using Identification Techniques. *Ciencia y tecnología de buques*, 11(22):19, 2018. ISSN 1909-8642.

[27] G. Caviglia and A. Morro. Kirchhoff's Equations for the Rigid Body Motion Revisited. *Meccanica*, 52(6):1485–1489, 2017. ISSN 15729648.

[28] Benoit Chachuat. Nonlinear and Dynamic Optimization. Lecture Notes IC-32 Spring Term, École Polytechnique Fédérale de Lausanne, 2009.

[29] M. S. Chislett and J. Strom-Tejsen. Planar Motion Mechanism Tests and Full- Scale Steering and Manoeuvring Predictions for a Mariner Class Vessel. *International Shipbuilding Progress*, 12(129):201–224, 1965.

[30] A. R Conn, Nicholas I. M Gould, Ph. L Toint, and Society for Industrial and Applied Mathematics. *Trust-Region Methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, USA, 2000. ISBN 978-0-89871-985-7.

[31] Satyan Coorg and Seth Teller. Real-Time Occlusion Culling for Models with Large Occluders. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pages 83–ff. ACM Press, 1997. ISBN 0-89791-884-3.

[32] Swarup Das, Pradeep Mishra, and S. K. Panigrah. Ships Steering Autopilot Design By Nomoto Model. *International Journal on Mechanical Engineering and Robotics Research (IJMERR)*, (3):37–41, 2015.

[33] Kenneth S. M. Davidson and Leonard I. Schiff. Turning and Course-Keeping Qualities. Technical Report 98, TU Delft, 1946.

[34] Fang Deng, Hua Lin Yang, and Long Jin Wang. Adaptive Unscented Kalman Filter Based Estimation and Filtering for Dynamic Positioning with Model Uncertainties. *International Journal of Control, Automation and Systems*, 17(3):667–678, 2019. ISSN 20054092.

[35] Fang Deng, Hua Lin Yang, Long Jin Wang, and Wei Min Yang. UKF Based Nonlinear Offset-free Model Predictive Control for Ship Dynamic Positioning under Stochastic Disturbances. *International Journal of Control, Automation and Systems*, 17(12):3079–3090, 2019. ISSN 20054092.

[36] David Eberly. *3D Game Engine Design*. CRC Press, 2nd edition, 2006. ISBN 978-1-4822-6730-3.

[37] Bjørn-Olav H. Eriksen, Glenn Bitar, Morten Breivik, and Anastasios M. Lekkas. Hybrid Collision Avoidance for ASVs Compliant With COLREGs Rules 8 and 13-17. *Frontiers in Robotics and AI*, 7:1–18, 2020.

[38] Julius Farkas. Theorie Der Einfachen Ungleichungen. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, (124):1–27, 1902.

[39] T. Faulwasser, B. Kern, and R. Findeisen. Model Predictive Path-Following for Constrained Nonlinear Systems. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference*, pages 8642–8647, Shanghai, China, 2009. IEEE. ISBN 978-1-4244-3871-6.

[40] Timm Faulwasser and Rolf Findeisen. Nonlinear Model Predictive Path-Following Control. In Manfred Morari, Manfred Thoma, Lalo Magni, Davide Martino Raimondo, and Frank Allgöwer, editors, *Nonlinear Model Predictive Control*, volume 384, pages 335–343. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-01093-4 978-3-642-01094-1.

[41] Timm Faulwasser and Rolf Findeisen. Constrained Output Path-Following for Nonlinear Systems Using Predictive Control. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 43(14):753–758, 2010. ISSN 14746670.

[42] Timm Faulwasser and Rolf Findeisen. Nonlinear Model Predictive Control for Constrained Output Path Following. *IEEE Transactions on Automatic Control*, 61(4): 1026–1039, 2016. ISSN 0018-9286.

[43] Timm Faulwasser, Veit Hagenmeyer, and Rolf Findeisen. Optimal Exact Path-Following for Constrained Differentially Flat Systems. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 18(PART 1):9875–9880, 2011. ISSN 14746670.

[44] K. K. Fedayevsky and G. K. Sobolev. Control and Stability in Ship Design. *State Union Shipbuilding Publishing House*, 1963.

[45] Roya Firoozi, Laura Ferranti, Xiaojing Zhang, Sebastian Nejadnik, and Francesco Borrelli. A Distributed Multi-Robot Coordination Algorithm for Navigation in Tight Environments. *arXiv*, abs/2006.11492, 2020.

[46] Fernando A. C. C. Fontes. A General Framework to Design Stabilizing Nonlinear Model Predictive Controllers. *Systems & Control Letters*, 42(2):127–143, 2001. ISSN 01676911.

[47] Fossen, Thor I., and Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control.* John Wiley & Sons, Ltd, Chichester, Sussex, 1st edition, 2011. ISBN 978-1-119-99413-8.

[48] Thor I. Fossen. *Guidance and Control of Ocean Vehicles.* John Wiley & Sons Ltd., Chichester, Sussex, 1st edition, 1994. ISBN 0-471-94113-1.

[49] Thor I. Fossen and A. Grovlen. Nonlinear Output Feedback Control of Dynamically Positioned Ships Using Vectorial Observer Backstepping. *IEEE Transactions on Control Systems Technology*, 6(1):121–128, 1998. ISSN 1558-0865.

[50] Thor I. Fossen and Tristan Perez. Kalman Filtering for Positioning and Heading Control of Ships and Offshore Rigs. *IEEE Control Systems Magazine*, 29(6):32–46, 2009. ISSN 1941-000X.

[51] Thor I. Fossen and Jann Peter Strand. Passive Nonlinear Observer Design for Ships Using Lyapunov Methods: Full-Scale Experiments with a Supply Vessel. *Automatica*, 35(1):3–16, 1999. ISSN 00051098.

[52] Thor I. Fossen, S. I. Sagatun, and A. J. Sørensen. Identification of Dynamically Positioned Ships. *Control Engineering Practice*, 4(3):369–376, 1996. ISSN 09670661.

[53] Thor I. Fossen, Morten Breivik, and Roger Skjetne. Line-of-Sight Path Following of Underactuated Marine Craft. *IFAC Proceedings Volumes*, 36(21):211–216, 2003. ISSN 1474-6670.

[54] Cory Tyler Fraser. *Adaptive Extended Kalman Filtering Strategies for Autonomous Relative Navigation of Formation Flying Spacecraft.* Master's thesis, Carleton University, 2018.

[55] J. J. Gehrt, Rene Zweigel, T. Konrad, and Dirk Abel. DVL-aided Navigation Filter for Maritime Applications. *IFAC-PapersOnLine*, 51(29):418–423, 2018. ISSN 24058963.

[56] Arthur Gelb. *Applied Optimal Estimation.* MIT Press, 1974. ISBN 0-262-20027-9.

[57] Matthias Gerdts, René Henrion, Dietmar Hömberg, and Chantal Landry. Path Planning and Collision Avoidance for Robots. *Numerical Algebra, Control and Optimization*, 2(3):437–463, 2012. ISSN 21553289.

[58] Reza Ghaemi, Soryeok Oh, and Jing Sun. Path Following of a Model Ship Using Model Predictive Control with Experimental Verification. In *Proceedings of the 2010 American Control Conference*, pages 5236–5241, 2010.

[59] E. Gilbert and D. Johnson. Distance Functions and Their Application to Robot Path Planning in the Presence of Obstacles. *IEEE Journal on Robotics and Automation*, 1 (1):21–30, 1985. ISSN 0882-4967.

[60] E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A Fast Procedure for Computing the Distance between Complex Objects in Three-Dimensional Space. *IEEE Journal on Robotics and Automation*, 4(2):193–203, 1988. ISSN 08824967.

[61] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using Matlab.* John Wiley & Sons, 4th edition, 2015. ISBN 978-1-118-85121-0.

[62] Ignacio Grossmann. Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques. *Optimization and Engineering*, 3(3):227–252, 2002. ISSN 1389-4420.

[63] Rebeca Grynspan. Review of Maritime Transport. Technical Report United Nations Conference on Trade and Development, United Nations, Geneva, 2022.

[64] Hubert Hahn. *Rigid Body Dynamics of Mechanisms.* Springer, Berlin, Heidelberg, 2002. ISBN 978-3-642-07617-6 978-3-662-04831-3.

[65] E. Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations.* Number 31 in Springer Series in Computational Mathematics. Springer, Berlin, New York, 2nd edition, 2006. ISBN 978-3-540-30663-4.

[66] Ola Härkegård. Efficient Active Set Algorithms for Solving Constrained Least Squares Problems in Aircraft Control Allocation. *Proceedings of the IEEE Conference on Decision and Control*, 2(December):1295–1300, 2002. ISSN 01912216.

[67] Ola Härkegård. *Backstepping and Control Allocation with Applications to Flight Control.* PhD thesis, Linköping University, 2003.

[68] Ola Härkegård and S. Torkel Glad. Resolving Actuator Redundancy - Optimal Control vs. Control Allocation. *Automatica*, 41(1):137–144, 2005. ISSN 00051098.

[69] Richard F. Hartl, Suresh P. Sethi, and Raymond G. Vickson. A Survey of the Maximum Principles for Optimal Control Problems with State Constraints. *SIAM Review*, 37(2):181–218, 1995. ISSN 0036-1445.

[70] J B Hiriart-Urruty and C Lemaréchal. *Convex Analysis and Minimization Algorithms I: Fundamentals*, volume 2nd of *Comprehensive Studies in Mathematics.* Springer, Berlin, Heidelberg, 1996. ISBN 978-3-642-08161-3.

[71] Maria Höffmann. *Risikoanalyse bei Schiffsmanövern unter dem Einfluss von Wind.* Master's thesis, Universität Bremen, 2020.

[72] Xujiang Huang, Abebe Geletu, and Pu Li. A Soft-Constrained Optimization Approach for Model Predictive Control of Multi-Agent Systems. *IFAC-PapersOnLine*, 28(25):16–21, 2015. ISSN 24058963.

[73] T. Hudson, D. Manocha, J. Cohen, M. Lin, K. Hoff, and H. Zhang. Accelerated Occlusion Culling Using Shadow Frusta. In *Proceedings of the Thirteenth Annual Symposium on Computational Geometry*, pages 1–10. ACM Press, 1997. ISBN 0-89791-878-9.

[74] John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, and Kurt Akeley. *Computer Graphics - Principles and Practice.* Addison-Wesley, 2014. ISBN 978-0-321-39952-6.

[75] IMO. Convention on the International Regulations for Preventing Collisions at Sea (COLREGs). https://www.imo.org/en/About/Conventions/Pages/COLREG.aspx, 1972. Online; accessed 10-September-2023.

[76] R. M. Isherwood. Wind Resistance of Merchant Ships. *Royal Institution of Naval Architects (RINA) Transactions*, 115(16), 1973.

[77] Alberto Isidori. *Nonlinear Control Systems.* Communications and Control Engineering. Springer, London, 1995. ISBN 978-1-4471-3909-6.

[78] Tor A. Johansen and Thor I. Fossen. Control Allocation - A Survey. *Automatica*, 49 (5):1087–1103, 2013. ISSN 00051098.

[79] Tor A. Johansen, Thomas P. Fuglseth, Petter Tøndel, and Thor I. Fossen. Optimal Constrained Control Allocation in Marine Surface Vessels with Rudders. *Control Engineering Practice*, 16(4):457–464, 2008. ISSN 09670661.

[80] Rudolf Emil Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35, 1960. ISSN 00219223.

[81] Ahmed Khelassi, Didier Theilliol, and Philippe Weber. Control Design for Over-Actuated Systems Based on Reliability Indicators. In *UKACC International Conference on Control 2010*, volume 2010, pages 1–6, 2010.

[82] Jacek Kierzenka and Lawrence F. Shampine. A BVP solver based on residual control and the Maltab PSE. *ACM Trans. Math. Softw.*, 27(3):299–316, September 2001. ISSN 0098-3500.

[83] Leticia Mayumi Kinjo, Stefan Wirtensohn, Johannes Reuter, Tomas Menard, and Olivier Gehan. Trajectory Tracking of a Fully-actuated Surface Vessel Using Nonlinear Model Predictive Control. *IFAC-PapersOnLine*, 54(16):51–56, 2021. ISSN 24058963.

[84] Gustav Kirchhoff. Ueber die bewegung eines rotationskörpers in einer flüssigkeit. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1870(71):237–262, 1870.

[85] Donald E. Kirk. *Optimal Control Theory: An Introduction.* Dover Publications, Mineola, New York, 1st edition, 2004. ISBN 0-486-43484-2.

[86] A. I. Korotkin. *Added Masses of Ship Structures*, volume 88 of *Fluid Mechanics and Its Applications.* Springer, United Kingdom, 2009. ISBN 978-1-4020-9431-6.

[87] Martin Kosch, Ahmed Elkhashap, Philipp Koschorrek, Rene Zweigel, and Dirk Abel. Hardware-in-the-Loop Trajectory Tracking and Collision Avoidance of Automated Inland Vessels Using Model Predictive Control. In *2021 European Control Conference (ECC)*, pages 2251–2256, Delft, Netherlands, 2021. IEEE. ISBN 978-94-6384-236-5.

[88] Philipp Koschorrek and Martin Kosch. An Approach to QP-based Thrust Allocation considering Inflow. *IFAC-PapersOnLine*, 54(16):126–131, 2021. ISSN 24058963.

[89] Philipp Koschorrek, Tobias Hahn, and Torsten Jeinsch. A Thrust Allocation Algorithm considering Dynamic Positioning and Roll Damping Thrust Demands using multi-step Quadratic Programming *. *IFAC-PapersOnLine*, 51(29):438–443, 2018. ISSN 24058963.

[90] Philipp Koschorrek, Martin Kosch, Maximilian Nitsch, Dirk Abel, and Dirk Jürgens. Towards Semi-Autonomous Operation of an over-Actuated River Ferry. *at - Automatisierungstechnik*, 70(5):433–443, 2022. ISSN 2196-677X, 0178-2312.

[91] Alberto Lastra. *Parametric Geometry of Curves and Surfaces: Architectural Form-Finding*, volume 5 of *Mathematics and the Built Environment.* Birkhäuser, Basel, 1st edition, 2021. ISBN 978-3-540-61545-3.

[92] Anastasios M. Lekkas, Ann L. Roald, and Morten Breivik. Online Path Planning for Surface Vehicles Exposed to Unknown Ocean Currents Using Pseudospectral Optimal Control. *IFAC-PapersOnLine*, 49(23):1–7, 2016. ISSN 24058963.

[93] Edward W. Lewandowski. *The Dynamics of Marine Craft.* World Scientific Publishing Co. Pte Ltd., 2004. ISBN 981-02-4755-9.

[94] Edward V. Lewis. *Principles of Naval Architecture Vol. III: Motions in Waves and Controllability.* The Society of Naval Architects and Marine Engineers, Jersey City, New Jersey, 1990. ISBN 0-939773-02-3.

[95] Frank L. Lewis, Lihua Xie, and Dan Popa. *Optimal and Robust Estimation: With An Introduction to Stochastic Control Theory.* Number 26 in Automation and Control Engineering. CRC Press, 2nd edition, 2008. ISBN 978-1-4200-0829-6.

[96] Shihua Li, Jun Yang, Wen-Hua Chen, and Xisong Chen. *Disturbance Observer-Based Control - Methods and Applications.* CRC Press, 2014. ISBN 978-1-4665-1580-2.

[97] Karl-Petter W. Lindegaard. *Acceleration Feedback in Dynamic Positioning.* PhD thesis, Norwegian University of Science and Technology (NTNU), 2003.

[98] Patrick Lindemann. The Gilbert-Johnson-Keerthi Distance Algorithm. Technical Report Proseminar Medieninformatik Summer Term, University of Munich, Munich, 2009.

[99] David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*, volume 8 of *International Series in Operations Research & Management Science.* Springer, Heidelberg, 4th edition, 2016. ISBN 978-3-319-18841-6.

[100] Jan Lunze. *Regelungstechnik 1.* Springer, Berlin, Heidelberg, 11th edition, 2016. ISBN 978-3-662-52677-4.

[101] Max Lutz and Thomas Meurer. Efficient Formulation of Collision Avoidance Constraints in Optimization Based Trajectory Planning and Control. *IEEE Conference on Control Technology and Application (CCTA)*, 2021.

[102] Jan Maciejowski and Eric C. Kerrigan. Soft Constraints And Exact Penalty Functions In Model Predictive Control. *Computer Aided Control System Design*, (September 2000):221–247, 2000.

[103] Reiner Marchthaler and Sebastian Dingler. *Kalman-Filter - Einführung in Die Zustandsschätzung Und Ihre Anwendung Für Eingebettete Systeme.* Springer, Berlin, Heidelberg, 1st edition, 2017. ISBN 978-3-658-16727-1.

[104] Andreas B. Martinsen, Anastasios M. Lekkas, and Sebastien Gros. Autonomous Docking Using Direct Optimal Control. *IFAC-PapersOnLine*, 52(21):97–102, 2019. ISSN 24058963.

[105] Enrico Massa and Stefano Vignolo. Newton-Euler Lagrange and Kirchhoff Formulations of Rigid Body Dynamics - a Unified Approach. *Meccanica*, 51(8):2019–2023, 2016. ISSN 15729648.

[106] Peter S. Maybeck. *Stochastic Models, Estimation and Control Vol. II*, volume 141–2 of *Mathematics in Science and Engineering*. Academic Press, New York, 1st edition, 1982. ISBN 0-12-480702-X.

[107] D. Q. Mayne, James B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained Model Predictive Control: Stability and Optimality. *Automatica*, 36(6):789–814, 2000. ISSN 00051098.

[108] Raman K. Mehra. Approaches to Adaptive Filtering. *IEEE Transactions on Automatic Control*, 17(5):693–698, 1972. ISSN 15582523.

[109] L.M. Milne-Thomson. *Theoretical Hydrodynamics.* MacMillan Education, London, 5th edition, 1968. ISBN 978-1-349-00519-2.

[110] A. H. Mohamed and K. P. Schwarz. Adaptive Kalman Filtering for INS/GPS. *Journal of Geodesy*, 73(4):193–203, 1999. ISSN 0949-7714.

[111] Mattia Montanari, Nik Petrinic, and Ettore Barbieri. Improving the GJK Algorithm for Faster and More Reliable Distance Queries Between Convex Objects. *ACM Transactions on Graphics*, 36(3):1–17, 2017. ISSN 0730-0301.

[112] Lúcia Moreira, Thor I. Fossen, and C. Guedes Soares. Path Following Control System for a Tanker Ship Model. *Ocean Engineering*, 34(14-15):2074–2085, 2007. ISSN 00298018.

[113] Steven Nahmias and Tava Lennon Olsen. *Production and Operations Analysis.* Waveland Press, 7th edition, 2015.

[114] Chris Nielsen and Manfredi Maggiore. Maneuver Regulation via Transverse Feedback Linearization: Theory and Examples. *IFAC Proceedings Volumes*, 37(13):57–64, 2004. ISSN 1474-6670.

[115] Christopher Nielsen and Manfredi Maggiore. On Local Transverse Feedback Linearization. *SIAM J. Control Optim.*, 47(5):2227–2250, 2008. ISSN 0363-0129, 1095-7138.

[116] Christopher Nielsen, Cameron Fulford, and Manfredi Maggiore. Path Following Using Transverse Feedback Linearization: Application to a Maglev Positioning System. *Automatica*, 46(3):585–590, 2010. ISSN 0005-1098.

[117] Henk Nijmeijer and Arjan van der Schaft. *Nonlinear Dynamical Control Systems*. Springer, New York, 1st edition, 1990. ISBN 978-0-387-97234-3.

[118] Maximilian Nitsch, Jan-Jöran Gehrt, Rene Zweigel, and Dirk Abel. Tightly Coupled INS/GNSS Navigation Filter for the Automation of a River Ferry. *IFAC-PapersOnLine*, 54(16):139–145, 2021. ISSN 24058963.

[119] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006. ISBN 978-0-387-30303-1.

[120] K. Nomoto, T. Taguchi, K. Honda, and S. Hirano. On the Steering Qualities of Ships. *Interantional Shipbuilding Progress*, 4(35):354–370, 1957. ISSN 15662829, 0020868X.

[121] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer, New York, 1st edition, 2003. ISBN 0-387-95482-1.

[122] Claudio Paliotta, Erjen Lefeber, and Kristin Ytterstad Pettersen. Trajectory Tracking of Under-Actuated Marine Vehicles. *2016 IEEE 55th Conference on Decision and Control, CDC 2016*, pages 5660–5667, 2016.

[123] Claudio Paliotta, Erjen Lefeber, Kristin Ytterstad Pettersen, Jose Pinto, Maria Costa, and Joao Tasso de Figueiredo Borgesde de Sousa. Trajectory Tracking and Path Following for Underactuated Marine Vehicles. *IEEE Transactions on Control Systems Technology*, 27(4):1423–1437, 2019. ISSN 1063-6536.

[124] Rushen B. Patel and Paul J. Goulart. Trajectory Generation for Aircraft Avoidance Maneuvers Using Online Optimization. *Journal of Guidance, Control, and Dynamics*, 34(1):218–230, 2011.

[125] Les Piegl and Wayne Tiller. *The NURBS Book*. Monographs in Visual Communications. Springer, Berlin, Heidelberg, 2nd edition, 2013. ISBN 978-3-540-61545-3.

[126] Ivan Popov, Philipp Koschorrek, Adel Haghani, and Torsten Jeinsch. Adaptive Kalman Filtering for Dynamic Positioning of Marine Vessels. *IFAC-PapersOnLine*, 50(1):1121–1126, 2017. ISSN 24058963.

[127] M. J. D. Powell. Nonlinear Programming—Sequential Unconstrained Minimization Techniques. *The Computer Journal*, 12(3):207–a, 1969. ISSN 0010-4620.

[128] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum Likelihood Estimates of Linear Dynamic Systems. *AIAA Journal*, 3(8):1445–1450, 1965. ISSN 0001-1452, 1533-385X.

[129] A Ricci. A Constuctive Geometry for Computer Graphics. *The Computer Journal*, 16(2):157–160, 1972.

[130] Arthur Richards and J.P. How. Aircraft Trajectory Planning with Collision Avoidance Using Mixed Integer Linear Programming. In *Proceedings of the 2002 American Control Conference (ACC)*, volume 3, pages 1936–1941, 2002. ISBN 0-7803-7298-0.

[131] Peter Ridley, Julien Fontan, and Peter Corke. Submarine Dynamic Modeling. In *Proceedings of the Australian Conference on Robotics and Automation*, pages 1–8, Brisbane, Australia, 2003.

[132] Andrew Ross. *Nonlinear Manoeuvring Models for Ships: A Lagrangian Approach*. PhD thesis, Norwegian University of Science and Technology (NTNU), 2008.

[133] John S. Sargent and Paul N. Cowgill. Design Considerations for Dynamically Positioned Utility Vessels. In *Offshore Technology Conference*, Houston, Texas, 1976.

[134] Tobias Schoels, Per Rutquist, Luigi Palmieri, Andrea Zanelli, Kai O. Arras, and Moritz Diehl. CIAO*: MPC-based Safe Motion Planning in Predictable Dynamic Environments. *arXiv*, abs/2001.05449, 2020. ISSN 23318422.

[135] Elmar Schömer, Joachim Reichel, and Thomas Warken. Efficient Collision Detection for Curved Solid Objects. In *ACM Symposium on Solid Modeling and Applications*, pages 321–328, New York, 2002. ACM Press. ISBN 978-1-58113-506-0.

[136] John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. Motion Planning with Sequential Convex Optimization and Convex Collision Checking. *The International Journal of Robotics Research*, 33(9):1251–1270, 2014. ISSN 0278-3649.

[137] Pierre O. M. Scokaert and James B. Rawlings. Feasibility Issues in Linear Model Predictive Control. *AIChE Journal*, 45(8):1649–1659, 1999. ISSN 00011541.

[138] Do Thanh Sen and Tran Canh Vinh. Determination of Added Mass and Inertia Moment of Marine Ships Moving in 6 Degrees of Freedom. *International Journal of Transportation Engineering and Technology*, 2(1):8, 2016. ISSN 2575-1751.

[139] J. A. Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. *Proceedings of the National Academy of Sciences of the United States of America*, 93(4):1591–1595, 1996. ISSN 00278424.

[140] C. Shi, D. Zhao, J. Peng, and C. Shen. Identification of Ship Maneuvering Model Using Extended Kalman Filters. *Marine Navigation and Safety of Sea Transportation*, 3(1):329–334, 2009. ISSN 2083-6473.

[141] Dan Simon. *Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches*. John Wiley & Sons, New York, 1st edition, 2006. ISBN 978-0-471-70858-5.

[142] Roger Skjetne. *The Maneuvering Problem*. PhD thesis, Norwegian University of Science and Technology (NTNU), 2005.

[143] Roger Skjetne, Øyvind Smogeli, and Thor I. Fossen. Modeling, Identification, and Adaptive Maneuvering of CyberShip II: A Complete Design with Experiments. *IFAC Proceedings Volumes*, 37(10):203–208, 2004. ISSN 14746670.

[144] Raffaele Soloperto, Johannes Kohler, Frank Allgöwer, and Matthias A. Muller. Collision Avoidance for Uncertain Nonlinear Systems with Moving Obstacles Using Robust Model Predictive Control. In *18th European Control Conference (ECC)*, pages 811–817, 2019. ISBN 978-3-907144-00-8.

[145] Asgeir J. Sørensen. A Survey of Dynamic Positioning Control Systems. *Annual Reviews in Control*, 35(1):123–136, 2011. ISSN 13675788.

[146] Barbara Stepien. Yara Birkeland and the Legal Problems of Autonomous Shipping. In *Boletim Da Sociedade Brasileira De Direito Internacional*, volume 104, Nr. 131-135, pages 403–427, 2018.

[147] Serge Sutulo and C. Guedes Soares. Mathematical Models for Simulation of Manoeuvring Performance of Ships. *Marine Technology and Engineering*, 1(January): 661–698, 2011.

[148] David Titterton and John Weston. *Strapdown Inertial Navigation Technology*. Institution of Electrical Engineers, 2nd edition, 2004. ISBN 978-0-86341-358-2.

[149] Michael S. Triantafyllou and Franz S. Hover. Maneuvering and Control of Marine Vehicles. Lecture Notes, Massachussets Institute of Technology (MIT), 2003.

[150] Gino van den Bergen. A Fast and Robust GJK Implementation for Collision Detection of Convex Objects. *Journal of Graphics Tools*, 4(2):7–25, 1999. ISSN 1086-7651.

[151] Gino van den Bergen. Proximity Queries and Penetration Depth Computation on 3d Game Objects. *Game Developers Conference (GDC)*, pages 1–17, 2001.

[152] Rudolph Van Der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, Oregon Health & Science University, 2004.

[153] Andreas Wächter. *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. PhD thesis, Carnegie Mellon University, 2002.

[154] R. A. Waltz, J. L. Morales, Jorge Nocedal, and D. Orban. An Interior Algorithm for Nonlinear Optimization That Combines Line Search and Trust Region Steps. *Mathematical Programming*, 107(3):391–408, 2006. ISSN 00255610.

[155] Eric A. Wan and Rudolph Van Der Merwe. *Kalman Filtering and Neural Networks*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, New York, 1st edition, 2001. ISBN 978-0-471-22154-6.

[156] Fang Wang, Ming Lv, and Feng Xu. Design and Implementation of a Triple-Redundant Dynamic Positioning Control System for Deepwater Drilling Rigs. *Applied Ocean Research*, 57:140–151, 2016. ISSN 0141-1187.

[157] Stefan Wirtensohn, Hanna Wenzl, Thomas Tietz, and Johannes Reuter. Parameter Identification and Validation Analysis for a Small USV. In *2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 701–706, Miedzyzdroje, Poland, 2015. IEEE. ISBN 978-1-4799-8701-6.

[158] Stefan Wirtensohn, Michael Schuster, and Johannes Reuter. Disturbance Estimation and Wave Filtering Using an Unscented Kalman Filter. *IFAC-PapersOnLine*, 49(23): 518–523, 2016. ISSN 24058963.

[159] A.D. Wnęk, A. Paço, X.-Q. Zhou, S. Sutulo, and C. Guedes Soares. Experimental Study of Aerodynamic Loads on an LNG Carrier and Floating Platform. *Applied Ocean Research*, 51:309–319, 2015. ISSN 01411187.

[160] Xiaojing Zhang, Alexander Liniger, Atsushi Sakai, and Francesco Borrelli. Autonomous Parking Using Optimization-Based Collision Avoidance. In *2018 IEEE Conference on Decision and Control (CDC)*, volume 2018, pages 4327–4332, 2018. ISBN 978-1-5386-1395-5.

[161] Xiaojing Zhang, Alexander Liniger, and Francesco Borrelli. Optimization-Based Collision Avoidance. *IEEE Transactions on Control Systems Technology*, 29(3):972–983, 2021. ISSN 1063-6536.

Supervised Theses

[162] Kristina Apelt. Entwurf und Implementierung einer modellprädiktiven Regelung von autonomen Schiffen unter Berücksichtigung der Kollsionsverhütungsregeln. Master's thesis. Kiel University. Suprvised by Simon Helling; Thomas Meurer, 2021.

[163] Sönke Bartels. Optimization-based docking maneuvers of autonomous surface vessel. Master's thesis. Kiel University. Suprvised by Simon Helling; Thomas Meurer, 2021.

[164] Torben Kannengiesser. Störgrößenschätzung für maritime Systeme. Bachelor's thesis. Kiel University. Suprvised by Simon Helling; Thomas Meurer, 2022.

[165] Lucas Pöhler. Optimierungsbasierte und nichtlineare Methoden zur Lösung des dynamischen Positionierens für autonome Schiffe. Master's thesis. Kiel University. Suprvised by Simon Helling; Thomas Meurer, 2021.

[166] Lukas Schirmer. Entwurf und Analyse von Pfadfolgeregelungsmethoden linearer und nichtlinearer Schiffsmodelle. Bachelor's thesis. Kiel University. Suprvised by Simon Helling; Thomas Meurer, 2020.

[167] Mariusz Supka. Lernbasierte Regelung von autonomen Schiffen mithilfe von Reinforcement Learning. Master's thesis. Kiel University. Suprvised by Simon Helling; Thomas Meurer, 2022.

Own Publications

Journal Publications

[168] Simon Helling and Thomas Meurer. Dual Collision Detection in Model Predictive Control Including Culling Techniques. *IEEE Transactions on Control Systems Technology*, 31(6):2449–2464, 2023.

[169] Simon Helling, Max Lutz, and Thomas Meurer. A combined guidance and control concept for autonomous ferries. *Automatisierungstechnik*, 70(5):444–455, 2022.

Conference Publications

[170] Sönke Bartels, Simon Helling, and Thomas Meurer. Rope-Assisted Docking Maneuvers for Autonomous Surface Vessels. In *American Control Conference (ACC)*, pages 2315–2320, Atlanta, USA, 2022. IEEE.

[171] Sönke Bartels, Simon Helling, and Thomas Meurer. Inequality Constrained Optimal Control for Rope-Assisted ASV Docking Maneuvers. In *IFAC-PapersOnLine*, volume 55 (31), pages 44–49, Kongens Lyngby, Denmark, 2022. IFAC.

[172] Simon Helling and Thomas Meurer. A culling procedure for collision avoidance model predictive control with application to ship autopilot models. In *IFAC-PapersOnLine*, volume 54 (16), pages 43–50, Oldenburg, Germany, 2021. IFAC.

[173] Simon Helling and Thomas Meurer. The Rotary Flexible Joint – A Holistic View from Modeling and Control to Observer Design and Experiments. In *IFAC-PapersOnLine*, volume 56 (2), pages 7567–7572, Yokohama, Japan, 2023. IFAC.

[174] Simon Helling, Max Lutz, and Thomas Meurer. Flatness-based MPC for underactuated surface vessels in confined areas. In *IFAC-PapersOnLine*, volume 53 (2), pages 14686–14691, Berlin, Germany, 2020. IFAC.

[175] Simon Helling, Christian Roduner, and Thomas Meurer. On the Dual Implementation of Collision-Avoidance Constraints in Path-Following MPC for Underactuated Surface Vessels. In *American Control Conference (ACC)*, pages 3366–3371, New Orleans, USA, 2021. IEEE.

Fortschrittsberichte aus der Regelungstechnik
und Prozessautomatisierung

Band 01

Herausgeber: Prof. Dr.-Ing. habil. Thomas Meurer

In this work, the modeling, optimal and nonlinear control of marine surface vessels is addressed along with collision avoidance, state and disturbance estimation.
The nonlinear dynamical vessel model is presented, which forms the basis for subsequent analysis of nonlinear and optimization-based control designs. A nonlinear feedback linearization controller is presented that is combined with a control allocation, which optimally distributes the desired control forces to actuators. The control allocation problem constitutes a constrained static optimization problem. In this context, an interior-point SQP line search method is derived. Additionally, a predictive controller is proposed that is able to handle state and input constraints. The former are used to achieve collision avoidance. To this end, a dual approach in combination with culling techniques is presented that is especially suited for confined environments.
The two control designs are combined with a Kalman Filter-based state and disturbance estimator to solve a dynamic positioning and path following problem, respectively, and are illustrated using high fidelity simulations.

ISSN 2943-0887