Trajectory optimization based on recursive B-spline approximation for automated longitudinal control of a battery electric vehicle

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)
von der KIT-Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)

angenommene
Dissertation
von

Jens Jauch, M.Sc.

Tag der mündlichen Prüfung:	11. Juli 2023
Hauptreferent:	Prof. Dr. rer. nat. Frank Gauterin
Korreferent:	Prof. Dr.-Ing. Eric Sax

Abstract

This work describes a novel method for approximating an unbounded number of data points using a B-spline function in the linear and nonlinear weighted least squares sense. The developed method is based on iterative algorithms for state estimation and its computational effort increases linearly with the number of data points. The method allows to shift the bounded definition range of a B-spline function during run-time such that the latest data point can be considered for approximation regardless of the initially chosen definition range. Furthermore, the shift operation allows to decrease the sizes of matrices in the state estimators in order to reduce computational effort in both offline applications, in which all data points are available at once for processing, and online applications, in which additional data points are observed in each time step.

The trajectory optimization problem is formulated such that the approximation method computes a B-spline function that represents the desired velocity trajectory with respect to time using data points created from map data. The computational effort of the resulting direct trajectory optimization method increases only linearly with the unbounded temporal length of the planned trajectory. The combination with an adaptive model that describes the power train properties of a battery electric vehicle with fixed gear box transmission ratio allows to optimize velocity trajectories with respect to travel time, comfort and energy consumption.

The trajectory optimization method is extended to a driver assistance system for automated vehicle longitudinal control that is tested in simulations as well as in real test drives. Simulated drives on a chosen reference route need up to 3.4 % less energy with the automated longitudinal control than with a human driver at the same average velocity. For the same energy consumption the automated longitudinal control achieves a 2.6 % higher average velocity than a human driver.

Kurzfassung

Diese Arbeit beschreibt ein neuartiges Verfahren zur linearen und nichtlinearen gewichteten Kleinste-Quadrate-Approximation einer unbeschränkten Anzahl von Datenpunkten mit einer B-Spline-Funktion. Das entwickelte Verfahren basiert auf iterativen Algorithmen zur Zustandsschätzung und sein Rechenaufwand nimmt linear mit der Anzahl der Datenpunkte zu. Das Verfahren ermöglicht eine Verschiebung des beschränkten Definitionsbereichs einer B-Spline-Funktion zur Laufzeit, sodass der aktuell betrachtete Datenpunkt ungeachtet des anfangs gewählten Definitionsbereichs bei der Approximation berücksichtigt werden kann. Zudem ermöglicht die Verschiebeoperation die Reduktion der Größen der Matrizen in den Zustandsschätzern zur Senkung des Rechenaufwands sowohl in Offline-Anwendungen, in denen alle Datenpunkte gleichzeitig zur Verarbeitung vorliegen, als auch in Online-Anwendungen, in denen in jedem Zeitschritt weitere Datenpunkte beobachtet werden.

Das Trajektorienoptimierungsproblem wird so formuliert, dass das Approximationsverfahren mit Datenpunkten aus Kartendaten eine B-Spline-Funktion berechnet, die die gewünschte Geschwindigkeitstrajektorie bezüglich der Zeit repräsentiert. Der Rechenaufwand des resultierenden direkten Trajektorienoptimierungsverfahrens steigt lediglich linear mit der unbeschränkten zeitlichen Trajektorienlänge an. Die Kombination mit einem adaptiven Modell des Antriebsstrangs eines batterie-elektrischen Fahrzeugs mit festem Getriebeübersetzungsverhältnis ermöglicht die Optimierung von Geschwindigkeitstrajektorien hinsichtlich Fahrzeit, Komfort und Energieverbrauch.

Das Trajektorienoptimierungsverfahren wird zu einem Fahrerassistenzsystem für die automatisierte Fahrzeuglängsführung erweitert, das simulativ und in realen Erprobungsfahrten getestet wird. Simulierte Fahrten auf der gewählten Referenzstrecke benötigten bis zu $3,4\ \%$ weniger Energie mit der automatisierten Längsführung als mit einem menschlichen Fahrer bei derselben Durchschnittsgeschwindigkeit. Für denselben Energieverbrauch erzielt die automatisierte Längsführung eine $2,6\ \%$ höhere Durchschnittsgeschwindigkeit als ein menschlicher Fahrer.

Acknowledgment

This dissertation was created within the research project *e-volution* funded by the German Federal Ministry of Education and Research during my time as research associate and doctoral candidate at the Institute of Vehicle System Technology (FAST) at Karlsruhe Institute of Technology (KIT).

First, I would like to thank my adviser Professor Dr. rer. nat. Frank Gauterin for giving me the opportunity to work on this research topic and for his support as well as scientific guidance regarding research project, publications and dissertation. Furthermore, I would like to thank Professor Dr.-Ing. Eric Sax for his comprehensive feedback to this work and for being my second adviser. Moreover, I am grateful to Professor Dr.-Ing. Tobias Düser for chairing the examination committee and to my research group manager Dr.-Ing. Michael Frey for his support, suggestions and reviews.

My colleagues I would like to thank for creating an inspiring atmosphere rich of scientific exchange. In particular previous works by Felix Bleimund, Dr.-Ing. Dominik Dörr and Björn Fath facilitated the orientation of my research towards algorithms for splines and trajectories. Additionally, I gratefully acknowledge the efforts of Dr.-Ing. Johannes Masino and Dr.-Ing. Stephan Rhode for joint publications.

Within *e-volution*, I also enjoyed collaborating with Dr.-Ing. Moritz Vaillant and Volker Watteroth, associates at Dr. Ing. h.c. F. Porsche AG and Porsche Engineering Services GmbH, respectively. Their project contribution enabled the application of the developed algorithms in a real vehicle.

The students Changjoon Lee, Tim Staiger and Adam Thorgeirsson contributed to my dissertation via *e-volution*-related work and also to a mutual publication.

Special thanks go to my sister for detailed reviews of this dissertation. Lastly, I am particularly grateful to my parents, to whom I dedicate this dissertation, for their continuous support and encouragement.

Karlsruhe, July 2023 *Jens Jauch*

Contents

Acronyms

AC	air conditioning
ACC	adaptive cruise control
AEPM	adaptive electrical power model
ALC	automated longitudinal control
ASM	asynchronous motor
ATFM	adaptive traction force model
BEV	battery electric vehicle
CAN	Controller Area Network
CC	cruise control
COG	center of gravity
DM	direct methods
DP	Dynamic Programming
ECU	electronic control unit
EKF	extended Kalman filter
EM	electric motor
FA	front axle
FB-KRLS	Fixed-Budget KRLS
GB	gear box
HEV	hybrid electric vehicle
HiL	Hardware-in-the-Loop
HQC	Hannan-Quinn information criterion
HV	high voltage
IDM	Intelligent Driver Model
IM	indirect methods
KAF	kernel adaptive filter
KF	Kalman filter
KRLS	kernel RLS
KRLS-T	KRLS Tracker
LiNMC	lithium nickel-manganese-cobalt oxide
LM	Levenberg-Marquardt
LS	least squares
MiL	Model-in-the-Loop
MLC	manual longitudinal control
MPC	model predictive control
MPF	marginalized particle filter
MPP	most probable path
NRBA	nonlinear recursive B-spline approximation
NRMSE	normalized root mean square error
NWLS	nonlinear weighted least squares
OBN	on-board network
OCP	optimal control problem
OR	optimization range
PDF	probability density function
PE	power electronics
PF	particle filter
PFD	partial fraction decomposition
PID	proportional-integral-derivative
PP	pressure point
PSM	permanently excited synchronous motor
PTC	positive temperature coefficient
PVOP	possible vehicle operating points
QP	quadratic programming
RA	rear axle
RBA	recursive B-spline approximation
RLS	recursive least squares
SCKF	square-root cubature Kalman filter
SiL	Software-in-the-Loop
SOC	state of charge
SQP	sequential quadratic programming
SSE	sum of squared errors
UKF	unscented Kalman filter
ViL	Vehicle-in-the-Loop
WLS	weighted least squares
WR	Weissach route

Symbols

Optimal control

$\mathcal{J}$	optimization function.
$\boldsymbol{\mathcal{F}}$	state constraint matrix.
$\boldsymbol{\mathcal{G}}$	input constraint matrix.
δt	cycle time.
T	time horizon.

Approximation

$\mathbf{0}$	zero matrix.
$\mathbf{1}$	matrix of ones.
$\boldsymbol{\mathcal{A}}^L$	state transition matrix for linear states.
$\boldsymbol{\mathcal{A}}^N$	state transition matrix for nonlinear states.
$\boldsymbol{\mathcal{A}}$	state transition matrix.
$\boldsymbol{\mathcal{B}}$	input matrix.
$\boldsymbol{B}$	B-spline matrix.
$\boldsymbol{b}$	B-spline vector.
b	B-spline.
$\boldsymbol{C}$	measurement matrix for KF, RBA, MPF and NRBA.
$\mathbf{C}$	measurement matrix for WLS.
c	nonlinear measurement function.
$\mathcal{D}$	definition range.
$\boldsymbol{B}'$	derivative of B-spline matrix.
d	degree.
δ	degree index.
i	iteration.
δ	correction step size.
$\boldsymbol{\lambda}$	damping parameter.
$\mathbf{e}$	error function.
$\boldsymbol{e}$	measurement error.
$\boldsymbol{\phi}$	function whose values depend nonlinearly on its control points.
ϵ	tolerance.
f	B-spline function.
$\boldsymbol{I}$	identity matrix.
I	number of spline intervals.
J	number of B-splines.
j	B-spline index.
$\boldsymbol{\mathcal{K}}$	Kalman gain.
K	number of knots.
k	knot index.
$\boldsymbol{\kappa}$	knot vector.
κ	knot.
L	linear quantity.
μ	spline interval index.
P	number of data points.
N	nonlinear quantity.
$\mathcal{O}$	upper asymptotic border.
$\boldsymbol{\mathcal{P}}^-$	covariance matrix of a priori estimation error.
$\boldsymbol{\mathcal{P}}^{L,-}$	covariance matrix of a priori estimation error of linear states.
$\boldsymbol{\mathcal{P}}^+$	covariance matrix of a posteriori estimation error.
$\boldsymbol{\mathcal{P}}^{L,+}$	covariance matrix of a posteriori estimation error.
$\boldsymbol{\mathcal{P}}^{L\xi,-}$	cross-covariance matrix of a priori estimation error.
$\bar{p}$	initial value in $\boldsymbol{\mathcal{P}}_0^+$ and $\boldsymbol{Q}$.
$\boldsymbol{\mathcal{P}}^{\xi,-}$	covariance matrix of a priori estimation error of nonlinear states.
$\boldsymbol{\mathcal{P}}^{\xi L,-}$	cross-covariance matrix of a priori estimation error.
$\boldsymbol{P}$	particle count.
p	particle index.
$\boldsymbol{Q}$	covariance matrix of process noise.
$\boldsymbol{Q}^L$	covariance matrix of process noise for linear states.
$\bar{q}$	initial value in $\boldsymbol{Q}$.
$\boldsymbol{Q}^N$	covariance matrix of process noise for nonlinear states.

q	particle importance weight.
$\tilde{q}$	normalized particle importance weight.
$\bar{q}^L$	initial value in $\boldsymbol{Q}^L$.
$\bar{q}^N$	initial value in $\boldsymbol{Q}^N$.
r	order of derivative.
$\boldsymbol{\mathcal{R}}$	covariance matrix of measurement noise for KF, RBA, MPF and NRBA.
$\mathbf{R}$	covariance matrix of measurement noise for WLS and LM.
rnd	random value drawn from the standard normal distribution.
$\boldsymbol{s}$	vector of independent variables.
s	independent variable.
σ	shift variable.
p	time step.
$\hat{\boldsymbol{u}}$	optimal control sequence.
$\boldsymbol{u}^L$	input signal vector for linear states.
$\boldsymbol{u}^N$	input signal vector for nonlinear states.
$\boldsymbol{u}$	input signal vector.
V	number of measurements.
v	index of measurement.
$\boldsymbol{\upsilon}$	random vector of measurement noise.
$\boldsymbol{\omega}^L$	random vector of process noise for linear states.
$\boldsymbol{\omega}$	random vector of process noise.
$\boldsymbol{\omega}^N$	random vector of process noise for nonlinear states.
$\bar{\boldsymbol{\omega}}^N$	random vector of process noise for particle filter states.
$\boldsymbol{x}$	state vector.
$\boldsymbol{x}^L$	linear state vector.
$\boldsymbol{x}^N$	nonlinear state vector.
$\hat{\boldsymbol{x}}$	state estimate.
$\hat{\boldsymbol{x}}^-$	a priori state estimate.
$\hat{\boldsymbol{x}}^+$	a posteriori state estimate.
$\bar{x}$	initial value for $\hat{x}$.
$\hat{\boldsymbol{x}}^L$	a priori state estimate.
$\hat{\boldsymbol{x}}^{L,-}$	linear a priori state estimate.
$\hat{\boldsymbol{x}}^{L,-,p}$	linear a priori particle.
$\hat{\boldsymbol{x}}^{L,+,p}$	linear a posteriori particle.
$\hat{\boldsymbol{x}}^{N,-,p}$	nonlinear a priori particle.
x	component of system state.
$\hat{x}$	component of system state estimate.
$\bar{x}^{\mathbf{Init}}$	initial value for $\hat{x}$ during initialization.
$\boldsymbol{\xi}$	linear effects of nonlinear state.
$\hat{\boldsymbol{\xi}}$	linear effects of nonlinear state estimate.
$\hat{\boldsymbol{\xi}}^p$	linear effects of nonlinear particle.
$\hat{y}$	function value.
y	measurement.
$\boldsymbol{y}$	vector of measurements.

Vehicle models

A	(m^2) effective vehicle cross-sectional area perpendicular to the vehicle longitudinal axis.
α	(rad) road slope angle.
$a_{\mathbf{x}}$	(m s^{-2}) sensor longitudinal acceleration.
$\dot{v}_{\mathbf{Vhcl,y}}$	(m s^{-2}) lateral acceleration.
$\dot{v}_{\mathbf{Vhcl,y,max}}$	(m s^{-2}) maximum absolute value of lateral acceleration.
c	(N rad^{-1}) tire cornering stiffness.
$\boldsymbol{c}_{\mathbf{Vhcl}}$	vehicle motion vector.
$c_{\mathbf{w}}$	longitudinal drag coefficient for $\tau_{\mathrm{rel}} = 0$.
$c_{\mathbf{x}}$	longitudinal drag coefficient.
$E_{\mathbf{Batt}}$	(W s) HV battery energy.
$E_{\mathbf{Loss}}$	(W s) energy loss between HV battery and wheel.
F_0	(N) rolling resistance constant.
$F_{\mathbf{air}}$	(N) air resistance.
$F_{\mathbf{air,x}}$	(N) longitudinal air resistance.
$F_{\mathbf{air,y}}$	(N) lateral air resistance.
$F_{\mathbf{centr}}$	(N) centrifugal force.
$F_{\mathbf{cl}}$	(N) climbing force.
$F_{\mathbf{cor}}$	cornering resistance.
$F_{\mathbf{inert}}$	(N) inertial force.
$f_{\mathbf{r}}$	rolling resistance coefficient.
$F_{\mathbf{roll}}$	(N) rolling resistance.
$F_{\mathbf{trac}}$	(N) traction force.
$F_{\mathbf{Whl,z,FA}}$	(N) vertical front axle wheel force.
$F_{\mathbf{Whl,z,RA}}$	(N) vertical rear axle wheel force.
g	gravitational constant.

I	(A) battery current.
i_{G}	gear ratio.
J_{Brake}	($\mathrm{kg\,m^2}$) mass moment of inertia of brake.
J_{EM}	($\mathrm{kg\,m^2}$) mass moment of inertia of electric motor.
J_{GB}	($\mathrm{kg\,m^2}$) mass moment of inertia of gear box.
J_{Wheel}	($\mathrm{kg\,m^2}$) mass moment of inertia of wheel.
l	(m) wheel base.
$l_{\mathbf{FA}}$	(m) distance between FA and COG.
$l_{\mathbf{RA}}$	(m) distance between RA and COG.
M	number of model parameters.
m_{eq}	(kg) equivalent mass.
$m_{\mathbf{vhcl}}$	(kg) vehicle mass.
$P_{\mathbf{AC}}$	(W) AC power consumption.
$P_{\mathbf{AEPM}}$	(W) output of adaptive electrical power model.
$P_{\mathbf{Batt}}$	(W) HV battery power.
$P_{\mathbf{Batt,dem}}$	(W) HV battery power demand.
$P_{\mathbf{Batt,Loss}}$	(W) HV battery power loss.
$P_{\mathbf{DC/DC}}$	(W) DC/DC converter power consumption.
$P_{\mathbf{grid}}$	(W) grid charging power.
$P_{\mathbf{Loss}}$	(W) power loss between HV battery and wheels.
$P_{\mathbf{PTC}}$	(W) PTC power consumption.
$P_{\mathbf{trac,elec}}$	(W) electrical traction power.
$P_{\mathbf{trac,mech}}$	(W) mechanical traction power.
$P_{\mathbf{wheel}}$	(W) wheel power.
r_{dyn}	(m) dynamic wheel radius.
$r_{\mathrm{dyn},\mathbf{FA}}$	(m) dynamic front wheel radius.
$r_{\mathrm{dyn},\mathbf{RA}}$	(m) dynamic rear wheel radius.
R^-	(Ω) discharge resistance.
R^+	(Ω) charge resistance.
T	($\mathrm{N\,m}$) torque.
$T_{\mathbf{Brk,FA}}$	($\mathrm{N\,m}$) front axle brake torque.
$T_{\mathbf{Brk,GB,FA}}$	($\mathrm{N\,m}$) front axle gear box brake torque.
$T_{\mathbf{Brk,GB,RA}}$	($\mathrm{N\,m}$) rear axle gear box brake torque.
$T_{\mathbf{Brk,RA}}$	($\mathrm{N\,m}$) rear axle brake torque.
$T_{\mathbf{des}}$	($\mathrm{N\,m}$) controller torque demand.
$T_{\mathbf{des,MPC}}$	($\mathrm{N\,m}$) MPC torque demand.
$T_{\mathbf{des,PC}}$	($\mathrm{N\,m}$) pilot control torque demand.
$T_{\mathbf{EM}}$	($\mathrm{N\,m}$) motor torque.
$T_{\mathbf{EM,FA}}$	($\mathrm{N\,m}$) front axle motor torque.
$T_{\mathbf{EM,RA}}$	($\mathrm{N\,m}$) rear axle motor torque.
$T_{\mathbf{FA}}$	($\mathrm{N\,m}$) front axle torque.
$T_{\mathbf{RA}}$	($\mathrm{N\,m}$) rear axle torque.
$t_{\mathbf{Trip}}$	(s) trip time.
$U_{\mathbf{OCV}}$	(V) open-circuit voltage.
$U_{\mathbf{TV}}$	(V) terminal voltage.
$v_{\mathbf{rel}}$	($\mathrm{m\,s^{-1}}$) relative air velocity.
$v_{\mathbf{Vhcl}}$	($\mathrm{m\,s^{-1}}$) vehicle velocity.
$\dot{v}_{\mathbf{Vhcl,x}}$	($\mathrm{m\,s^{-2}}$) vehicle longitudinal acceleration.
$\ddot{v}_{\mathbf{Vhcl,x}}$	($\mathrm{m\,s^{-3}}$) vehicle longitudinal jerk.
$v_{\mathbf{Wind}}$	($\mathrm{m\,s^{-1}}$) environmental wind velocity.
$\boldsymbol{x}_{\mathbf{Vhcl}}$	vehicle parameter vector.
γ	(%) road slope.
κ	(1/m) road curvature.
$\omega_{\mathbf{EM}}$	($\mathrm{rad\,s^{-1}}$) motor angular velocity.
$\omega_{\mathbf{EM,FA}}$	($\mathrm{rad\,s^{-1}}$) front axle motor angular velocity.
$\omega_{\mathbf{EM,RA}}$	($\mathrm{rad\,s^{-1}}$) rear axle motor angular velocity.
$\omega_{\mathbf{Whl,FA}}$	($\mathrm{rad\,s^{-1}}$) front wheel angular velocity.
$\omega_{\mathbf{Whl,RA}}$	($\mathrm{rad\,s^{-1}}$) rear wheel angular velocity.
R	(Ω) internal ohmic resistance.
ρ	($\mathrm{kg\,m^{-3}}$) air density.
σ^2	kernel variance.
$\tau_{\mathbf{rel}}$	(rad) relative air flow angle.
$\tau_{\mathbf{wind}}$	(rad) environmental wind angle.

Trajectory planning and longitudinal control

$a_{\mathbf{des}}$	($\mathrm{m\,s^{-2}}$) desired acceleration.
$a_{\mathbf{free}}$	($\mathrm{m\,s^{-2}}$) free-road acceleration term.
$a_{\mathbf{IDM}}$	($\mathrm{m\,s^{-2}}$) desired acceleration of IDM.
$a_{\mathbf{int}}$	($\mathrm{m\,s^{-2}}$) interaction acceleration term.
$a_{\mathbf{des,neg}}$	($\mathrm{m\,s^{-2}}$) desired negative longitudinal acceleration.

$a_{\text{des,pos}}$	(m s^{-2}) desired positive longitudinal acceleration.
a_{Set}	(m s^{-2}) set point of trajectory acceleration.
a_{TJY}	(m s^{-2}) trajectory acceleration.
j_{TJY}	(m s^{-3}) trajectory jerk.
R_a	variance of acceleration measurement.
R_a^{-1}	weight of acceleration error square.
R_j	variance of jerk measurement.
R_j^{-1}	weight of jerk error square.
R_P	variance of power measurement.
R_P^{-1}	weight of power error square.
R_v	variance of velocity measurement.
R_v^{-1}	weight of velocity error square.
$R_{\text{v<vLim}}$	error weighting if v_{TJY} below $v_{\text{Lim,TJY}}$.
$R_{\text{v>vLim}}$	error weighting if v_{TJY} above $v_{\text{Lim,TJY}}$.
s	(m) position.
s_{TJY}	(m) trajectory position.
t	(s) time.
v	(m s^{-1}) velocity.
v_{des}	(m s^{-1}) desired velocity.
$v_{\text{Lim,Crest}}$	(m s^{-1}) speed limit resulting from crests.
$v_{\text{Lim,Curve}}$	(m s^{-1}) speed limit resulting from curvature.
$v_{\text{Lim,Law}}$	(m s^{-1}) legal speed limit.
$v_{\text{Lim,Map}}$	(m s^{-1}) speed limit resulting from map data.
$v_{\text{Lim,Map},\dot{v}}$	(m s^{-1}) speed limit from map data with desired acceleration.
$v_{\text{Lim,Speedo}}$	(m s^{-1}) legal speed limit including speedometer offset.
$v_{\text{Lim,TJY}}$	(m s^{-1}) speed limit for trajectory optimization.
v_{Set}	(m s^{-1}) set point of trajectory velocity.
v_{TJY}	(m s^{-1}) trajectory velocity.
$v_{\text{VhclAhead}}$	(m s^{-1}) velocity of the vehicle ahead.
Δs_{des}	(m) desired minimum distance.
Δs_{Jam}	(m) desired traffic jam distance.
$\Delta s_{\text{Sighting}}$	(m) sighting distance.
$\Delta s_{\text{VhclAhead}}$	(m) distance to the vehicle ahead.
Δt_{It}	(s) temporal distance of neighboring data points.
Δt_{κ}	(s) temporal distance of neighboring knots.
$\Delta t_{\text{Lim,TJY}}$	(s) temporal safety margin to upper speed limit.
$\Delta t_{\text{VhclAdapt}}$	(s) temporal distance of adaption to vehicle state.
$\Delta t_{\text{VhclAhead}}$	(s) desired time gap to the vehicle ahead.
δ	acceleration exponent.

1 Introduction

1.1 Background

The topic of this work falls into the context of driver assistance systems for automated vehicle longitudinal control based on map data. Such systems can increase safety, improve comfort and reduce the energy consumption of the vehicle by computing an energy-efficient course of velocity for the road section ahead.

In the recent past, energy savings have mainly been achieved by power train improvements such as downsizing of the internal combustion engine as well as hybrid and purely electric power trains. When taking into account the increasing costs for further power train improvements, assistance systems for energy-efficient longitudinal control offer a more cost-efficient way to obtain additional energy savings. According to [117, p. 141], efficient driving has the second largest potential for energy savings (30 %) after electrification of the power train (75 %).

Driver assistance systems for automated longitudinal control (ALC) compute the desired course of velocity, also called velocity trajectory, by solving a trajectory optimization problem. Different optimization approaches are known, e.g. Dynamic Programming (DP) and direct methods (DM).

The popular DP approach finds the global optimum for a given optimization problem and its computational effort grows linearly with the temporal length of the planned trajectory. However, only coarsely discretized problems with few state dimensions can be solved in real-time on a series electronic control unit (ECU) with limited resources.

A possible way is to compute a rough long-term trajectory with DP that serves as a reference trajectory for a DM. In addition to discretizing the optimization problem, DM approximate the system state and control between the discretization steps by a function with respect to time. DM are, however, only suitable for short-term trajectories since the effort of DM increases exponentially with the temporal trajectory length.

This work addresses the need for computationally less demanding trajectory optimization algorithms for ALC of a battery electric vehicle (BEV).

1.2 Approach

The research problem is approached by developing a novel method for the generic task of approximating an unbounded number of data points using a B-spline function in the linear and nonlinear weighted least squares (WLS) sense. The developed method is based on iterative algorithms for state estimation.

The approximation problem is reformulated as a trajectory optimization problem such that the approximation method computes a velocity trajectory with respect to time using data points created from map data. The novel trajectory optimization method falls into the category of DM and its effort increases only linearly with the trajectory length instead of exponentially. The combination with an adaptive model that describes the power train properties of the BEV allows to plan velocity trajectories whose resulting energy consumption varies depending on the chosen relative weighting of different target criteria.

The trajectory optimization is extented to a driver assistance system for ALC that is tested in simulation as well as in real test drives. In simulations on a chosen reference route the ALC is compared to a recorded and re-simulated real drive with manual longitudinal control (MLC) regarding energy consumption and average velocity.

1.3 Outline

Figure 1.1 illustrates the structure of this work, which follows the V model. The V model describes the software development process and distinguishes different process levels. On the descending branch of the V model, the development goal is analysed and iteratively broken down into subgoals. Thereby requirements are specified for subsystems that ultimately need to be implemented.

Chapter 2 of the work corresponds to the descending branch of the V model and will provide literature background as well as identify research gaps. The development goal on system level is a driver assistance system for energy-efficient ALC of a BEV (Section 2.1) which is broken down into a trajectory optimization method on component level (Section 2.2) and B-spline approximation methods on algorithm level (Section 2.3). Section 2.4 gives an introductive overview of adaptive filters for Chapter 3 and Chapter 5 without identifying a research gap.

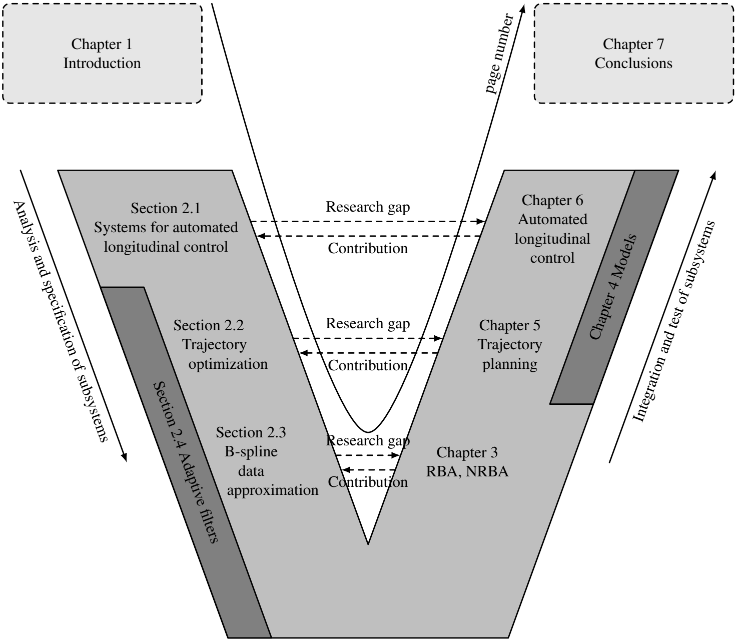


Figure 1.1: Structure of this work illustrated by the V model that is used to describe the software development process.

The ascending branch of the V model includes testing and iterative integration of subcomponents into larger systems until the main goal is reached. Chapter 3 to Chapter 6 address this phase by presenting contributions for closing the research gaps on each level of the V model.

Chapter 3 presents novel algorithms for iterative linear and nonlinear WLS approximation of data with a B-spline function. The algorithms are referred to as recursive B-spline approximation (RBA) and nonlinear recursive B-spline approximation (NRBA), respectively.

Chapter 4 provides neccessary foundations for Chapter 5 and Chapter 6. Without having a direct counterpart to the left branch, it states models of the vehicle that are used for determining energy-efficient trajectories and for evaluating the energy consumption of the research vehicle in simulations on a chosen reference route.

Chapter 5 applies RBA and NRBA to the trajectory optimization problem and presents extensions for taking into account trajectory constraints.

Chapter 6 describes the development process of the ALC and its system architecture including the interaction of its components. Furthermore, Chapter 6 investigates the energy-saving potential with the ALC on the reference route in comparison to a recorded and re-simulated real drive with MLC as well as effects of ALC parameters on the energy consumption that results with the ALC.

Chapter 7 concludes the journey through the V model, summarizes the contributions to research gaps and gives suggestions for future research.

1.4 Works, contributions and support by others

The research described in this work took place within the research project *e-volution* [41] funded by the German Federal Ministry of Education and Research (support code 16EMO0071). The Dr. Ing. h.c. F. Porsche AG was among the research partners and supported the project work by providing a battery electric research vehicle for test drives as well as vehicle data and measurement data. Furthermore, an employee of the Porsche Engineering Services GmbH integrated the developed ALC system into the prototyping ECU of the research vehicle and conducted test drives.

The provided research vehicle was developed during the preceding research project *e-generation* [22], which was also funded by the German Federal Ministry of Education and Research (support code 16N 11865) and realized in cooperation with the above mentioned companies. The author of this work did not contribute to *e-generation* but reused the following results:

Despite differing strongly in detail, the B-spline data approximation algorithms stated in Chapter 3 adopt the iterative, filter-based characteristics of the previous method for polynomial data approximation [22, pp. 29-35], which was patented by F. Bleimund and S. Rhode [21]. A detailed differentiation between the approaches is given at the end of Chapter 3.

The adaptive traction force model (ATFM) stated in Subsection 4.7.1 stems from F. Bleimund [22, pp. 24-29] and is applied without modifications. The idea to model the power train characteristics with a kernel regression model, as done in Subsection 4.7.2 using the adaptive electrical power model (AEPM), originates from S. Rhode [146], who also provided a first script in MATLAB that performs this task. Based on this, both authors continued the investigations independently from each other. Problem-specific adaptions for using the model for trajectory optimization as well as the determination of model hyperparameters were done by the author of this work.

Trajectory optimization within the previous ALC [22, pp. 29-35] developed by F. Bleimund differs significantly from the one described in this work, although both are examples of DM. The previous ALC defined trajectories with respect to position by a cubic polynomial. The few degrees of freedom of this function only allowed to represent trajectories with simple shape over short distances. The energy that the vehicle will need for tracking the planned trajectories was not considered in the optimization process.

Therefore, regarding trajectory optimization this work only adopts the generation of an upper speed limit (Section 5.1) from *e-generation* [22, pp. 29, 30]. The adaptions made are limited to parameterization improvements. At an early stage, with non-final trajectory representation, AEPM and nonlinear filter, tests for finding suitable target criteria weightings for the trajectory optimization including the elecrical power (Section 5.4) were done by A. Thorgeirsson [173] as a student assistant.

Moreover, reused results include agreed interfaces between the ALC system and the vehicle, the technical architecture of the ALC system as well as a Hardware-in-the-Loop (HiL) test bench, which simulates the research vehicle on the reference route and runs the developed ALC on the same ECU type as in the real research vehicle. The HiL test bench (Section 6.1) was created by B. Fath [22, pp. 60-73].

The route data module (Subsection 6.2.1) was implemented by D. Dörr [22, pp. 23, 24]. Within this work, the parameterization was enhanced and logic for processing roundabout data was added. The parameter adaption module (Subsection 6.2.2) including ATFM was developed by F. Bleimund [22, pp. 24-29]. The author of this work added the AEPM.

To the controller module the author of this work added the functionality stated at the beginning of Subsection 6.2.4 and replaced the original proportional-integral-derivative (PID) control by a model predictive control (MPC). Overall architecture and control loop including the pilot control (Subsubsection 6.2.4) were adopted from F. Bleimund [22, pp. 35, 36]. First implementations and simplified tests of the MPC (Subsubsection 6.2.4) were done by C. Lee [98] during his time as student assistant after completing his Master's thesis.

First evaluations of the energy-saving potential (Section 6.3) with non-final ALC setup were conducted by A. Thorgeirsson [173] while being employed as a student assistant. For more detailed differentiations regarding the individual contributions, the reader is referred to the afore-stated parts of this work.

1.5 Prepublications and their citations

The research community benefits from fast publication of new findings. Therefore, parts of this dissertation have been prepublished. Namely, RBA and NRBA described in Chapter 3 are the topics of [78, 80] and the contents of Chapter 4, Chapter 5 and Chapter 6 are mentioned in a very condensed way in [78, 79].

This work will put the topics of these publications into a broader context and present them in much more detail. For more flexibility in rearranging content compared to a cumulative dissertation, only parts of previous publications have been added. To avoid interruptions of the reading flow because of frequent changes between paraphrases and direct quotes, all content adopted from prepublications, regardless of the extent of adaptions made, i.e. both paraphrases as well as direct quations, will be indicated as follows:

□ ... Prepublished content ... *Adopted from [prepublication].*

For reproducibility and convenient further use by other researchers the MATLAB source code for RBA, NRBA and trajectory planning has been provided along with the corresponding publication [75, 76, 77]. However, the source code of the trajectory planning method was published with a generic vehicle model instead of the parameters of the research vehicle and for the simplest planning case only. Therefore the results presented in Chapter 4 to Chapter 6 are not exactly reproducible by others.

1.6 Notation

In this work vectors are printed in bold font and matrices are printed bold in capital letters. For example, a and A are scalars, $\boldsymbol{a}$ is a vector and $\boldsymbol{A}$ is a matrix. Unless explicitly stated, the sizes of vectors and matrices are assumed to be chosen appropriately.

A hat above a variable $\boldsymbol{a}$ such as $\hat{\boldsymbol{a}}$ indicates an estimate of $\boldsymbol{a}$ or the solution of an optimization problem with respect to the variable $\boldsymbol{a}$.

This work also adopts the MATLAB-like index notation. The first index of a matrix $\boldsymbol{A}$ refers to rows, the second to columns. For example, $\boldsymbol{A}_{i,j}$ refers to the entry in row i and column j of matrix $\boldsymbol{A}$. The colon operator (:) refers to all rows or columns, or a range. $\boldsymbol{A}_{:,j}$ means all rows of column j. Analogously, $\boldsymbol{A}_{i,:}$ denotes all columns of row i. $\boldsymbol{A}_i$ is an abbreviation of $\boldsymbol{A}_{i,:}$. The colon operator in the index of $\boldsymbol{A}_{1:4,j}$ extracts a column vector of rows one to four in column j of matrix $\boldsymbol{A}$. Additionally, $\boldsymbol{a}[i]$ refers to the i-th element of vector $\boldsymbol{a}$.

$|\cdot|$ denotes the absolute value and $()^\top$ the transpose operation.

2 Scientific and technical state of the art

This chapter provides background information and identifies research gaps regarding driver assistance systems for ALC in Section 2.1, trajectory optimization in Section 2.2 and types of spline representations as well as B-spline approximation methods in Section 2.3. Furthermore, Section 2.4 gives an overview on adaptive filters for Chapter 3 and Section 4.7.

2.1 Assistance systems for automated longitudinal control

The tasks that the driver performs can be divided into three levels:

- Navigation: Choosing a travel route
- Guidance: Determining a suitable speed from the traffic environment
- Stabilization: Aligning the vehicle motion with the guidance variables

Driver assistance systems can support on each of these levels.

□ Regarding the vehicle guidance, assistance functions can be distinguished by their operating mode as depicted in Table 2.1 [187, pp. 36-40]:

- Informing and warning functions
- Continuously automated functions
- Intervening emergency functions

This work deals with continuously automated functions for longitudinal control. Such functions can contribute to increasing safety and comfort as well as to reducing energy consumption to different extents.

For example, the cruise control (CC) function controls the brake and engine torque so that the vehicle maintains the speed selected by the driver. Adaptive cruise control (ACC) is an extension of CC that uses a radar sensor to detect a vehicle ahead. ACC can reduce the selected vehicle velocity to maintain a chosen time gap to a vehicle ahead [52, 196]. Studies cited in [187, pp. 1140-1145] report that with activated ACC drivers mention feeling safer and more relaxed compared to manual driving. *Adopted from [79].* ACC-like systems increase comfort by reducing the driver workload. However, the effects on safety are not as clear as the effects on comfort and require more comprehensive investigations including psychological effects.

Table 2.1: Operating modes regarding vehicle guidance [187, p. 38, adapted]

Mode name	Informing and warning functions	Continuously automated functions	Intervening emergency functions
Mode character	A	B	C
Vehicle control	Only indirect control via the driver	• Immediate direct control • Division of driving tasks between human driver and function • Control always remains overridable	Immediate direct control in near-accident situations or situations that cannot handled by an average driver
Examples	• Display of current speed limit based on traffic sign recognition • Lane departure warning	Usually convenience functions: • Adaptive cruise control • Lane centering	Usually safety functions: Automatic emergency braking (system triggered)

These are adressed by works that are mentioned in the above cited literature. Briefly summarized, automation of tasks favors that the driver diverts the attention to things that are irrelevant for driving. In assistance systems that still require monitoring by the driver, this can cause a safety issue if the driver unexpectedly needs to take back control. For example, in one of the studies, drivers looked away from the traffic ahead for longer periods when driving with ACC [187, pp. 1140-1145].

Driving efficiency oriented extensions of ACC use a corridor of allowed distances to the vehicle ahead instead of a specific distance value given by the selected time gap. Within this corridor the vehicle is controlled according to an optimization that considers the energy consumption of the vehicle, either explicitly with an energy consumption model [18, 88, 127, 195] or implicitely using a proxy for the energy consumption, such as acceleration [1, 107, 112]. The effectiveness of both approaches is compared in [81]. Section 4.7 will address consumption models in more detail.

□ One step further go ACC enhancements that additionally determine the appropriate speed depending on map data so that the vehicle automatically slows down if a curve is ahead. Map information enables such systems to determine an even more energy-efficient or time-efficient driving strategy as Table 2.2 illustrates. *Adopted from [79].*

Further developments of assistance systems increase the degree of automation and finally lead to autonomous vehicles that do not require a driver. The degree of automation according to SAE International is depicted in Ta-

Table 2.2: Time and energy savings with automated longitudinal controls that consider map data compared to benchmark control methods. Dynamic Programming (DP), direct methods (DM), conventional vehicle (CV), hybrid electric vehicle (HEV), battery electric vehicle (BEV), urban road (U), country road (C), highway (H), artificial road profile (A).

Source	Optimization type	Vehicle type	Energy difference	Time difference	Benchmark	Road type
[114]	DP	CV	−10 %	0 %	Human	U,C
[139]	DP	CV	−23.8 %	1.5 %	Human	U,C
[139]	DP	CV	−10.2 %	−1.3 %	Human	U,C
[100]	DP	CV	−12 %	−3 %	Human	U,C
[100]	DP	CV	−8.28 %	0.01 %	Human	U,C,H
[100]	DP	CV	1.71 %	−3.60 %	Human	U,C,H
[183]	DP	HEV	−18.1 %	−1.15 %	Heuristic strategy	U,C,H
[183]	DP	HEV	0 %	−21.13 %	Heuristic strategy	U,C,H
[86]	DM	HEV	−4 %	n/a	Constant velocity	H
[188]	DP	BEV	−5.83 %	n/a	PID control	A
[188]	DM	BEV	−5.40 %	n/a	PID control	A
[188]	DM	BEV	−1.27 %	n/a	PID control	H
[188]	DM	BEV	−1.14 %	n/a	PID control	H

ble 2.3. Both mode A and mode C functions according to Table 2.1 are SAE level 0 functions because they are only capable of warnings and momentary assistance. Most ACC systems fall into the category SAE level 1, because they can perform longitudinal control in certain use cases but the driver has to monitor them, has to override them occasionally and needs to perform the lateral control by operating the steering wheel. Some system enhancements include lateral guidance [187, p. 1146]. Systems that offer both longitudinal and lateral control belong to at least SAE level 2.

□ ALC systems that determine an energy-efficient driving strategy mainly have been developed for vehicles with internal combustion engine [139] or

Table 2.3: SAE J3016 levels of driving automation [172, adapted].

Level name	No Driving Automation	Driver Assistance	Partial Driving Automation	Conditional Driving Automation	High Driving Automation	Full Driving Automation
Level number	0	1	2	3	4	5
Type	Driver support system			Automated driving system		
Vehicle control	Human is driving when the system is engaged, even if pedals and steering wheel are not touched			Human is not driving when the system is engaged, even if sitting in the driver's seat		
Monitoring and fallback solution	Human must constantly supervise the system and override it as the situation requires			On request by the system, human must take over driving	Feature will not require human to take over driving	
Capability	Provide warnings and momentary assistance	Provide EITHER longitudinal OR lateral control support	Provide longitudinal AND lateral control support	Can drive the vehicle under limited conditions as long as all required conditions are met		Can drive the vehicle under all conditions
Example	Display of current speed limit based on traffic sign recognition Lane departure warning Automatic emergency braking	EITHER lane centering OR adaptive cruise control	Lane centering AND adaptive cruise control SIMULTA-NEOUSLY	Traffic jam chauffeur	Local driverless taxi Installation of pedals and steering wheel optional	Same as 4 but system can drive everywhere in all conditions

hybrid electric vehicles [183]. The degrees of freedom of the power train in a conventional vehicle include motor torque, clutch state and selected gear. The various operating modes of a hybrid power train translate to additional degrees of freedom. Therefore an energy-efficient driving strategy is the solution to a nonlinear and high dimensional problem. To perform the required computations on a ECU with limited computational power in real-time, the referenced approaches need to reduce the problem complexity, discretize it coarsely and partly use parallelizable, iterative and approximative approaches.

Research gap: In contrast to conventional and hybrid vehicles, a BEV usually has a 1-speed gear box and therefore a constant gear ratio. If the BEV has multiple motors, as the research vehicle does, they have at least similar characteristics so that the possible benefits from including the torque distribution in the optimization problem over a rule-based torque distribution strategy that considers efficiency and driving stability are small. With a rule-based strategy, as described in Section 4.3 for the research vehicle, the only degree of freedom that is left in the power train for optimization is the total torque. The power required by auxilaries can also be significant but will not be considered in the optimization based on the rationales mentioned in Section 4.4.

The lack of ALC systems that take advantage of this resulting simple structure by incorporating a trajectory optimization approach that is on the one hand less suitable for the complexity of the power train of a conventional or hybrid vehicle but on the other hand oriented towards low computation power demand by design, such as a local optimization based on adaptive filters, an overview of which is given in Section 2.4, poses a research gap. The next section will illustrate and compare the features of different trajectory optimization approaches in more detail. *Adopted from [78, 79].*

2.2 Trajectory optimization

Driver assistance systems that perform continuously automated functions calculate input commands to steering, brakes, engine and power train in order to achieve a desired vehicle motion, also called trajectory. Trajectory planning is also known as motion planning in robotics [187, p. 1414] and differs from a path planning in that it additionally assigns a time law to a geometric path [55]. Reviews of state of the art motion planning methods for automated driving are provided in [40, 59, 90].

With increasing extent of automation and number of degrees of freedom the decision making process of driver assistance systems becomes more complex and the trajectory is found by solving an optimization problem. Trajectory optimization denotes the process of determining a trajectory of a dynamical system including the control input to the system. Thereby the trajectory must meet the system constraints and optimize a performance measure [185, p. 3]. This section gives an overview of trajectory optimization methods.

The trajectory is often optimized with respect to a trade-off between target criteria such as comfort, safety, energy comsumption and travel time. Trajectory constraints result from the vehicle dynamics, for example power restrictions, as well as from the environment, for example lanes and other vehicles [187, p. 1414].

In static optimization problems the optimal values of a finite set of variables $\boldsymbol{p}$ need to be determined such that a cost function $J(\boldsymbol{p})$ is minimized. Trajectory optimization problems fall into the category of dynamic or infinite dimensional optimization because the optimization variables are functions $x(t)$ of an independent variable t, usually time. Assessing the performance of the trajectory by a scalar quantity requires a cost functional, which is a function of a function.

The trajectory optimization problem can be mathematically formulated as an optimal control problem (OCP):

$$\begin{aligned}
&\boldsymbol{u}^*(t) = \underset{\boldsymbol{u}(t)}{\arg\min}\, J(\boldsymbol{u}(t)) \\
&J(\boldsymbol{u}(t)) = \int_0^{t_f} l\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) \mathrm{d}t + V\left(\boldsymbol{x}(t_f), t_f\right) \\
&\text{subject to} \\
&\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right),\ \boldsymbol{x}(0) = \boldsymbol{x}_0, \\
&\boldsymbol{g}\left(\boldsymbol{x}(t_f), t_f\right) = \boldsymbol{0},\ \ \boldsymbol{h}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) \leq \boldsymbol{0}\ \forall t \in [0, t_f].
\end{aligned} \tag{2.1}$$

In an OCP we seek for $t \in [0, t_f]$ the control input trajectory $\boldsymbol{u}(t) \in \mathbb{R}^m$ for a system with state $\boldsymbol{x} \in \mathbb{R}^n$ that leads the system from its initial state $\boldsymbol{x}_0$ to a terminal state $\boldsymbol{x}_{t_f}$ while minimizing the cost functional J and meeting the system dynamics model $\boldsymbol{f}$, equality constraints $\boldsymbol{g}$ and inequality constraints $\boldsymbol{h}$. $\boldsymbol{u}^*(t)$ is the optimal input trajectory and $\boldsymbol{x}_t^*$ the resulting state trajectory. J includes integral costs l and terminal costs V [187, pp. 1415-1416].

□ Figure 2.1 illustrates that known solution approaches to the trajectory optimization problem can be categorized into the three principles Dynamic Programming (DP), indirect methods (IM) and direct methods (DM). *Adopted from [78, 79].* However, some approaches combine characteristics of several principles. Therefore the assignment is not necessarily unique [131, p. 27]. Apart from simple problems and special cases, solutions are derived numerically [185, p. 9].

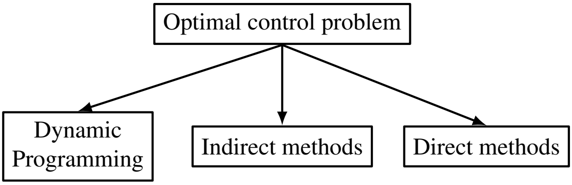


Figure 2.1: Methods for solving the optimal control problem

The following subsections describe the three principles along with application examples.

Dynamic Programming

For purely continuous systems a solution to the Hamilton-Jacobi-Bellman partial differential equation provides sufficient optimality conditions for an optimal solution of the control problem. The DP algorithm samples the continuous state and control spaces in order to achieve a discrete OCP:

$$\begin{aligned}&\boldsymbol{u}^*(t) = \underset{\boldsymbol{u}(t)}{\arg\min}\, J(\boldsymbol{u}(t)), \quad J(\boldsymbol{u}(t)) = \sum_{k=0}^{k_f-1} l\left(\boldsymbol{x}(k), \boldsymbol{u}(k), k\right)\\ &\text{subject to}\\ &\boldsymbol{x}(k+1) = \boldsymbol{f}\left(\boldsymbol{x}(k), \boldsymbol{u}(k)\right), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \quad k = 0, \ldots, k_f - 1\end{aligned} \tag{2.2}$$

The optimal control sequence $\boldsymbol{u}^*(k)$ that minimizes the sum of cost from the initial state to the final state is computed iteratively using value iteration. Equality or inequality constraints are taken into account by setting the costs $l = \infty$ for cases that violate constraints [187, pp. 1425-1430].

The algorithm takes advantage of the optimality principle of Bellman, which states that an optimal trajectory is composed of optimal subtrajectories. This principle allows to split the OCP into many smaller problems.

Since the value function depends on the system state $\boldsymbol{x}$, the optimal control is a closed feedback control loop. The closed-loop control is valid within the bounds of the sampled state space [131, p. 5].

DP is especially beneficial for trajectory planning in driving scenarios with dynamic environment and constraints resulting from other traffic participants [87] and for planning parking maneuvers [103]. □ An ACC based on DP is proposed in [53]. DP based algorithms for energy-efficient ALC exist for vehicles with internal combustion engine [100, 139], hybrid electric vehicles [183], plug-in hybrid electric vehicles [197] and BEVs [188]. *Adopted from [78, 79].*

If the time when a vehicle reaches a location is unimportant, the problem can be simplified. However, without temporal dependency the decision graph is cyclic. Therefore the value iteration needs to be replaced with another graph searching method that draws on the DP paradigm, for example the Dijkstra algorithm or the faster A^* algorithm [187, p. 1430]. In the DARPA urban challenge 2007 many successful autonomous vehicles applied an A^* based path planning algorithm [34].

Indirect methods

Indirect methods (IM) transform the OCP into the problem of solving a system of nonlinear equations. First, they determine the neccessary first-order optimality conditions for an OCP, which leads to a boundary value problem consisting of a set of differential equations, the so-called Hamilton equations. Then they solve the boundary value problem numerically, e.g by a Newton method [185, pp. 10, 11].

For example, variational calculus requires that the first derivative of the functional $J(\boldsymbol{u}(t))$ diminishes for the optimal control $\boldsymbol{u}^*(t)$. The Lagrange multiplier method allows incorporating differential equality constraints resulting from the system dynamics. Without inequality constraints the Hamilton equations then read

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t), \;\; \dot{\boldsymbol{\lambda}} = -\frac{\partial l}{\partial \boldsymbol{x}} - \left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right]^\top \boldsymbol{\lambda}, \;\; \boldsymbol{0} = -\frac{\partial l}{\partial \boldsymbol{u}} - \left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right]^\top \boldsymbol{\lambda}. \tag{2.3}$$

Herein $\boldsymbol{\lambda}$ denotes the Lagrange multipliers [187, pp. 1417-1418]. The approach is also known as maximum principle of Pontryagin [131, p. 32] or minimum principle of Pontryagin [185, p. 10], depending on the formulation of the optimization problem.

IM only state a necessary condition for optimality and considering state constraints is often difficult [127]. Furthermore, they require initial conditions for the Lagrange multipliers. This made them unsuitable for many automotive applications [187, p. 1420]. Nevertheless, there are indirect trajectory optimization applications in literature, e.g. for conventional vehicles in [126, 160] and for electric vehicles in [39, 132].

Direct methods

□ DM compute the solution of an OCP for a continuous system by discretizing it with respect to time. Between the discrete time steps they approximate the system state and control with a function of time. Thereby DM translate the infinite dimensional problem into a finite dimensional optimization problem. *Adopted from [79].* System dynamics enter the optimization problem via constraints for discrete points in time. Therefore constraints are also finite.

For example, the input trajectory can be represented by a polynomial $\boldsymbol{\psi}$ and its finite dimensional parameter vector $\bar{\boldsymbol{u}}$:

$$\boldsymbol{u}(t) = \boldsymbol{\psi}(t, \bar{\boldsymbol{u}}) \tag{2.4}$$

Then the system dynamics are

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{\psi}(t, \bar{\boldsymbol{u}}), t), \;\; \boldsymbol{x}(t_0) = \boldsymbol{x}_0. \tag{2.5}$$

This initial value problem can be solved by a differential equation solver. The system trajectory is denoted by

$$\boldsymbol{x}(t) = \boldsymbol{\phi}(t, \bar{\boldsymbol{u}}). \tag{2.6}$$

The resulting static problem that approximates the dynamic OCP reads

$$\begin{aligned}
&\boldsymbol{u}^*(t) = \arg\min_{\bar{\boldsymbol{u}}} J(\bar{\boldsymbol{u}}),\\
&J(\bar{\boldsymbol{u}}) = \int_0^{t_f} l\left(\boldsymbol{\phi}(t, \bar{\boldsymbol{u}}), \boldsymbol{\psi}(t, \bar{\boldsymbol{u}}), t\right) \mathrm{d}t + V\left(\boldsymbol{\phi}(t_f, \bar{\boldsymbol{u}})\right)\\
&\text{subject to}\\
&\boldsymbol{g}(\boldsymbol{\phi}(t_f, \bar{\boldsymbol{u}}), t_f) = \boldsymbol{0}, \;\; \boldsymbol{h}(\boldsymbol{\phi}(t_i, \bar{\boldsymbol{u}}), \boldsymbol{\psi}(t_i, \bar{\boldsymbol{u}})) \leq \boldsymbol{0}, \;\; i = 1, \ldots, N.
\end{aligned} \tag{2.7}$$

□ The cost functional J is optimized directly using an optimization method that varies the finite state and control values of the functional representation. Usually the optimization problem is nonlinear and solved using sequential quadratic programming methods (e.g. [65, 92]) or interior point methods.

DM often occur in literature as MPC, which solves (2.7) on a receding horizon [187, pp. 1420-1425,1431-1432]. *Adopted from [78].* For example, [56] applies MPC for generating locally optimal trajectories, [28]

Table 2.4: Comparison of methods for solving an optimal control problem (OCP). The table shows average values for common algorithms applied to the same OCP. Some methods may not fulfill all of the generalized statements. Summarized version derived from [131, p. 8]. *: According to [131, p. 8], an exception that achieves polynomial increase using a convexification approach is stated in [151].

Category	Dynamic Programming (DP)	Indirect methods (IM)	Direct methods (DM)
Optimality of solution	Global	Local	Local
Domain of convergence	Global	Smaller than DM	Larger than IM
Ease of initialization	Higher than DM	Lower than DM	Medium
Classes of solvable systems	Less than IM	Medium	More than IM
Control	Closed-loop	Open-loop	Open-loop
Increase of complexity with increased			
- Accuracy	Polynomial	Linear	Polynomial
- Time horizon	Linear	Polynomial	Exponential*
- # of continuous states	Exponential	Polynomial	Polynomial
- # of discrete states	Linear	Polynomial	Polynomial
Required system knowledge	High	Higher than DP	Lower than DP

proposes a MPC based lateral control that tracks the lane centerline while avoiding collisions, [47] calculates the control inputs with MPC so that the vehicle tracks a planned path and [127] uses MPC for an ACC system that additionally aims to reduce the energy consumption.

Comparison

A comparison of the stated solution principles is presented in [131, p. 8] and further summarized in Table 2.4. The generalized statements apply to many methods found in literature but individual methods might differ.

DP computes a globally optimal control because of the sufficient optimality conditions of the Hamilton-Jacobi-Bellman equations. The computational effort grows linearly with increasing time horizon and number of discrete states. DP offers a closed-loop feedback control that is beneficial for the system stability. DP algorithms converge globally in the discrete state space. However, the required discretization leads to an exponential increase of computational effort with the number of continuous states. Therefore, DP usually can only be applied to systems with a state and control dimension up to four and requires coarse discretization. Therefore the solution accuracy is comparatively low. If the discretization is refined for more accurate solutions, the computational effort increases polynomially. DP requires a high system knowledge and can solve less classes of OCP than other principles.

IM are very accurate and the computational effort only grows linearly when the accuracy is increased. With respect to the time horizon and number of states, the complexity grows polynomially. IM can solve more OCP classes than DP but less than DM. IM only compute locally optimal open-loop controls which means that the control loop is not closed using system feedback information. IM converge within a smaller domain than DM and in general, applying IM requires more knowledge than other methods.

□ DM converge within a larger domain than IM. They can deal with numerous states, require less control knowledge and can solve more OCP classes than other methods. Like IM, DM provide only locally optimal open-loop control. The acceptable solution accuracy mainly results from discretization errors. The computational effort grows polynomially with the solution accuracy and the number of states. With respect to the time horizon, effort usually grows exponentially because in each time step all controls have to be analyzed. Therefore the optimization horizon is mostly restricted to few seconds [131, pp. 2-12, 27-37], [187, pp. 1415-1431].

Due to their complementary properties, different approaches are combined for solving difficult, farsighted trajectory optimization problems. Then DP provides a rough long-term plan or reference trajectory for a DM that computes feasible trajectories within a short time horizon. *Adopted from [78, 79].* An example is the energy-optimal adaptive cruise control in [186]. Closed-form solutions of calculus of variations can be applied to speed up DP. Table 2.5 shows the benefits of combining different methods [187, pp. 1430-1431].

Research gap: Both DP and DM are popular for automotive applications on their own and combining their orthogonal features offers great potential. However, the exponential growth of computational effort with increasing time horizon limits the application of DM to short time horizons. Hence,

Table 2.5: Suitability of Dynamic Programming (DP), direct methods (DM), indirect methods (IM) and combinations for selected problem features. Particularly suitable (++), suitable (+), less suitable (−). Derived from [187, p. 1430].

Approach	Many states	Continuous states	Global optimum	Long horizon
DP	−	−	+	+
DM	+	+	−	−
DP + DM	+	+	+	+
DP + DM + IM	+	+	+	++

the research gap regarding trajectory optimization consists of available DM with lower complexity.

2.3 B-spline data approximation

Data approximation can be viewed as a special case of trajectory optimization using the DM approach and instead of a polynomial, a B-spline function can represent the input trajectory from (2.4) or the system trajectory from (2.6). Following an overview of spline representations, this section states methods for determining a data approximating B-spline function and identifies the corresponding research gap.

Spline representations and their features

A function that is used for approximating data points needs to possess a sufficient number of parameters which translates to the number of degrees of freedom. A complicated relationship in the data usually requires more function parameters in order to achieve an acceptable representation.

With polynomials the degree is coupled to the number of parameters and as the degree is increased, the polynomials become computationally expensive to evaluate [5]. Assume a polynomial $p(u)$ given by

$$p(u) = \sum_{i=0}^{d} u^i c_i . \tag{2.8}$$

With a high degree d the monomial basis functions $u^d, u^{d-1}, \dots$ tend to take large values and therefore need to be multiplied with small coefficient values $c_d, c_{d-1}, \dots$. This can lead to ill-conditioned matrices, from which numerical instability issues can arise [35, pp. 272-274]. Increasing the degree of a polynomial approximation function also tends to result in an undesired oscillating behavior instead of better accuracy [35, pp. 292-294]. This effect is called Runge's phenomenon [150].

The explicit form of a curve $y = f(x)$ in two dimensions, x, y, states the value of one dependent variable, y, in terms of the independent variable, x. Most curves can also be represented in the implicit form $f(x, y) = 0$.

However, curves in three dimensions are not as easily defined in implicit form. Instead, they are usually represented in parametric form in terms of an independent variable u, called parameter. An advantage of the parametric form is that each dimension is defined by an explicit function that is not coupled to the others:

$$\boldsymbol{p}(u) = [x(u),\ y(u),\ z(u)]^\top \tag{2.9}$$

For example, a cubic polynomial parametric curve $\boldsymbol{p}(u)$ reads

$$\begin{aligned} \boldsymbol{p}(u) &= \sum_{k=0}^{3} u^k \boldsymbol{c}_k = \boldsymbol{u}^\top \boldsymbol{c},\ \boldsymbol{u} = \left[1,\ u,\ u^2,\ u^3\right]^\top, \\ \boldsymbol{c}_k &= \left[c_{xk},\ c_{yk},\ c_{zk}\right]^\top,\ \boldsymbol{c} = \left[\boldsymbol{c}_0,\ \boldsymbol{c}_1,\ \boldsymbol{c}_2,\ \boldsymbol{c}_3\right]^\top. \end{aligned} \tag{2.10}$$

$\boldsymbol{c}$ is viewed as a 4×1 matrix, whose elements $\boldsymbol{c}_k$ are 3×1 matrices. Each coefficient vector $\boldsymbol{c}_k$ has independent components for each dimension, hence there are three independent functions, each of the form (2.8) with $d = 3$. Without loss of generality, $0 \leq u \leq 1$ can be assumed. Figure 2.2 illustrates such a curve segment.

Defining each explicit function in terms of two parameters u and v, leads to a parametric surface $\boldsymbol{p}(u, v)$, e.g. for three dimensions:

$$\boldsymbol{p}(u, v) = [x(u, v),\ y(u, v),\ z(u, v)]^\top \tag{2.11}$$

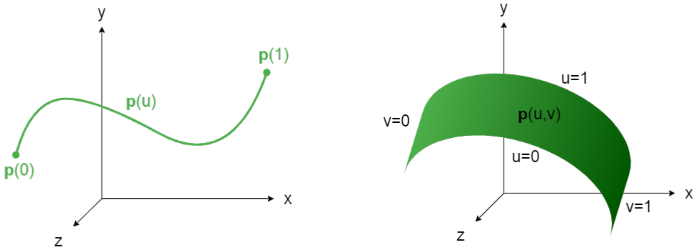


Figure 2.2: Curve segment (left) and surface patch (right) [5, pp. 508-509,adapted].

A surface patch, as also depicted by Figure 2.2, can be defined by varying u and v over the rectangle $0 \leq u, v \leq 1$. A bicubic surface patch is given by

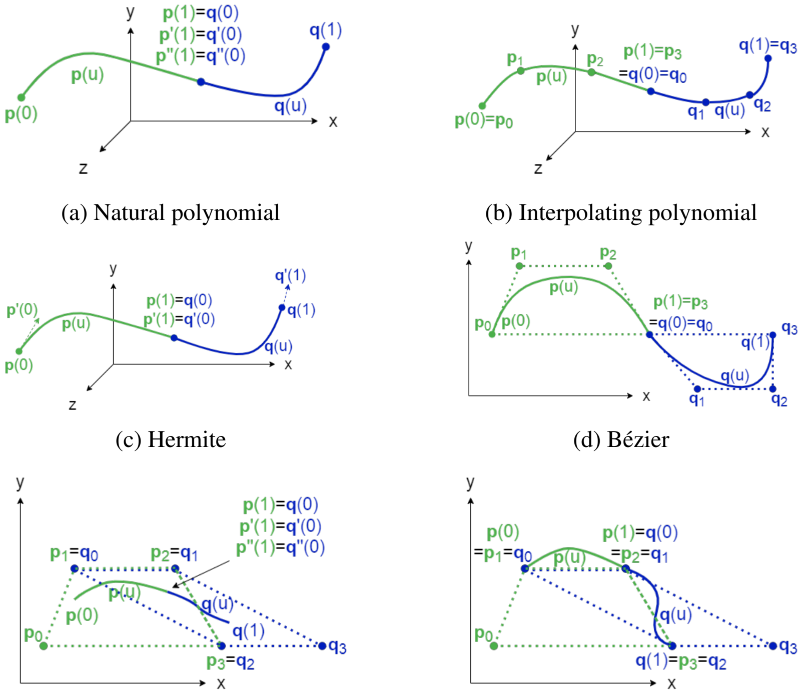
$$\begin{gathered} \boldsymbol{p}(u, v) = \sum_{i=0}^{3} \sum_{j=0}^{3} u^i v^j \boldsymbol{c}_{ij} = \boldsymbol{u}^\top \boldsymbol{C} \boldsymbol{v}, \\ \boldsymbol{v} = \left[1,\ v,\ v^2,\ v^3\right]^\top,\ \boldsymbol{c}_{ij} = \left[c_{xij},\ c_{yij},\ c_{zij}\right]^\top \end{gathered} \tag{2.12}$$

$\boldsymbol{C} = \left[\boldsymbol{c}_{ij}\right]$ is a 4x4 matrix with elements $\boldsymbol{c}_{ij}$. A surface patch can be seen as the limit of a collection of curves that result from keeping one parameter constant and varying the other [5, pp. 503-515].

Spline functions are piecewise defined, which decouples the number of parameters from the degree of the function. Spline functions allow to increase the number of function parameters while still using polynomials of low degree in order to avoid numerical stability issues and Runge's phenomenon. At the join points, neighboring pieces of a spline function are continuously differentiable to a certain extent. A function that is continuously differentiable up to its r^{th} derivative is called a C^r continuous function [35, p. 26].

The remainder of this subsection reviews the most common cubic spline types. Figure 2.3 illustrates them and Table 2.6 compares their features.

A straight-forward approach to construct a cubic spline function is to connect cubic polynomials by specifying continuity requirements at their join points as explained in [35, pp. 294-299]. In literature this approach is also referred to as natural polynomial spline [9].



(a) Natural polynomial

(b) Interpolating polynomial

(c) Hermite

(d) Bézier

(e) B-spline

(f) Catmull-Rom spline

Figure 2.3: Cubic curve segments $\boldsymbol{p}(u)$, and $\boldsymbol{q}(u)$ of various representation types depending on parameter $u \in [0, 1]$ with their control points $\boldsymbol{p}_i$, $\boldsymbol{q}_i$, $i = 0, \ldots, 3$ indicated by dots. Dotted lines indicate borders of convex hulls defined by the control points. [5, pp. 510-534, adapted].

For example, connecting two cubic polynomials $\boldsymbol{p}(u)$ and $\boldsymbol{q}(u)$ while demanding continuity up to the second derivative leads to eight degrees of freedom, which need to fulfill three equality constraints at the join point. Figure 2.3a illustrates this case. In order to determine possible spline functions, an undetermined linear system of equations needs to be solved. Each additional polynomial segment leads to four additional degrees of freedom and three additional continuity constraints. Furthermore, the parameters of all segments of the same dimension influence each other via the continuity

Table 2.6: Cubic representation types for functions, curves and surfaces.

Representation type	Control	Continuity at join points	Control point interpolation	Convex hull property
Natural polynomial	Global	C^2	–	–
Interpolating polynomial	Local	C^0	Yes	No
Hermite	Local	C^1	Yes	No
Bézier	Local	C^0	Yes	Yes
B-spline	Local	C^2	No	Yes
Catmull-Rom spline	Local	C^1	Yes	No

contraints and therefore changing the spline function at any point can affect the shape of the function at any other point. This feature is called *global control* and mostly undesired because it requires that all segments are designed within a single global calculation.

In contrast, the spline types described below offer *local control* of shape, meaning that changing a parameter only influences the function locally. Therefore each segment can be designed individually. Furthermore, they achieve continuity without solving a system of equations during application.

Geometrically interpretable parameters called *control points* are used to design the shape of these functions. Assume four control points

$$\boldsymbol{p}_k = [x_k,\ y_k,\ z_k]^\top,\ k = 0, 1, 2, 3, \tag{2.13}$$

which are equally spaced at $u = 0, {}^1\!/_3, {}^2\!/_3, 1$. $\boldsymbol{c}$ is determined such that $\boldsymbol{p}(u)$ from (2.10) interpolates the control points $\boldsymbol{p}_k$. The conditions

$$\boldsymbol{p}_0 = \boldsymbol{p}(0),\ \boldsymbol{p}_1 = \boldsymbol{p}({}^1\!/_3),\ \boldsymbol{p}_2 = \boldsymbol{p}({}^2\!/_3),\ \boldsymbol{p}_3 = \boldsymbol{p}(1) \tag{2.14}$$

read in matrix form

$$\boldsymbol{p} = \boldsymbol{A}\boldsymbol{c},\ \boldsymbol{p} = \begin{bmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \end{bmatrix},\ \boldsymbol{A} = \begin{bmatrix} \boldsymbol{u}(0)^\top \\ \boldsymbol{u}({}^1\!/_3)^\top \\ \boldsymbol{u}({}^2\!/_3)^\top \\ \boldsymbol{u}(1)^\top \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & {}^1\!/_3 & ({}^1\!/_3)^2 & ({}^1\!/_3)^3 \\ 1 & {}^2\!/_3 & ({}^2\!/_3)^2 & ({}^2\!/_3)^3 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{2.15}$$

$\boldsymbol{p}$ and $\boldsymbol{c}$ are 4×1 matrices, whose elements are 3×1 matrices. The control points given by $\boldsymbol{c} = \boldsymbol{A}^{-1}\boldsymbol{p} = \boldsymbol{M}\boldsymbol{p}$ with geometry matrix $\boldsymbol{M} = \boldsymbol{A}^{-1}$ achieve continuity at join points, but not for the derivatives. Figure 2.3b shows two curve segments $\boldsymbol{p}(u)$, $\boldsymbol{q}(u)$ with their control points [5, pp. 509-517].

For C^2 continuity, quintic polynomials would have to be used. Increasing the degree of a polynomial segment from three on complicates the calculation process with each increment significantly [11]. In particular, for polynomials of degree higher than four there is no general closed-form solution [23]. This makes nonnegativity checks by determining the roots costly.

By substituting $\boldsymbol{M}$ into (2.10), $\boldsymbol{p}(u)$ can be expressed in terms of the blending polynomials $\boldsymbol{b}(u)$ with

$$\boldsymbol{b}(u) = \boldsymbol{u}^\top \boldsymbol{M} = [b_0(u),\ b_1(u),\ b_2(u),\ b_3(u)] \tag{2.16}$$

and the control point vector $\boldsymbol{p}$:

$$\boldsymbol{p}(u) = \boldsymbol{u}^\top \boldsymbol{c} = \boldsymbol{u}^\top(\boldsymbol{M}\boldsymbol{p}) = (\boldsymbol{u}^\top \boldsymbol{M})\boldsymbol{p} = \boldsymbol{b}(u)\boldsymbol{p} \tag{2.17}$$

Interpolating curves can be extended to interpolating surfaces. Assume a 4×4 control point matrix $\boldsymbol{P} = \left[\boldsymbol{p}_{ij}\right]$ comprising 16 independent three-dimensional control points $\boldsymbol{p}_{ij},\ i = 0, \ldots, 3,\ j = 0, \ldots, 3$, which are again equally spaced at $u, v = 0, 1/3, 2/3, 1$. The curve for $v = 0$ must interpolate $\boldsymbol{p}_{00}, \boldsymbol{p}_{10}, \boldsymbol{p}_{20}, \boldsymbol{p}_{30}$:

$$\boldsymbol{p}(u, 0) = \boldsymbol{u}^\top \boldsymbol{M} \left[\boldsymbol{p}_{00}^\top,\ \boldsymbol{p}_{10}^\top,\ \boldsymbol{p}_{20}^\top,\ \boldsymbol{p}_{30}^\top\right]^\top = \boldsymbol{u}^\top \boldsymbol{C}\,[1,\ 0,\ 0,\ 0]^\top \tag{2.18}$$

$v = 1/3, 2/3, 1$ give the other three interpolating curves. Putting all of them together leads to $\boldsymbol{u}^\top \boldsymbol{M}\boldsymbol{P} = \boldsymbol{u}^\top \boldsymbol{C}\boldsymbol{A}^\top$ with $\boldsymbol{A}^{-1} = \boldsymbol{M}$. Solving for coefficient matrix $\boldsymbol{C} = \boldsymbol{M}\boldsymbol{P}\boldsymbol{M}^\top$ and substituting with (2.16) results in

$$\boldsymbol{p}(u, v) = \boldsymbol{u}^\top \boldsymbol{M}\boldsymbol{P}\boldsymbol{M}^\top \boldsymbol{v} = \boldsymbol{b}(u)\boldsymbol{P}\boldsymbol{b}(v)^\top = \sum_{i=0}^{3}\sum_{j=0}^{3} b_i(u) b_j(v) \boldsymbol{p}_{ij}. \tag{2.19}$$

Each term $b_i(u)b_j(v)$ describes a blending patch. The surface is formed from 16 blending patches, each weighted by a control point. Surfaces that are created in such a way are called *tensor product surfaces* and are an example of separable surfaces. They allow to work with functions in u and v independently [5, pp. 510-517].

The spline types described in the following can also be represented using (2.17) and (2.19). They differ only in the blending functions $\boldsymbol{b}(u)$ from (2.16), because their designs fulfill conditions other than (2.14) and (2.18).

Hermite curves depicted in Figure 2.3c offer C^1 continuity between different segments. However, instead of $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, the user needs to specify the values at the join points up to their first derivatives $\boldsymbol{p}'(0)$ and $\boldsymbol{p}'(1)$, so that consecutive tangents are collinear. This is often undesired because without an analytic formulation of the data, no derivative information is available.

Bézier curves are approximations to Hermite curves and do not need derivative information. They result when the derivatives of two segments at their join point are not required to be exactly equal. Therefore cubic Bézier curves only have C^0 continuity. Each segment is within the convex hull given by its control points, see Figure 2.3d. For spline types with this convex hull feature, only the discrete control points but not the continuous curve itself need to be evaluated to ensure that the curve fulfills certain constraints, e.g. nonnegativity.

For higher continuity, an alternative to increasing the polynomial degree is to not require the curve to interpolate any control point. This leads to B-spline curves (Figure 2.3e). Each of their segments only spans the distance between its two middle control points. In general they only come close to the control points but compensate with C^2 continuity at the join points.

Omitting the requirement that each spline segment must lie within the convex hull of its control point allows to form other types of splines. One of the most popular is the C^1 continuous Catmull-Rom spline (Figure 2.3f) that interpolates its control points. Compared to a Hermite curve it does also not require derivative information [5, pp. 509-535].

The remainder of this work will use cubic B-spline functions. Compared to natural splines, they offer numerically stable computations, local control and do not require solving a system of equations during their application. Compared to the other cubic spline functions, only B-splines are both C^2 continuous and fulfill the convex hull property. C^2 continuity is an important requirement for technical applications such as the definition of a jerk-free trajectory [55].

The convex hull property in combination with the geometrically interpretable control points offers a convenient and computationally efficient way to enforce constraints on the function because no evaluation of the function is needed. Enforcing constraints on an approximation function can be beneficial in case of knowledge about the data, e.g. that it is nonnegative.

□ Due to their favorable features, B-spline functions, curves and surfaces are widely used for data approximation [4, 84, 202] and for defining paths and trajectories of vehicles [17, 42, 45], robots [26, 45, 108, 162] and industrial machines [69, 200]. Moreover, the B-spline representation is common in computer graphics [93, 193] as well as in signal processing for filter design [128, 129, 148] and signal representation [120, 142, 143, 176, 177]. *Adopted from [78, 80].*

Methods for B-spline data approximation

□ According to (2.17), the value of a B-spline function equals the sum of basis functions (B-splines) or blending functions, weighted with their corresponding control points. Each B-spline in (2.16) is only nonzero within a bounded interval so that effects of control point changes are only local. Hence, the definition range of a B-spline function is also bounded. More extensively these features will be described in Section 3.1 and by Figure 3.2. *Adopted from [80].*

□ A common application of B-spline functions, curves and surfaces is fitting of data points. Fitting can either be interpolation or approximation. An interpolating B-spline function f must pass through all of the data points (s_p, y_p), i.e. $f(s_p) = y_p \ \forall p$. In contrast, an approximating B-spline function only minimizes the residuals $f(s_p) - y_p$ between the function and the data but does not pass through the data points in general.

In *offline applications*, in which all data points are available at once, fitting B-spline functions are often determined by least squares (LS) methods [31, 113, 159]. With the standard formula in batch form, all data points have to be collected first and then processed simultaneously. Therefore the number of data points P must be bounded. The computation usually involves a Cholesky or QR factorization and requires $\mathcal{O}(P)$ operations if one takes advantage of the banded matrix structure [20, pp. 327-331]. Such algorithms are stated in [20, pp. 117-121] and [58, pp. 152-160]. With the least squares (LS) algorithm each data point influences the result to the same extent. The weighted least squares (WLS) estimator, stated in (3.14), allows to weight measurements relative to each other [113, pp. 119-123].

In *online applications* data points are observed one after another and an ever-growing amount of data is common. Two groups of LS algorithms for online applications can be distinguished: First, growing memory LS algorithms apply an exponential weighting that forgets old data. Second, sliding window LS algorithms discard old data completely and need only finite storage [201]. Sliding window LS and sliding window WLS algorithms are proposed in [83, 201] and [33, 184], respectively. Re-computing the fitting function from scratch with each new data point is costly. Rank update and rank downdate methods allow to reuse an already known factorization for an efficient update of a solution after observations have been added or deleted [66, 124, 125].

In online applications data points can be outside the definition range of the B-spline function if the magnitude of the data points is not exactly known or changes over time, e.g. because one dimension of the data point refers to its observation time, that keeps increasing. Data points outside the definition range cannot be taken into account for approximation. Thereby the problem arises that the approximation might not reflect the data anymore. When using WLS the bounded definition range of B-spline functions does not present a problem because the number and position of B-splines can be changed if the fitting function is re-computed from scratch. Moreover, rank modification methods also support adding or deleting matrix columns [66]. This allows to extend, shrink or shift the definition range of the B-spline function. *Adopted from [80].*

□ In nonlinear weighted least squares (NWLS) problems, the solution depends on the function parameters in a nonlinear fashion. Based on the results of numerical experiments, [73] reports that a B-spline function is useful for solving NWLS problems as well because of its piecewise polynomial character and smoothness. For NWLS problems, several batch methods that work iteratively can be found in literature, e.g. the Newton method, Gauss-Newton method, Levenberg-Marquardt method, dog leg method of Powell, hybrid method of Madsen, Levenberg-Marquardt-Fletcher method. None of the algorithms is an exact method that computes an optimal solution [73]. A method for separable NWLS problems in which some parameters affect the solution linearly is derived in [149].

The Levenberg-Marquardt (LM) algorithm, described in more detail in Subsection 3.4.1, solves in each iteration a linearized NWLS problem [35, pp. 222-224]. A sliding window implementation of the LM algorithm is stated in [38]. *Adopted from [78].*

□ Recursive methods compute an approximating B-spline function recursively meaning that the approximation is updated with each new data point. This approach is preferred in online applications, in which data points are observed one after another. Recursive algorithms such as recursive least squares (RLS) [164, pp. 84-88] usually require less computational power than batch algorithms because they use smaller matrices and vectors whose sizes do not depend on the number of data points. The recursive computation is also referred to as progressive, iterative or sequential [110, 192]. This work will use all mentioned terms synonymously. In computer science, however, an iteration differs from a recursion. In [106], fitting B-spline curves and surfaces are iteratively constructed based on the idea of profit and loss modification without solving a linear system. The authors of [36] build on the progressive and iterative approximation technique for B-spline curve and surface fitting and prove that their algorithm achieves a least squares fit to the data points. A recursive algorithm for optimal smoothing B-spline surfaces inspired by the RLS method is presented in [50]. Algorithms that involve a Kalman filter (KF) are stated in [67, 104]. *Adopted from [80].*

Recursive algorithms for NWLS problems can be based on nonlinear filters. Overviews of commonly used filters and their features are provided by [57] and [187, p. 615]. In [2] a solution via a modified extended Kalman filter (EKF) is investigated and algorithms for offline and online applications are stated. The two-step estimator proposed in [68] splits the problem into a linear subproblem which is solved with a KF and a nonlinear subproblem to which the Gauss-Newton algorithm is applied. Section 2.4 will give an introduction into the aforementioned adaptive filters and categorize them.

Research gap: □ Publications of other researchers regarding the recursive data approximation with a B-spline function assume a constant definition range. For example, the approaches based on the KF in [67, 104] require that the KF state vector contains all control points that are estimated during the whole approximation procedure. Therefore the number of control points has to be bounded and specified in advance. As a result, these algorithms can only approximate data points that are within the bounded B-spline function definition range determined at the beginning. Other researchers have not addressed the possible issue of data points outside the inital B-spline function definition range. *Adopted from [80].*

Therefore the research gap regarding the recursive data approximation with a B-spline function consists of the lack of available online methods for data approximation with B-spline functions in the linear WLS and NWLS sense that can handle data that leaves the initially chosen bounded B-spline function definition range. For example, leaving the initial definition range is needed if an additional data point of an unbounded data set is observed in each time step and the values in at least one dimension of the data points keep increasing because they refer to the observation time. More detailed application examples are stated on page 70 and illustrated by Figure 3.9.

2.4 Adaptive filters

□ As a foundation for Chapter 3 and Chapter 5 this section gives an overview of several kinds of adaptive filters with focus on the Bayesian approach to state estimation, which calculates the probability density function (PDF) of the unknown state of a dynamic system. The required information for calculating the PDF stems partly from a system model and partly from previous measurements. The state estimation is performed by a recursive filter that alternates between a time update that predicts the state via the system model and a measurement update that corrects the estimate with the current measurement. *Adopted from [78].*

Assume a system whose input u and output y are measurable and known for a sequence of time steps $p = 1, \ldots, P$.

Adaptive filters include a system model $\hat{f}$ with free model parameter x. Sequentially they adjust x based on the sequence of u and y such that the parameter estimate $\hat{x}$ minimizes the residuals between the system model output $\hat{y} = \hat{f}(\hat{x}, u)$ and y [111, p. 1].

Filtering means that $\hat{x}_p$, referring to time step p, is calculated based on data of the current and all previous time steps, hence $p = P$. This is also referred to as measurement update. The estimation problem is called *prediction* in case of $p > P$ and *smoothing* if $p < P$ [72, p. 7].

Adaptive filters can be subdivided according to their system assumptions. In systems, in which x is assumed to change over time, it is also called *system state*, whereas x is mostly referred to as *parameter* in systems, in which it is assumed to be constant.

Linear adaptive filters

□ RLS and KF compute an optimal state estimate for systems with linear system and measurement equations as well as Gaussian system and measurement noise. They differ in that RLS assumes a constant state, whereas the KF is designed for tracking a time-variant state [164, p. 129], [57, pp. 3-5]. Use cases include parameter estimation [83] and path planning [189], respectively. The KF will be explained in Subsection 3.3.2 and applied within RBA in Subsection 3.3.3.

Nonlinear adaptive filters

In many scenarios the linear Gaussian assumptions do not apply and suboptimal approximate nonlinear Bayesian filters such as the extended Kalman filter (EKF), unscented Kalman filter (UKF) or particle filter (PF) are required [8].

The EKF applies a local first order Taylor approximation to the nonlinear system and measurement functions via Jacobians, in order to keep the linear state and measurement equations. System and measurement noise are both approximated with zero-mean Gaussian PDFs [57, p. 52]. Although the EKF is not suitable for systems with strong nonlinearity or non-Gaussian noise, it is still often successfully used for nonlinear state estimation [27]. For example, NWLS approximation via a modified EKF is presented in [2].

An alternative to the approximation of nonlinear state and measurement functions is the approximation of the PDFs. This can be done by propagating few state samples called sigma points through the nonlinear functions. A filter that follows this approach is referred to as sigma point Kalman filter. A well known representative is the UKF. It uses $2 \cdot J + 1$ deterministically chosen sigma points, whereby J denotes the system state dimensions. The PDFs are approximated as Gaussians whose means and variances are determined from the propagated sigma points [57, pp. 3-5, 62-70].

Compared to the EKF, the UKF offers at least second order accuracy [37] and is a derivative free filter [57, pp. 62-63], meaning that it does not require the evaluation of Jacobians, which is often computationally expensive in the EKF [27]. Several publications report nonlinear problems in which the UKF performs better than the EKF, e.g. for trajectory estimation [37, 63]. However, if the PDF cannot be well approximated by a Gaussian, because the PDF is multi-modal or has a strong skew, the UKF will also not perform well. Under such conditions, sequential Monte Carlo methods like the PF outperform Gaussian filters like EKF and UKF [8].

The PF approximates the PDF by a large set of randomly chosen state samples called particles. The state estimate is a weighted average of the particles. With increasing number of particles the PDF approximation by the particles becomes equivalent to the functional PDF representation and the estimate converges against the optimal estimate [8]. For nonlinear and non-Gaussian systems the PF allows to determine various statistical moments, whereas EKF and UKF are limited to the approximation of the first two moments [27]. However, the number of particles that is needed for sufficient approximation of the PDF increases exponentially with the state dimension [122]. The PF has been successfully applied to optimization [99] and prediction [190] of trajectories as well.

Many use cases involve a mixed linear/nonlinear system. Typically there are few nonlinear state dimensions and comparatively many linear Gaussian state dimensions. The marginalized particle filter (MPF) is beneficial for such problems as it combines KF and PF. The PF is only applied to the nonlinear states because the linear part of the state vector is marginalized out and optimally filtered with the KF.

This state decomposition is called Rao-Blackwellization [72, pp. 49-50] and can be described as an optimal Gaussian mixture approximation. Therefore the MPF is also called Rao-Blackwellized particle filter [70]. Marginalizing out linear states from the PF strongly reduces the computational effort because less particles suffice. This often enables real-time applications. Simultaneously the estimation accuracy usually increases [27, 205]. When the state dimension is large, the MPF tends to outperform the PF [123].

In the recent past, several publications have proposed approaches for localization [135, 191] and trajectory tracking [109, 205] that are based on the MPF because of its advantages for mixed linear/nonlinear systems. Automotive use cases include a road target tracking application, whose multimodality requires using a PF or MPF [166]. The MPF is chosen as it allows reducing the number of particles for less computational effort. Similarly, [122] presents a MPF application for lane tracking, in which the achieved particle reduction compared to a pure PF enables executing the algorithm in real-time in an embedded system. *Adopted from [78].*

According to (2.17), between spline function value and spline function control points there is a known linear relationship given by the blending or basis functions. Section 3.1 will state in (3.3) and (3.5) that this applies to both the value of a B-spline function and its derivatives as well. Therefore NWLS approximation leads to a mixed linear/nonlinear problem as long as there are target criteria that refer to the B-spline function or its derivatives directly. For being able to take into account this known linear relationship instead of having to estimate it, the iterative algorithm for NWLS approximation, NRBA, that is defined in Subsection 3.4.3, includes an MPF, which is described in Subsection 3.4.2.

Kernel adaptive filters

The two previous subsections assumed that knowledge about the system state with respect to the measured input u and measured output y is available. Sometimes this does not apply or the system features are difficult to specify. In such cases the unknown system function f with $y = f(x)$ can be approximated by $\hat{f}$. The approximation $\hat{f}$ is also called black-box model because it is created only from the data itself without system knowledge.

$\hat{f}$ can be determined with a kernel adaptive filter (KAF). Using a nonlinear mapping $\Phi(\cdot)$, KAFs transform the input data u into a high-dimensional feature space, in which more degrees of freedom are available to solve the problem with a linear adaptive filter. Due to a property of reproducing kernel Hilbert spaces no costly explicit mapping is needed to compute inner products $\langle \cdot, \cdot \rangle$ of the transformed data $\Phi(\boldsymbol{x}_i)$. Instead, inner product algorithms can be performed implicitly in feature space by replacing all inner products in the original problem by a kernel function κ [146]:

$$\kappa(\boldsymbol{x}_j, \boldsymbol{x}_m) = \langle \Phi(\boldsymbol{x}_j), \Phi(\boldsymbol{x}_m) \rangle \tag{2.20}$$

Often Gaussian kernels are used. These kernels differ from multi-dimensional PDFs of the standard normal distribution with variance σ^2 only in the missing normalization constant $1/\left(\sigma\sqrt{2\pi}\right)$:

$$\kappa(\boldsymbol{x}_j, \boldsymbol{x}_m) = \exp(-\left\|\boldsymbol{x}_j - \boldsymbol{x}_m\right\|^2 / 2\sigma^2) \tag{2.21}$$

The nonlinear mapping $\hat{f}$ is calculated as a linear combination of kernels, each of which is weighted by its coefficient. For example, by transforming the RLS cost function into feature space, a nonlinear kernel RLS (KRLS) algorithm is derived. With most kernel-based methods the required memory increases with the processed data points. In contrast, the Fixed-Budget KRLS (FB-KRLS) algorithm can operate with constant memory by pruning the least significant data [180]. This enables long-term online applications.

In summary KAFs are nonlinear adaptive filters that result from applying kernel method concepts to linear adaptive filters. KAFs combine the capability of approximating any nonlinear function with the advantage that they only require solving a convex optimization problem instead of a nonlinear problem. For convex problems there are solvers that operate within a fixed time interval. This is important for real-time applications [146].

Subsection 4.7.2 will apply the FB-KRLS to represent the power train of the research vehicle. The model is updated constantly during vehicle operation.

3 Data approximation with B-spline functions

This chapter deals with the general problem of recursive least squares approximation of data points with a B-spline function. First, Section 3.1 states a B-spline function definition in matrix form. Section 3.2 introduces a data set which will be approximated by a B-spline function. Section 3.3 addresses the case of linear least squares approximation and Section 3.4 the nonlinear case. Each of the two latter sections proposes a novel recursive algorithm and compares it with a well-known method for batch processing. Section 3.5 summarizes the scientific contribution.

3.1 B-spline function definition

A B-spline function consists of several polynomial basis splines (B-splines), all of which have the same degree. A B-spline of degree d is piecewise defined using $d + 1$ polynomial functions of degree d. Figure 3.1 illustrates the composition of a cubic B-spline. The solidly drawn parts of the depicted polynomial functions p_1, p_2, p_3 and p_4 build a B-spline.

Let $[\kappa_1, \kappa_2]$ denote a closed interval between and including κ_1 and κ_2 and let (κ_1, κ_2) denote an open interval between but not including κ_1 and κ_2.

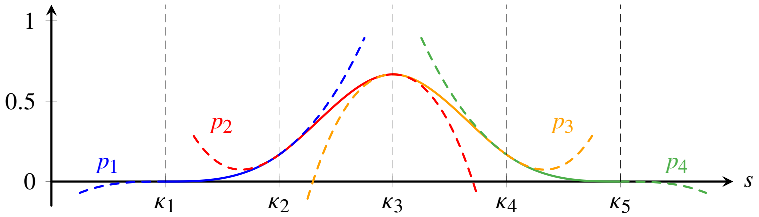


Figure 3.1: Piecewise definition of a cubic B-spline by the solid parts of four cubic polynomials $p_1, p_2, \ldots, p_4$. Vertical lines indicate the knots $\kappa_1, \ldots, \kappa_5$.

In the interval $[\kappa_1, \kappa_2)$, the B-spline is given by the first polynomial p_1, in the interval $[\kappa_2, \kappa_3)$ it is given by the second polynomial p_2 and so on. Outside the interval (κ_1, κ_5) the B-spline is zero. The dashed curves are only shown for better visualization of the polynomials but are not part of the B-spline. The interval borders $\kappa_1, \kappa_2, \ldots, \kappa_5$ are also called *knots*. Their distances influence the shape of the polynomials and therefore the shape of the B-spline as well.

□ A B-spline function is also piecewise defined. Its value is given by the weighted sum of J B-splines of degree d. The knot vector is $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \ldots, \kappa_{J+d+1})$. Strictly increasing knot values ($\kappa_k < \kappa_{k+1}$, $k = 1, 2, \ldots, J + d$) are assumed. $\boldsymbol{\kappa}$ and d determine the number and shape of B-splines. The j-th B-spline $b_j(s)$, $j = 1, 2, \ldots, J$ is positive only for $s \in (\kappa_j, \kappa_{j+d+1})$ and zero elsewhere [113, pp. 37-42].

The following definitions originate from [113, pp. 47-50, 65-70]: Let $[\kappa_\mu, \kappa_{\mu+1})$ be a spline interval and let μ denote the spline interval index with $d + 1 \leq \mu \leq J$. For $s \in [\kappa_\mu, \kappa_{\mu+1})$, the B-splines $b_j(s)$, $j = \mu - d, \ldots, \mu$ can be nonzero. Their values for a specific $s \in [\kappa_\mu, \kappa_{\mu+1})$ can be summarized in the B-spline vector $\boldsymbol{b}_{\mu,d}(s) = (b_{\mu-d}(s), b_{\mu-d+1}(s), \ldots, b_\mu(s)) \in \mathbb{R}^{1\times(d+1)}$ which can be computed according to (3.1):

$$\boldsymbol{b}_{\mu,d}(s) = \underbrace{\boldsymbol{B}_{\mu,1}(s)}_{\in\mathbb{R}^{1\times 2}} \underbrace{\boldsymbol{B}_{\mu,2}(s)}_{\in\mathbb{R}^{2\times 3}} \ldots \underbrace{\boldsymbol{B}_{\mu,\delta}(s)}_{\in\mathbb{R}^{\delta\times(\delta+1)}} \ldots \underbrace{\boldsymbol{B}_{\mu,d}(s)}_{\in\mathbb{R}^{d\times(d+1)}} \tag{3.1}$$

The B-spline matrix $\boldsymbol{B}_{\mu,\delta}(s) \in \mathbb{R}^{\delta\times(\delta+1)}$ is defined for each $\delta \in \mathbb{N}$ with $\delta \leq d$ and given by

$$\boldsymbol{B}_{\mu,\delta}(s) = \begin{bmatrix} \frac{\kappa_{\mu+1}-s}{\kappa_{\mu+1}-\kappa_{\mu+1-\delta}} & \frac{s-\kappa_{\mu+1-\delta}}{\kappa_{\mu+1}-\kappa_{\mu+1-\delta}} & 0 & \ldots & 0 \\ 0 & \frac{\kappa_{\mu+2}-s}{\kappa_{\mu+2}-\kappa_{\mu+2-\delta}} & \frac{s-\kappa_{\mu+2-\delta}}{\kappa_{\mu+2}-\kappa_{\mu+2-\delta}} & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & \frac{\kappa_{\mu+\delta}-s}{\kappa_{\mu+\delta}-\kappa_\mu} & \frac{s-\kappa_\mu}{\kappa_{\mu+\delta}-\kappa_\mu} \end{bmatrix}. \tag{3.2}$$

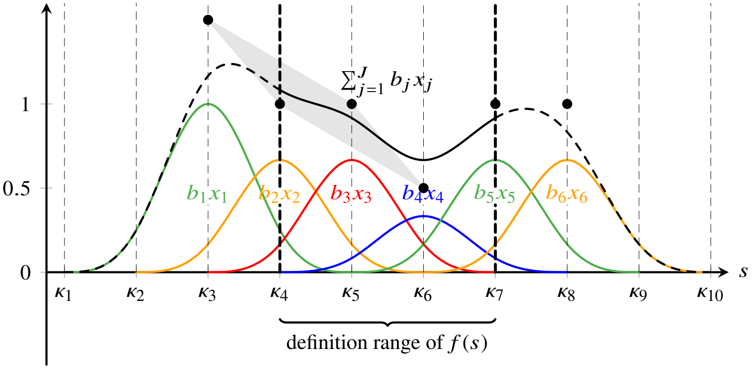


Figure 3.2: Construction of a cubic B-spline function: Equidistant knots $\kappa_1, \kappa_2, \ldots, \kappa_{10}$ (indicated by vertical straight lines) and $J = 6$ resulting B-splines b_j, $j = 1, 2, \ldots, J$ weighted with control points $x_1, x_2, \ldots, x_6$. The black line indicates the sum of all weighted B-splines and its solid part is the B-spline function. Black dots represent the control points and the gray area is the convex hull formed by the first four control points. *Adopted from [80].*

The B-spline function $f : \mathcal{D} \to \mathbb{R}, s \mapsto f(s)$ has the definition range $\mathcal{D} = [\kappa_{d+1}, \kappa_{J+1})$. For $s \in [\kappa_\mu, \kappa_{\mu+1})$, the B-spline function is given by

$$f(s) = \boldsymbol{b}_{\mu,d}(s)\boldsymbol{x}_{\mu,d} \tag{3.3}$$

with control point vector

$$\boldsymbol{x}_{\mu,d} = \left(x_{\mu-d}, x_{\mu-d+1}, \ldots, x_\mu\right)^\top . \tag{3.4}$$

Adopted from [78, 80]. As stated, the spline types mentioned in Section 2.3 can all be defined using the same structure because they only differ in the geometry matrix $\boldsymbol{M}$. Therefore (3.1) and (3.3) correspond to (2.16) and (2.17), which were obtained for an interpolating polynomial curve.

Figure 3.2 illustrates the construction of a B-spline function. Each cubic (d=3) B-spline b_j is weighted with a corresponding control point x_j, here $x_1 = 1.5$, $x_4 = 0.5$ and all other control points equal one. The cubic B-spline function $f(s)$ is given by the weighted sum of B-splines but only defined in the interval $[\kappa_4, \kappa_7)$. Its shape is given by the solid part of the black curve.

The black dots denote the control points of the B-spline function. The horizontal position of the j-th point is determined by the value of s, for which the j-th B-spline reaches its maximum. The vertical position is identical to x_j. In each interval $[\kappa_k, \kappa_{k+1})$, $k = d+1, \dots, K$, the B-spline function lies within the convex hull of the relevant control points. For $s \in [\kappa_4, \kappa_5)$, the four leftmost control points are the relevant ones. They form the convex hull indicated by the gray shaded area.

□ A B-spline function of degree d is $d-1$ times continuously differentiable. For $r \in \mathbb{N}_0$, the r-th derivative $\frac{\partial^r}{\partial s^r} f(s)$ of the B-spline function with respect to s is given by

$$\frac{\partial^r}{\partial s^r} f(s) = \frac{\partial^r}{\partial s^r} \boldsymbol{b}_{\mu,d}(s) \boldsymbol{x}_{\mu,d} \tag{3.5}$$

with B-spline vector

$$\begin{aligned}&\frac{\partial^r}{\partial s^r} \boldsymbol{b}_{\mu,d}(s) = \\ &\begin{cases} \frac{d!}{(d-r)!} \boldsymbol{B}_{\mu,1}(s) \dots \boldsymbol{B}_{\mu,d-r}(s) \boldsymbol{B}'_{\mu,d-r+1} \dots \boldsymbol{B}'_{\mu,d}, & \text{if } r \leq d \\ \boldsymbol{0}_{1\times(d+1)}, & \text{otherwise.} \end{cases}\end{aligned} \tag{3.6}$$

$\boldsymbol{0}_{1\times(d+1)}$ denotes a $1 \times (d+1)$ zero matrix. The matrix $\boldsymbol{B}'_{\mu,\delta} \in \mathbb{R}^{\delta\times(\delta+1)}$ is obtained by differentiating all entries in $\boldsymbol{B}_{\mu,\delta}(s)$ with respect to s:

$$\boldsymbol{B}'_{\mu,\delta} = \begin{bmatrix} \frac{-1}{\kappa_{\mu+1}-\kappa_{\mu+1-\delta}} & \frac{1}{\kappa_{\mu+1}-\kappa_{\mu+1-\delta}} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & \frac{-1}{\kappa_{\mu+\delta}-\kappa_{\mu}} & \frac{1}{\kappa_{\mu+\delta}-\kappa_{\mu}} \end{bmatrix} \tag{3.7}$$

Adopted from [78, 80].

According to [161] the antiderivative or indefinite integral $F(s)$ of $f(s)$ with degree $d^{\text{Int}} = d+1$ is given by

$$F(s) = \int f(s) ds = \boldsymbol{b}^{\text{Int}}_{\mu^{\text{Int}},d^{\text{Int}}}(s) \boldsymbol{x}^{\text{Int}}_{\mu^{\text{Int}},d} \tag{3.8}$$

$\boldsymbol{b}^{\text{Int}}$ uses the stricly increasing knot vector $\boldsymbol{\kappa}^{\text{Int}}$ which differs from $\boldsymbol{\kappa}$ only by an additional knot at either end for the increased degree. μ^{Int} equals $\mu + 1$.

The control point vector $\boldsymbol{x}^{\text{Int}}$ comprises one additional component than $\boldsymbol{x}$ and can be computed according to (3.9):

$$x_1^{\text{Int}} = 0,\ x_{j+1}^{\text{Int}} = \frac{\kappa_{j+d^{\text{Int}}+1} - \kappa_j}{d^{\text{Int}} + 1} \cdot \sum_{i=1}^{j} x_i,\ j = 1, \dots, J \tag{3.9}$$

The definite integral of $f(s)$ over the interval $[s_1, s_2]$ then reads

$$\int_{s_1}^{s_2} f(s)ds = F(s_2) - F(s_1). \tag{3.10}$$

3.2 Structure of the data set

□ Section 3.3 and Section 3.4 will work with the set $\{(s_p, \boldsymbol{y}_p)\}_{p=1,2,\dots,P}$ of P data points. p denotes the time step, at which data point $(s_p, \boldsymbol{y}_p)$ is measured or observed.

s_p is the value of the independent variable s at time step p. The vector of independent variables $\boldsymbol{s}$ is given by $\boldsymbol{s} = (s_1, \dots, s_p, \dots, s_P)^\top$.

$\boldsymbol{y}_p = (y_{p,1}, y_{p,2}, \dots, y_{p,v}, \dots, y_{p,V_p})^\top$ is a vector of V_p measurements y that refer to s_p and may come from different sensors. $V_p \in \mathbb{N}$ can be different for each $\boldsymbol{y}_p$ but it is assumed that $V_p \ll P\ \forall p$. The vector of all measurements $\boldsymbol{y}$ is composed as follows:

$$\boldsymbol{y}^\top = (\underbrace{y_{1,1}, \dots, y_{1,V_1}}_{=:\boldsymbol{y}_1^\top}, \dots, \boldsymbol{y}_p^\top, \dots, \underbrace{y_{P,1}, \dots, y_{P,V_P}}_{=:\boldsymbol{y}_P^\top}) \tag{3.11}$$

Adopted from [78, 80].

3.3 Methods for linear weighted least squares problems

Subsection 3.3.1 describes the WLS approach followed by the KF algorithm in Subsection 3.3.2. Subsection 3.3.3 presents the RBA algorithm. Its effectiveness is demonstrated in comparison with the WLS solution in Subsection 3.3.4.

3.3.1 Weighted least squares estimator

□ The linear weighted least squares (WLS) method estimates the constant state vector $\boldsymbol{x} \in \mathbb{R}^{L\times 1}$ of a linear system

$$\boldsymbol{y} = \mathbf{C}\boldsymbol{x} + \boldsymbol{v}. \tag{3.12}$$

$\boldsymbol{y} \in \mathbb{R}^{N\times 1}$ is the vector of measurements from (3.11) and $\mathbf{C}$ denotes the measurement matrix that relates $\boldsymbol{x}$ to $\boldsymbol{y}$. The measurement noise $\boldsymbol{v} \in \mathbb{R}^{N\times 1}$ is assumed to be an uncorrelated white noise process with mean zero. This implies that the covariance matrix of measurement noise $\mathbf{R}$ is a diagonal matrix and $\mathbf{R}_{i;i},\ i = 1,\dots,N$ is the variance of measurement $\boldsymbol{y}_i$ which can differ from the variances of other measurements [164]. The assumptions for $\{\boldsymbol{v}\}$ can be generalized to a correlated noise process. This is termed generalized linear model [138, p. 143]. Then $\mathbf{R}$ is a positive definite matrix [20, p. 374].

The linear WLS estimate $\hat{\boldsymbol{x}}$ minimizes the sum of squared errors between the measurements $\boldsymbol{y}$ and the vector $\mathbf{C}\boldsymbol{x}$ which are weighted with the reciprocals of the variances of the measurements:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} (\boldsymbol{y} - \mathbf{C}\boldsymbol{x})^\top \mathbf{R}^{-1} (\boldsymbol{y} - \mathbf{C}\boldsymbol{x}) \tag{3.13}$$

The solution to optimization problem (3.13) is given by the closed-form estimator

$$\hat{\boldsymbol{x}} = (\mathbf{C}^\top \mathbf{R}^{-1} \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{R}^{-1} \boldsymbol{y} \text{ [164].} \tag{3.14}$$

From (3.3) follows that the value of a B-spline function is a linear combination of its control points. Therefore WLS can be used to determine the control points such that the function approximates the set of data points defined in Section 3.2. Then $\mathbf{C}$ is a $\left(\sum_{p=1}^{P} V_p\right) \times J$ matrix because $\boldsymbol{y}$ comprises $\sum_{p=1}^{P} V_p$ scalar components $y_{p,v},\ p = 1,\dots,P,\ v = 1,\dots,V_p$ (c.f. (3.11)) and there are J B-splines. $y_{p,v}$ is the $\tilde{v}$-th component of $\boldsymbol{y}$ $\left(\tilde{v} = \sum_{\tilde{p}=1}^{p-1} V_{\tilde{p}} + v\right)$ and provides information about $\frac{\partial^r}{\partial s^r} f(s_p)$ with $s_p \in [\kappa_\mu, \kappa_{\mu+1})$ and an $r \in \mathbb{N}_0$. The $\tilde{v}$-th row of $\mathbf{C}$ is given by $\mathbf{C}_{\tilde{v};1,\dots,J} = \boldsymbol{c}$ with

$$\boldsymbol{c} = \left(\mathbf{0}_{1\times(\mu-(d+1))}, \frac{\partial^r}{\partial s^r} \boldsymbol{b}_{\mu,d}(s_p), \mathbf{0}_{1\times(J-\mu)}\right) \tag{3.15}$$

Herein $\mathbf{0}_{r\times c}$ denotes a $r \times c$ zero matrix. *Adopted from [80].*

3.3.2 Kalman filter

□ The Kalman filter (KF) is an established method for estimating the state of a dynamic system. Applications include tracking, navigation, sensor data fusion and process control [60, pp. 4-5]. The KF can be seen as a generalization of the RLS method [164, p. 129].

The linear KF estimates the state vector $\boldsymbol{x}_p \in \mathbb{R}^{L\times 1}$ of a linear time-discrete system

$$\boldsymbol{x}_p = \boldsymbol{\mathcal{A}}_p \boldsymbol{x}_{p-1} + \boldsymbol{\mathcal{B}}_p \boldsymbol{u}_p + \boldsymbol{\omega}_p \qquad \text{(State equation)} \tag{3.16}$$

$$\boldsymbol{y}_p = \boldsymbol{C}_p \boldsymbol{x}_p + \boldsymbol{v}_p \qquad \text{(Measurement equation)} \tag{3.17}$$

where $p \in \mathbb{N}$ denotes the time step. $\boldsymbol{\mathcal{A}}_p$ is the state transition matrix that relates $\boldsymbol{x}_{p-1}$ to $\boldsymbol{x}_p$, $\boldsymbol{u}_p \in \mathbb{R}^{M\times 1}$ is an input signal vector with known influence on $\boldsymbol{x}_p$ and $\boldsymbol{\mathcal{B}}_p$ is the input matrix that relates $\boldsymbol{u}_p$ to $\boldsymbol{x}_p$. The vector of measurements is denoted by $\boldsymbol{y}_p \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{C}_p$ is the measurement matrix that relates $\boldsymbol{x}_p$ to $\boldsymbol{y}_p$. $\boldsymbol{\omega}_p \in \mathbb{R}^{L\times 1}$ is the process noise with covariance matrix $\boldsymbol{Q}_p$ and $\boldsymbol{v}_p \in \mathbb{R}^{N\times 1}$ is the measurement noise with covariance matrix $\boldsymbol{\mathcal{R}}_p$. Both $\{\boldsymbol{\omega}_p\}$ and $\{\boldsymbol{v}_p\}$ are uncorrelated white noise processes with mean zero which implies that $\boldsymbol{Q}_p$ and $\boldsymbol{\mathcal{R}}_p$ are diagonal matrices [164, p. 124].

Algorithm 1: Kalman filter. *Adopted from [80].*

```
Input: $\hat{\boldsymbol{x}}^+_{p-1}, \boldsymbol{\mathcal{P}}^+_{p-1}, \boldsymbol{u}_p, \boldsymbol{y}_p, \boldsymbol{\mathcal{A}}_p, \boldsymbol{\mathcal{B}}_p, \boldsymbol{C}_p, \boldsymbol{Q}_p, \boldsymbol{\mathcal{R}}_p$
/* Time update                                        */
$\hat{\boldsymbol{x}}^-_p \leftarrow \boldsymbol{\mathcal{A}}_p \hat{\boldsymbol{x}}^+_{p-1} + \boldsymbol{\mathcal{B}}_p \boldsymbol{u}_p$
$\boldsymbol{\mathcal{P}}^-_p \leftarrow \boldsymbol{\mathcal{A}}_p \boldsymbol{\mathcal{P}}^+_{p-1} \boldsymbol{\mathcal{A}}_p{}^\top + \boldsymbol{Q}_p$
/* Measurement update                                 */
$\boldsymbol{\mathcal{K}}_p \leftarrow \boldsymbol{\mathcal{P}}^-_p \boldsymbol{C}_p{}^\top (\boldsymbol{C}_p \boldsymbol{\mathcal{P}}^-_p \boldsymbol{C}_p{}^\top + \boldsymbol{\mathcal{R}}_p)^{-1}$
$\hat{\boldsymbol{x}}^+_p \leftarrow \hat{\boldsymbol{x}}^-_p + \boldsymbol{\mathcal{K}}_p (\boldsymbol{y}_p - \boldsymbol{C}_p \hat{\boldsymbol{x}}^-_p)$
$\boldsymbol{\mathcal{P}}^+_p \leftarrow (\boldsymbol{I} - \boldsymbol{\mathcal{K}}_p \boldsymbol{C}_p) \boldsymbol{\mathcal{P}}^-_p (\boldsymbol{I} - \boldsymbol{\mathcal{K}}_p \boldsymbol{C}_p)^\top + \boldsymbol{\mathcal{K}}_p \boldsymbol{\mathcal{R}}_p \boldsymbol{\mathcal{K}}_p{}^\top$
Output: $\hat{\boldsymbol{x}}^+_p, \boldsymbol{\mathcal{P}}^+_p$
```

The KF consists of a sequel of equations, which are computed for each time step and summarized in Algorithm 1, in which $\boldsymbol{I}$ denotes the identity matrix with appropriate dimensions. The KF performs a time update followed by a measurement update. During the time update, the state estimate is updated based on the knowledge about the system specified by (3.16).

Both the a priori estimate $\hat{\boldsymbol{x}}_p^-$ and the covariance $\mathcal{P}_p^-$ of the a priori estimation error are calculated. During the measurement update, the Kalman gain $\mathcal{K}_p$ is computed and used together with the information provided by measurement $\boldsymbol{y}_p$ for the calculation of the corresponding a posteriori quantities $\hat{\boldsymbol{x}}_p^+$ and $\mathcal{P}_p^+$ [164, pp. 124-129]. KF generalizations for correlated or colored noise processes are stated in [164, pp. 183-193].

If the state vector $\boldsymbol{x}_p$ is constant, then $\mathcal{A}_p = \boldsymbol{I}$, $\boldsymbol{\omega}_p = \boldsymbol{0}$ and $\boldsymbol{u}_p = \boldsymbol{0}$, whereby $\boldsymbol{0}$ is the zero matrix with appropriate dimensions. In this case, the time update is redundant and the KF simplifies to the RLS algorithm [164, p. 129].

Data approximation by a polynomial using RLS is described in [164, pp. 92-93]. Due to the unbounded definition range of a polynomial, applying RLS saves computational effort without limiting the approximation compared to a KF.

Data approximation by a B-spline based on the KF is proposed in [67, 104]. These approaches require that the state vector $\boldsymbol{x}_p$ comprises all control points that are estimated during the whole approximation procedure. This means that $\boldsymbol{x}_p$, i.e. what is estimated, is constant and only the estimation $\hat{\boldsymbol{x}}_p$ can change. Hence, RLS would suffice. Combined with the bounded definition range of a B-spline function, the missing ability to change which control points are estimated leads to a definition range of the approximation that is also bounded, constant and needs to be specified in advance.

In contrast to these approaches, the recursive B-spline approximation algorithm proposed in Subsection 3.3.3 takes advantage of the time update, that the KF provides, to shift the control points in $\boldsymbol{x}_p$. As a consequence, RBA can shift the definition range of the B-spline function to consider data points outside of the current definition range. *Adopted from [80].*

3.3.3 Recursive B-spline approximation algorithm

□ The recursive B-spline approximation (RBA) algorithm computes an approximating B-spline function $f(s)$ of degree d for the set of data points from Section 3.2 iteratively using the KF. Algorithm 2 summarizes the calculations. $I \in \mathbb{N}$ denotes the constant number of spline intervals of $f(s)$. The KF state estimate $\hat{\boldsymbol{x}}_p = (\hat{x}_{p_1}, \hat{x}_{p_2}, \ldots, \hat{x}_{p_J})^\top$ comprises $J = d + I$ components which are the estimated control points of $f(s)$. The knot vector $\boldsymbol{\kappa}_p = (\kappa_{p_1}, \kappa_{p_2}, \ldots, \kappa_{p_K})$ for time step p has to contain $K = J + d + 1$ knots. $\mathcal{D}_p = [\kappa_{p_{d+1}}, \kappa_{p_{J+1}})$ is the definition range of $f(s)$ at p.

Algorithm 2: Recursive B-spline approximation. *Adopted from [80].*

```
Input: $\boldsymbol{\kappa}_{p-1}, \hat{\boldsymbol{x}}^{+}_{p-1}, \boldsymbol{\mathcal{P}}^{+}_{p-1}, \boldsymbol{\mathcal{R}}_{p}, s_p, \boldsymbol{y}_p, \bar{\boldsymbol{\kappa}}_p, \bar{x}, \bar{p}, \bar{q}$
$J \leftarrow$ length$(\hat{\boldsymbol{x}}^{+}_{p-1})$
$K \leftarrow$ length$(\boldsymbol{\kappa}_{p-1})$
$d \leftarrow K - J - 1$
$I \leftarrow J - d$
$\sigma \leftarrow 0$
if $s_p \geq \kappa_{p-1_{J+1}}$ then
    if $s_p \geq \kappa_{p-1_{K}}$ then
        $\sigma \leftarrow d + 1$
    else
        $\sigma$ such that $s_p \in [\kappa_{p-1_{d+I+1+\sigma}}, \kappa_{p-1_{d+I+2+\sigma}})$
    end
else if $s_p < \kappa_{p-1_{d+1}}$ then
    if $s_p < \kappa_{p-1_{1}}$ then
        $\sigma \leftarrow -(d + 1)$
    else
        $\sigma$ such that $s_p \in [\kappa_{p-1_{d+1+\sigma}}, \kappa_{p-1_{d+2+\sigma}})$
    end
end
$\boldsymbol{\mathcal{A}}_p \in \mathbb{R}^{J \times J}$ from (3.19)
$\boldsymbol{Q}_p \leftarrow \bar{q}\boldsymbol{I}_{J \times J}$
if $\sigma \geq 0$ then
    $\boldsymbol{\kappa}_p$ from (3.18)
    $\boldsymbol{u}_p \leftarrow (\boldsymbol{0}_{1\times(J-\sigma)}, \bar{x}\boldsymbol{1}_{1\times\sigma})^{\top}$
    $\boldsymbol{Q}_{p_{m;m}} \leftarrow \bar{p},\ m = J - \sigma + 1, J - \sigma + 2, \ldots, J$
else
    $\boldsymbol{\kappa}_p$ from (3.25)
    $\boldsymbol{u}_p \leftarrow (\bar{x}\boldsymbol{1}_{1\times(-\sigma)}, \boldsymbol{0}_{1\times(J+\sigma)})^{\top}$
    $\boldsymbol{Q}_{p_{m;m}} \leftarrow \bar{p},\ m = 1, 2, \ldots, -\sigma$
end
$\mu$ such that $s_p \in [\kappa_{p_{\mu}}, \kappa_{p_{\mu+1}})$
$V_p \leftarrow$ length$(\boldsymbol{y}_p)$
$\boldsymbol{C}_p \in \mathbb{R}^{V_p \times J}$ from (3.27)
$[\hat{\boldsymbol{x}}^{+}_{p}, \boldsymbol{\mathcal{P}}^{+}_{p}] \leftarrow$ Algorithm 1$(\hat{\boldsymbol{x}}^{+}_{p-1}, \boldsymbol{\mathcal{P}}^{+}_{p-1}, \boldsymbol{u}_p, \boldsymbol{y}_p, \boldsymbol{\mathcal{A}}_p, \boldsymbol{I}_{J\times J}, \boldsymbol{C}_p, \boldsymbol{Q}_p, \boldsymbol{\mathcal{R}}_p)$
Output: $\boldsymbol{\kappa}_p, \hat{\boldsymbol{x}}^{+}_{p}, \boldsymbol{\mathcal{P}}^{+}_{p}$
```

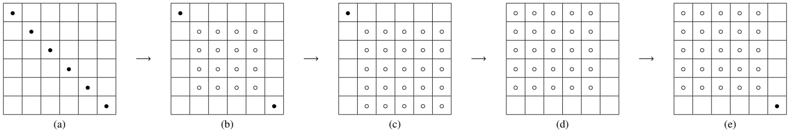


Figure 3.3: Changes of covariance matrix elements for $d = 3$ and $I = 3$. • indicates a large positive value, empty cells indicate zeros and ∘ denotes comparatively small values. *Adopted from [80].*

Initialization

$\hat{\boldsymbol{x}}_p$ is initialized with $\hat{\boldsymbol{x}}_0^+ = \bar{x}\boldsymbol{1}_{J\times 1}$, where $\boldsymbol{1}_{J\times 1}$ denotes a $J \times 1$ matrix of ones and $\bar{x}$ a scalar quantity of the magnitude of measurements $y_{p,v}$ that refer to $\frac{\partial^r}{\partial s^r} f(s)$ with $r = 0$.

The covariance matrix of a posteriori estimation error $\boldsymbol{\mathcal{P}}^+$ is initialized with $\boldsymbol{\mathcal{P}}_0^+ = \bar{p}\boldsymbol{I}_{J\times J}$, where $\boldsymbol{I}_{J\times J}$ is a $J \times J$ identity matrix. The scalar $\bar{p}$ should be chosen large $\left(\text{e.g. } 10^4\right)$ because then $\hat{\boldsymbol{x}}_p$ will quickly deviate from its initial value $\hat{\boldsymbol{x}}_0^+$ in such a way that $f(s)$ approximates the data points. If the elements in $\boldsymbol{\mathcal{P}}_p^-$ are small, this signals to the KF that the state estimate $\hat{\boldsymbol{x}}_p^-$ is very certain and therefore it will hardly be updated using the measurements. If the KF updates $\hat{\boldsymbol{x}}_p^+$ as intended, the elements in $\boldsymbol{\mathcal{P}}_p^+$ become smaller as p increases.

In the long-run, $\boldsymbol{\mathcal{P}}_p^-$ is strongly influenced by the covariance matrix of process noise $\boldsymbol{Q}_p$ according to line 2 of Algorithm 1. If the elements in $\boldsymbol{Q}_p$ are large, the elements in $\boldsymbol{\mathcal{P}}_p^-$ remain large too. This can lead to volatile state estimates $\hat{\boldsymbol{x}}_p$ that do not converge to a certain value. Then $f(s)$ will not approximate the data points well. For that reason a very small positive value $\bar{q}$ $\left(\text{e.g. } \bar{q} = 10^{-12}\right)$ is chosen for $\boldsymbol{Q}_p = \bar{q}\boldsymbol{I}_{J\times J}$.

Time update with shift operation

RBA compares the knot vector $\boldsymbol{\kappa}_{p-1}$ with s_p in order to determine whether $s_p \in \mathcal{D}_{p-1}$. If necessary, a shift of $\mathcal{D}_{p-1}$ is performed during the time update of the KF such that $s_p \in \mathcal{D}_p$. The variable σ indicates the shift direction of $\mathcal{D}_{p-1}$ and the number of positions by which elements are shifted. $\sigma > 0$ means a right shift of $\mathcal{D}_{p-1}$, $\sigma < 0$ a left shift of $\mathcal{D}_{p-1}$ and for $\sigma = 0$ no shift is performed because $s_p \in \mathcal{D}_{p-1}$. Algorithm 2 computes σ from line 5 to line 18.

For example, assume $d = 3$, $I = 3$ and $\boldsymbol{\kappa}_{p-1} = (1, 2, \dots, 10)$, then $\mathcal{D}_{p-1} = [4, 7)$. If $s_p = 8.5$, two additional knots are needed to be able to perform a right shift by two elements ($\sigma = 2$). 11 and 12 as additional knots give $\boldsymbol{\kappa}_p = (3, 4, \dots, 12)$ and hence $s_p \in \mathcal{D}_p = [6, 9)$.

Algorithm 2 distinguishes between $\sigma \geq 0$ and $\sigma < 0$. It assumes that the $|\sigma|$ additional knots are the σ last components of the knot vector $\bar{\boldsymbol{\kappa}}_p = (\bar{\kappa}_{p_1}, \bar{\kappa}_{p_2}, \dots, \bar{\kappa}_{p_K})$ in case of $\sigma > 0$ and that they are the $-\sigma$ first components of $\bar{\boldsymbol{\kappa}}_p$ in case of $\sigma < 0$.

Case 1: Right shift of definition range ($\sigma \geq 0$)

The new knot vector is

$$\begin{aligned}\boldsymbol{\kappa}_p \leftarrow (&\kappa_{p-1_{\sigma+1}}, \kappa_{p-1_{\sigma+2}}, \dots, \kappa_{p-1_K},\\ &\bar{\kappa}_{p_{K-\sigma+1}}, \bar{\kappa}_{p_{K-\sigma+2}}, \dots, \bar{\kappa}_{p_K}).\end{aligned} \tag{3.18}$$

The elements in $\hat{\boldsymbol{x}}^+_{p-1}$ are shifted by line 1 of Algorithm 1 using the state transition matrix

$$\boldsymbol{\mathcal{A}}_p \in \mathbb{R}^{J \times J} \text{ with } \boldsymbol{\mathcal{A}}_{p_{g;h}} = \begin{cases} 1, & \text{if } h = g + \sigma \\ 0, & \text{otherwise.} \end{cases} \tag{3.19}$$

$\hat{\boldsymbol{x}}^-_p \leftarrow \boldsymbol{\mathcal{A}}_p \hat{\boldsymbol{x}}^+_{p-1}$ updates the old estimate $\hat{\boldsymbol{x}}^+_{p-1}$ to

$$\hat{\boldsymbol{x}}^-_p = (\hat{x}_{p-1_{\sigma+1}}, \hat{x}_{p-1_{\sigma+2}}, \dots, \hat{x}_{p-1_{J-\sigma}}, \boldsymbol{0}_{1\times\sigma})^\top. \tag{3.20}$$

With the second part of the instruction ($\hat{\boldsymbol{x}}^-_p \leftarrow \hat{\boldsymbol{x}}^-_p + \boldsymbol{\mathcal{B}}_p \boldsymbol{u}_p$), input matrix $\boldsymbol{\mathcal{B}}_p = \boldsymbol{I}_{J\times J}$ and input signal vector

$$\boldsymbol{u}_p = (\boldsymbol{0}_{1\times(J-\sigma)}, \bar{x}\boldsymbol{1}_{1\times\sigma})^\top \tag{3.21}$$

arbitrary initial estimates $\bar{x}$ in $\hat{\boldsymbol{x}}^-_p$ can be obtained:

$$\hat{\boldsymbol{x}}^-_p = \left(\hat{x}_{p-1_{\sigma+1}}, \hat{x}_{p-1_{\sigma+2}}, \dots, \hat{x}_{p-1_{J-\sigma}}, \bar{x}\boldsymbol{1}_{1\times\sigma}\right)^\top \tag{3.22}$$

$\boldsymbol{\mathcal{P}}^+_{p-1}$ is updated during the time update as well by line 2 of Algorithm 1. The first part of the instruction ($\boldsymbol{\mathcal{P}}^-_p \leftarrow \boldsymbol{\mathcal{A}}_p \boldsymbol{\mathcal{P}}^+_{p-1} \boldsymbol{\mathcal{A}}_p{}^\top$) leads to $\boldsymbol{\mathcal{P}}^+_{p_{1,\dots,J-\sigma;1,\dots,J-\sigma}} \leftarrow \boldsymbol{\mathcal{P}}^-_{p-1_{\sigma+1,\dots,J;\sigma+1,\dots,J}}$ and all elements in the rows or columns $J - \sigma + 1, J - \sigma + 2, \dots, J$ of $\boldsymbol{\mathcal{P}}^-_p$ equal zero. Especially zeros on the main diagonal prevent that $\hat{x}_{p_{J-\sigma+1}}, \hat{x}_{p_{J-\sigma+2}}, \ldots, \hat{x}_{p_J}$ become different from the initial value $\bar{x}$.

These zeros in $\mathcal{P}_p^-$ can be replaced with large values using the second part of the instruction, $\mathcal{P}_p^- \leftarrow \mathcal{P}_p^- + \boldsymbol{Q}_p$, with

$$\boldsymbol{Q}_p \in \mathbb{R}^{J\times J} \text{ with } \boldsymbol{Q}_{p_{g;h}} = \begin{cases} \bar{p}, & \text{if } h = g \wedge Q \\ \bar{q}, & \text{if } h = g \wedge \neg Q \\ 0, & \text{otherwise} \end{cases} \tag{3.23}$$

with

$$Q = \begin{cases} h \geq J - \sigma + 1, & \text{if } \sigma \geq 0 \\ h \leq -\sigma, & \text{if } \sigma < 0. \end{cases} \tag{3.24}$$

Figure 3.3 depicts different states of $\mathcal{P}^+$ and $\mathcal{P}^-$, respectively. $d = 3$ and $I = 3$ lead to a 6×6 matrix $\mathcal{P}^+$. The diagonal values of $\mathcal{P}_0^+$ are initialized with a large positive value as depicted in (a). All other elements of $\mathcal{P}_0^+$ are initialized with zeros. After some data points in the second spline interval have been processed, different comparatively small values are in the submatrix $\mathcal{P}^+_{2:5,2:5}$ (b).

After data points that fall into the third spline interval have been taken into account, only the elements in the first row and column of $\mathcal{P}^+$ still have their initialization values (c). If only the first part of the update instruction is executed during a right shift by one element ($\sigma = 1$), the elements in the last row and column of $\mathcal{P}^-$ become zero (d). With the second part of the instruction, these elements can be set to nonzero values. For $\boldsymbol{Q}_{6,6} = \bullet$ and all other elements of $\boldsymbol{Q}$ equal zero, matrix (e) is obtained.

Case 2: Left shift of definition range ($\sigma < 0$)

The new knot vector is

$$\boldsymbol{\kappa}_p \leftarrow \left(\bar{\kappa}_{p_1}, \bar{\kappa}_{p_2}, \ldots, \bar{\kappa}_{p_{-\sigma}}, \kappa_{p-1_1}, \kappa_{p-1_2}, \ldots, \kappa_{p-1_{K+\sigma}}\right) \tag{3.25}$$

and the input signal vector reads

$$\boldsymbol{u}_p = \left(\bar{x}\mathbf{1}_{1\times(-\sigma)}, \mathbf{0}_{1\times(J+\sigma)}\right)^\top . \tag{3.26}$$

Effect of shift operation

The shift operation is the distinguishing feature of RBA compared to the algorithms in [67, 104]. Due to the shift operation, the total number of control points that will be estimated during the application of RBA does neither have to be known in advance nor to be bounded. Using $\boldsymbol{\kappa}_p$ and $\hat{\boldsymbol{x}}^+_p$, the determined B-spline function $f(s)$ can be evaluated at any $s \in [\kappa_{p_{d+1}}, \kappa_{p_{d+I+1}})$.

By shifting the entries in $\boldsymbol{\kappa}_{p-1}$, $\hat{\boldsymbol{x}}^+_{p-1}$ and $\mathcal{P}^+_{p-1}$, the required memory is held constant. In Algorithm 2 $|\sigma|$ is chosen just large enough that the current data point can be taken into account during the measurement update. The reason for this is that the shift operation comes at the cost that parts of an already computed approximation are forgotten unless the values of $\boldsymbol{\kappa}_{p-1}$, $\hat{\boldsymbol{x}}^+_{p-1}$ and $\mathcal{P}^+_{p-1}$ are stored separately from Algorithm 2 before they are removed from the vectors or matrix, respectively.

Measurement update

During the measurement update, the information provided by $(s_p, \boldsymbol{y}_p)$ is used to update $f(s)$. With the covariance matrix of measurement noise $\mathcal{R}_p$, different components of the measurement vector $\boldsymbol{y}_p$ can be weighted relatively to each other. $\mathcal{R}_p \in \mathbb{R}^{V_p \times V_p}$ is a diagonal matrix with positive elements on its diagonal. The smaller an entry $\mathcal{R}_{p_{v;v}}$ is, the greater is the effect of the v-th component of the measurement error $(\boldsymbol{y}_p - \boldsymbol{C}_p \hat{\boldsymbol{x}}^-_p)$ on $\hat{\boldsymbol{x}}^+_p$.

The measurement matrix $\boldsymbol{C}_p$ is a $V_p \times J$ matrix. $y_{p,v}$ is the v-th component of $\boldsymbol{y}_p$ and provides information about $\frac{\partial^r}{\partial s^r} f(s_p)$ with $s_p \in [\kappa_\mu, \kappa_{\mu+1})$ and an $r \in \mathbb{N}_0$. The v-th row of $\boldsymbol{C}_p$ is given by

$$\boldsymbol{C}_{p_{v;1,\dots,J}} = \boldsymbol{c} \tag{3.27}$$

with $\boldsymbol{c}$ from (3.15). According to $\boldsymbol{C}_p$, $(s_p, \boldsymbol{y}_p)$ influences only the estimates $\hat{x}_{p_{\mu-d}}, \hat{x}_{p_{\mu-d+1}}, \dots, \hat{x}_{p_\mu}$. However, other estimates can still be updated by the KF using the information stored in $\mathcal{P}^-_p$. *Adopted from [80].*

3.3.4 Effectiveness of recursive B-spline approximation

□ A numerical experiment in [80] demonstrates the effectiveness of Algorithm 2 in comparison with the corresponding WLS solution. The data set $\{(s_p, \boldsymbol{y}_p)\}_{p=1,2,\ldots,P}$ from Section 3.2 with $P = 5000$ is chosen, whereby

$$s_p = 0.01 + 0.02(p-1), \tag{3.28}$$

$$y_{p,1} = \begin{cases} 20, \text{ if } 30 \leq s_p < 70 \\ 10, \text{ otherwise} \end{cases} \quad \text{and} \tag{3.29}$$

$$y_{p,2} = y_{p,3} = 0 \; \forall p. \tag{3.30}$$

A cubic ($d = 3$) B-spline function $f(s)$ approximates the data set, whereby it is assumed that the measurements $y_{p,2}$ refer to the slope $\frac{\partial}{\partial s} f(s)$ of the B-spline function and the measurements $y_{p,3}$ to the curvature $\frac{\partial^2}{\partial s^2} f(s)$. The reciprocals of the relative weights between the different target criteria are specified by the diagonal measurement covariance matrix $\mathcal{R}_p \in \mathbb{R}^{3\times3}$ with $\mathcal{R}_{p_{1;1}} = 1$, $\mathcal{R}_{p_{2;2}} = 10^{-2}$ and $\mathcal{R}_{p_{3;3}} = 10^{-3}$. The diagonal measurement covariance matrix of WLS $\mathbf{R} \in \mathbb{R}^{3P\times3P}$ has analogous values:

$$\begin{aligned} &\mathbf{R}_{i+1;i+1} = 1, \; \mathbf{R}_{i+2;i+2} = 10^{-2}, \; \mathbf{R}_{i+3;i+3} = 10^{-3}, \\ &i = 3(r-1), \; r = 1, 2, \ldots, P \end{aligned} \tag{3.31}$$

The chosen weighting helps to prevent overshoots and oscillations of $f(s)$ and leads to a B-spline function that smooths the jumps of $y_{p,1}$ in the data set. The parameter values are $\bar{q} = 10^{-12}$, $\bar{x} = 0$ and $\bar{p} = 10^4$.

In order to investigate the effect of the choice of I, four runs of RBA with $I = 1, 3, 7$ and 20, respectively, are performed. $\boldsymbol{\kappa}_0 = (-15, 10, 15, 20)$ is used for $I = 1$, $\boldsymbol{\kappa}_0 = (-15, 10, \ldots, 30)$ for $I = 3$, $\boldsymbol{\kappa}_0 = (-15, 10, \ldots, 50)$ for $I = 7$, and $\boldsymbol{\kappa}_0 = (-15, 10, \ldots, 115)$ for $I = 20$. For $I = 20$ the resulting $\mathcal{D}_0$ comprises all s_p of the data set and therefore no shift operation is needed. For $I = 1, 3$ and 7, RBA has to perform several right shifts by one element in order to be able to process all data points. For each shift operation, an additional knot $\bar{\kappa}_{p_K}$ has to be defined in the vector $\bar{\boldsymbol{\kappa}}_p$. $\bar{\kappa}_{p_K} = 25, 30, \ldots, 115$ is chosen for $I = 1$, $\bar{\kappa}_{p_K} = 35, 40, \ldots, 115$ for $I = 3$ and $\bar{\kappa}_{p_K} = 55, 60, \ldots, 115$ for $I = 7$. For evaluation purposes $\hat{x}_{p-1_1}$ and κ_{p-1_1} are stored separately from RBA before a shift operation is performed.

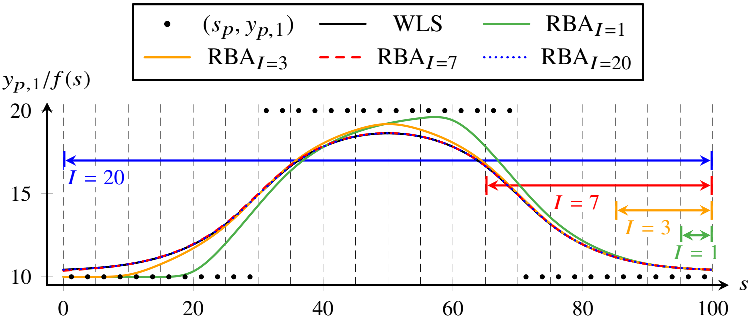


Figure 3.4: Effectiveness of RBA: 40 of the 5000 data points $(s_p, y_{p,1})$ (black dots) and knots $0, 5, \ldots, 100$ (vertical dashed lines) are shown as well as the B-spline function $f(s)$ determined by WLS and RBA with $I = 1, 3, 7$ and 20. Arrows visualize the definition range of $f(s)$ while data points in the interval $[95, 100)$ are processed. *Adopted from [80].*

Figure 3.4 displays the results. As both the data set and knot vector are symmetrical to a vertical straight line through $s = 50$, the WLS solution is symmetrical as well. The RBA solution converges to the WLS solution as I increases. For $I = 1$ and $I = 3$, the resulting B-spline function is asymmetrical with respect to a straight vertical line through $s = 50$. For $I = 7$, RBA provides almost the same result as for $I = 20$. Consequently, in this example I can be reduced from 20 to 7 without noticeably worsening the quality of the approximation. Lowering I leads to less computational effort in the KF because $\mathcal{P}_p^-$, $\mathcal{P}_p^+$ and $\boldsymbol{Q}_p$ are $(d+I)\times(d+I)$ matrices and therefore the asymptotic time complexity of each individual iteration of Algorithm 1 is $\mathcal{O}((d+I)^3)$ if the standard method for matrix multiplication is used [25, 97]. Under the same conditions both RLS and the known methods of other researchers based on the KF need 20×20 matrices because shift operations are not possible.

Due to the large possible savings in computational effort, RBA is also beneficial in offline applications with known finite data set and therefore a definition range that is known in advance.

In further experiments in [80] κ_0 is chosen such that the first data points lie in the rightmost spline interval and runs with $I = 3$, $I = 7$ and $I = 20$ are performed. The deviations of the resulting control point vectors from the those derived when the first data points are in the leftmost interval are close to the machine accuracy indicating that the effect of the shift operation on the approximation result is negligible. *Adopted from [80].*

3.4 Methods for nonlinear weighted least squares problems

Subsection 3.4.1 describes the Levenberg-Marquardt (LM) algorithm followed by the marginalized particle filter in Subsection 3.4.2. Subsection 3.4.3 presents the nonlinear recursive B-spline approximation algorithm for nonlinear weighted least squares problems. Its effectiveness is demonstrated in comparison with the LM solution in Subsection 3.4.4.

3.4.1 Levenberg-Marquardt algorithm

Consider the problem

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \mathbf{e}^\top \mathbf{e} \tag{3.32}$$

with the error function $\mathbf{e}$ given by

$$\mathbf{e} = \sqrt{\mathbf{R}^{-1}} \cdot (\boldsymbol{y} - \boldsymbol{\phi}(\boldsymbol{s}, \boldsymbol{x})) \,. \tag{3.33}$$

$\boldsymbol{s}$ denotes the vector of independent variables, $\boldsymbol{y}$ the vector of measurements and $\mathbf{R}$ the covariance matrix of measurement noise. In (3.33) the difference between $\boldsymbol{y}$ and $\boldsymbol{\phi}$ is weighted with the square root of the reciprocal of the covariance matrix of measurement noise $\mathbf{R}$. $\boldsymbol{\phi}$ is a function whose values depend nonlinearly on its control points which are summarized in the control point vector $\boldsymbol{x}$. Due to this nonlinear relationship (3.32) is a NWLS problem.

The Levenberg-Marquardt (LM) algorithm solves this problem iteratively. $\hat{\boldsymbol{x}}^i$ denotes the known approximation of the solution $\hat{\boldsymbol{x}}$ of (3.32) in iteration i. LM approximates the error function linearly using

$$\boldsymbol{e}(\hat{\boldsymbol{x}}) = \mathbf{e}(\hat{\boldsymbol{x}}^i) + \mathbf{e}'(\hat{\boldsymbol{x}}^i)(\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}^i) \tag{3.34}$$

and updates $\hat{\boldsymbol{x}}^i$ according to

$$\hat{\boldsymbol{x}}^{i+1} = \hat{\boldsymbol{x}}^i + \hat{\delta}^i \tag{3.35}$$

with correction step size δ. The optimal step size $\hat{\delta}$ is derived by solving the linear LS problem

$$\hat{\delta}^i = \arg\min_{\delta^i} \|\mathbf{e}'(\hat{\boldsymbol{x}}^i)\delta + \mathbf{e}(\hat{\boldsymbol{x}}^i)\|_2^2 + \boldsymbol{\lambda}\|\delta^i\|_2^2 \tag{3.36}$$

which includes a damping parameter $\boldsymbol{\lambda} > 0$. The solution $\hat{\delta}^i$ to (3.36) with

$$\hat{\delta}^i = -\left[\mathbf{e}'(\hat{\boldsymbol{x}}^i)^\top \mathbf{e}'(\hat{\boldsymbol{x}}^i) + \boldsymbol{\lambda}^2 \boldsymbol{I}\right]^{-1} \mathbf{e}'(\hat{\boldsymbol{x}}^i)^\top \mathbf{e}(\hat{\boldsymbol{x}}^i) \tag{3.37}$$

is determined by setting the first derivative of its optimization function with respect to δ to zero. A too large δ can be avoided by choosing $\boldsymbol{\lambda}$ appropriately because of

$$\|\delta^i\|_2 \leq \frac{\|\mathbf{e}(\hat{\boldsymbol{x}}^i)\|_2}{\boldsymbol{\lambda}}. \tag{3.38}$$

The LM algorithm terminates if

$$\|\mathbf{e}'(\hat{\boldsymbol{x}}^i)^\top \mathbf{e}(\hat{\boldsymbol{x}}^i)\|_2 \leq \epsilon \tag{3.39}$$

with a specified tolerance ϵ. For convergence $\boldsymbol{\lambda}$ needs to be sufficiently large. However, a large $\boldsymbol{\lambda}$ results in a small correction step size and slow convergence if the solution is still far away from the optimum. Therefore $\boldsymbol{\lambda}$ is controlled using heuristic criteria [35, pp. 222-224].

The computational effort of a single LM iteration roughly equals the effort of the WLS method. The total computational effort of LM depends on the number of performed iterations and is given by $\mathcal{O}(\epsilon^{-2})$ [175].

3.4.2 Marginalized particle filter

□ The marginalized particle filter (MPF) is an iterative algorithm for estimating the unknown state vector $\boldsymbol{x}_p$ of a system at time step $p \in \mathbb{N}$.

In the MPF, $\boldsymbol{x}_p$ is subdivided into $\boldsymbol{x}_p = \left(\left(\boldsymbol{x}_p^L\right)^\top, \left(\boldsymbol{x}_p^N\right)^\top\right)^\top$, whereby the KF optimally estimates the linear state vector $\boldsymbol{x}_p^L$ and a PF estimates the nonlinear state vector $\boldsymbol{x}_p^N$. Exploiting linear substructures allows for better estimates and a reduction of the computational effort. Therefore, the MPF is beneficial for mixed linear/nonlinear state-space models [155]. Due to Equations (3.3) and (3.5), linear substructures will occur in approximation problems as long as there are target criteria that refer to the value of the B-spline function or its derivatives directly.

MPF algorithms for several state-space models can be found in [155] along with a MATLAB example that can be downloaded from [156]. An equivalent but new formulation of the MPF that allows for reused, efficient, and well-studied implementations of standard filtering components is stated in [70].

For a NWLS approximation, the following state-space model derived from [70] is applied:

$$\begin{aligned} \boldsymbol{x}_{p+1}^{N} &= \boldsymbol{\mathcal{A}}_p^{N}\boldsymbol{x}_p^{N} + \boldsymbol{\omega}_p^{N} + \boldsymbol{u}_p^{N} && \text{(Nonlinear state equation)} \qquad (3.40)\\ \boldsymbol{x}_{p+1}^{L} &= \boldsymbol{\mathcal{A}}_p^{L}\boldsymbol{x}_p^{L} + \boldsymbol{\omega}_p^{L} + \boldsymbol{u}_p^{L} && \text{(Linear state equation)} \qquad (3.41)\\ \boldsymbol{y}_p &= \boldsymbol{C}\boldsymbol{x}_p^{L} + c\left(\boldsymbol{x}_p^{N}\right) + \boldsymbol{v}_p && \text{(Measurement equation)} \qquad (3.42) \end{aligned}$$

The superscripts ^{L} and ^{N} indicate that the corresponding quantity refers to linear or nonlinear state variables, respectively. $\boldsymbol{\mathcal{A}}_p$ denotes the state transition matrix, $\boldsymbol{u}_p$ is the known input vector, $\boldsymbol{y}_p$ is the vector of measurements, $\boldsymbol{C}_p$ is the measurement matrix, and c is the nonlinear measurement function that depends on $\boldsymbol{x}_p^N$. $\boldsymbol{\omega}_p^L$ denotes the process noise of the linear state vector with a covariance matrix $\boldsymbol{Q}_p^L$, $\boldsymbol{\omega}_p^N$ is the process noise of the nonlinear state vector with a covariance matrix $\boldsymbol{Q}_p^N$, and $\boldsymbol{v}_p$ is the measurement noise with a covariance matrix $\boldsymbol{\mathcal{R}}_p$. The model of the conditionally linear subsystem in the KF has the state vector $\left(\boldsymbol{\xi}^\top, \left(\boldsymbol{x}^L\right)^\top\right)^\top$, whereby $\boldsymbol{\xi}$ describes the linear dynamics of $\boldsymbol{x}^N$:

$$\begin{aligned} \begin{pmatrix} \boldsymbol{\xi}_{p+1} \\ \boldsymbol{x}_{p+1}^L \end{pmatrix} &= \begin{pmatrix} \boldsymbol{0} & \boldsymbol{\mathcal{A}}_p^N \\ \boldsymbol{0} & \boldsymbol{\mathcal{A}}_p^L \end{pmatrix} \begin{pmatrix} \boldsymbol{\xi}_p \\ \boldsymbol{x}_p^L \end{pmatrix} + \begin{pmatrix} \boldsymbol{u}_p^N \\ \boldsymbol{u}_p^L \end{pmatrix} + \begin{pmatrix} \boldsymbol{\omega}_p^N \\ \boldsymbol{\omega}_p^L \end{pmatrix} \\ \boldsymbol{y}_p &= \begin{pmatrix} \boldsymbol{0} & \boldsymbol{C}_p \end{pmatrix} \begin{pmatrix} \boldsymbol{\xi}_p \\ \boldsymbol{x}_p^L \end{pmatrix} + c\left(\boldsymbol{x}_p^L\right) + \boldsymbol{v}_p \end{aligned} \qquad (3.43)$$

The covariance matrix of process noise is $\begin{pmatrix} \boldsymbol{Q}_p^N & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_p^L \end{pmatrix}$, and $\boldsymbol{0}$ denotes a zero matrix with a suitable size.

A PF with the model

$$\begin{aligned} \boldsymbol{x}_{p+1}^{N} &= \bar{\boldsymbol{\omega}}_{p}^{N} \\ \boldsymbol{y}_{p} &= \bar{\boldsymbol{v}}_{p} \end{aligned} \tag{3.44}$$

deals with the remaining nonlinear effects. The noise depends on the estimates indicated by ^ from the conditionally linear model:

$$\begin{aligned} \bar{\boldsymbol{\omega}}_{p}^{N} &\sim \mathcal{N}\left(\hat{\boldsymbol{\xi}}_{p}, \boldsymbol{\mathcal{P}}_{p}^{\xi,-}\right) \\ \bar{\boldsymbol{v}}_{p} &\sim \mathcal{N}\left(c\left(\boldsymbol{x}_{p}^{N}\right) + \boldsymbol{C}_{p}\left(\boldsymbol{x}_{p}^{N}\right)\hat{\boldsymbol{x}}^{L,-}, \boldsymbol{S}_{p}\right) \end{aligned} \tag{3.45}$$

with

$$\boldsymbol{S}_{p} = \boldsymbol{C}_{p}\boldsymbol{\mathcal{P}}_{p}^{L,-}\boldsymbol{C}_{p}^{\top} + \boldsymbol{\mathcal{R}}_{p} \tag{3.46}$$

where the superscript $^{-}$ refers to a priori quantities that are computed in the time update, which is based on the state of (3.40) and (3.41). In contrast, $^{+}$ denotes a posteriori quantities that are calculated in the following measurement update based on the measurement of (3.42).

$\boldsymbol{\mathcal{P}}_{p}^{L,-}$ and $\boldsymbol{\mathcal{P}}_{p}^{\xi,-}$ are the covariance matrices of the estimation errors that belong to $\hat{\boldsymbol{x}}_{p}^{L}$ and $\hat{\boldsymbol{\xi}}_{p}$, respectively.

The PF uses multiple state estimates, called *particles*, simultaneously. The superscript ^{p} with $p = 1, \ldots, P$ is the particle index and P is the particle count. In general, a KF is used for each particle. In the chosen state-space model, however, $\boldsymbol{\mathcal{A}}_{p}^{L}$, $\boldsymbol{\mathcal{A}}_{p}^{N}$, $\boldsymbol{Q}_{p}^{L}$, and $\boldsymbol{Q}_{p}^{N}$ are independent of $\boldsymbol{x}_{p}^{L}$ and $\boldsymbol{x}_{p}^{N}$. This implies that $\boldsymbol{\mathcal{P}}_{p}^{L,-}$ and $\boldsymbol{\mathcal{P}}_{p}^{\xi,-}$ are identical for all KFs, which reduces the computational effort substantially [70, 155].

Algorithm 3 states the equations for one MPF iteration and was derived from [70, 155]. For an implementation in MATLAB, the example from [156] was adapted. Note that in Algorithm 3 the measurement update of the previous time step $p-1$ occurs before the time update for the current time step p, similar to the algorithm in [6].

In line 4 of Algorithm 3, linear particles are resampled according to their corresponding normalized importance weights. After resampling, particles with a low measurement error occur more frequently in the set of particles. Subsequently, all particles $\hat{\boldsymbol{x}}_{p-1}^{L,+,p}$ are aggregated in line 5 to a single estimate $\hat{\boldsymbol{x}}_{p-1}^{+}$ by calculating their mean.

Algorithm 3: Marginalized particle filter derived from [70, 155]. *Adopted from [78].*

```
Input: $\mathcal{A}_p^L, \mathcal{A}_p^N, \boldsymbol{C}_{p-1}, c, \mathcal{P}_{p-1}^{L,-}, \boldsymbol{Q}_p^L, \boldsymbol{Q}_p^N, \mathcal{R}_{p-1}, \boldsymbol{u}_p^L, \boldsymbol{u}_p^N,$
       $\hat{\boldsymbol{x}}_{p-1}^{L,-,p}, \hat{\boldsymbol{x}}_{p-1}^{N,-,p}, \boldsymbol{y}_{p-1}$
/* 1a PF measurement update */
For $p = 1, \ldots, P$, compute the particle importance weights $q_p^p$ using the likelihood $q_p^p = \mathcal{N}(\hat{\boldsymbol{y}}^p, \boldsymbol{S}_p)$, $\hat{\boldsymbol{y}}^p = \boldsymbol{C}_{p-1}\mathcal{P}_{p-1}^{L,-}\hat{\boldsymbol{x}}_{p-1}^{L,-,p} + c\left(\hat{\boldsymbol{x}}_{p-1}^{N,-,p}\right)$, $\boldsymbol{S}_{p-1} = \boldsymbol{C}_{p-1}\mathcal{P}_{p-1}^{L,-}\boldsymbol{C}_{p-1}^{\top} + \mathcal{R}_{p-1}$ and compute the normalized weights $\tilde{q}_p^p = \frac{q_p^p}{\sum_{p'=1}^{P} q_p^{(p')}}$.
/* 1b KF measurement update */
$\hat{\boldsymbol{x}}_{p-1}^{L,+,p} \leftarrow \hat{\boldsymbol{x}}_{p-1}^{L,-,p} + \mathcal{P}_{p-1}^{L,-}\boldsymbol{C}_{p-1}^{\top}\boldsymbol{S}_{p-1}^{-1}\left(\boldsymbol{y}_{p-1} - \hat{\boldsymbol{y}}^p\right)$
$\mathcal{P}_{p-1}^{L,+} \leftarrow \mathcal{P}_{p-1}^{L,-} - \mathcal{P}_{p-1}^{L,-}\boldsymbol{C}_{p-1}^{\top}\boldsymbol{S}_{p-1}^{-1}\boldsymbol{C}_{p-1}\mathcal{P}_{p-1}^{L,-}$
/* 1c Resampling */
Resample $P$ particles with replacement,
probability $\left(\hat{\boldsymbol{x}}_{p-1}^{L,+,(p')} = \hat{\boldsymbol{x}}_{p-1}^{L,+,p}\right) = \tilde{q}_p^p$.
$\hat{\boldsymbol{x}}_{p-1}^{+} \leftarrow$ mean of $\hat{\boldsymbol{x}}_{p-1}^{L,+,p},\ p = 1, \ldots, P$
/* 2a KF time update */
$\hat{\boldsymbol{x}}_p^{L,-,p} \leftarrow \mathcal{A}_p^L\hat{\boldsymbol{x}}_{p-1}^{L,+,p} + \boldsymbol{u}_p^L$
$\hat{\boldsymbol{\xi}}_p^p \leftarrow \mathcal{A}_p^N\hat{\boldsymbol{x}}_{p-1}^{L,+,p} + \boldsymbol{u}_p^N$
$\mathcal{P}_p^{L,-} \leftarrow \mathcal{A}_p^L\mathcal{P}_{p-1}^{L,+}\left(\mathcal{A}_p^L\right)^{\top} + \boldsymbol{Q}_p^L$
$\mathcal{P}_p^{\xi,-} \leftarrow \mathcal{A}_p^N\mathcal{P}_{p-1}^{L,+}\left(\mathcal{A}_p^N\right)^{\top} + \boldsymbol{Q}_p^N$
$\mathcal{P}_p^{\xi L,-} \leftarrow \mathcal{A}_p^N\mathcal{P}_{p-1}^{L,+}\left(\mathcal{A}_p^L\right)^{\top}$
$\mathcal{P}_p^{L\xi,-} \leftarrow \left(\mathcal{P}_p^{\xi L,-}\right)^{\top}$
/* 2b PF time update */
For $p = 1, \ldots, P$, predict new particles, $\hat{\boldsymbol{x}}_p^{N,-,p} \sim \mathcal{N}\left(\hat{\boldsymbol{\xi}}_p^p, \mathcal{P}_p^{\xi,-}\right)$.
/* 2c Mixing step, update KF */
$\hat{\boldsymbol{x}}_p^{L,-,p} \leftarrow \hat{\boldsymbol{x}}_p^{L,-,p} + \mathcal{P}_p^{L\xi,-}\left(\mathcal{P}_p^{\xi,-}\right)^{-1}\left(\hat{\boldsymbol{x}}_p^{N,-,p} - \hat{\boldsymbol{\xi}}_p^p\right)$
$\mathcal{P}_p^{L,-} \leftarrow \mathcal{P}_p^{L,-} - \mathcal{P}_p^{L\xi,-}\left(\mathcal{P}_p^{\xi,-}\right)^{-1}\mathcal{P}_p^{\xi L,-}$
Output: $\mathcal{P}_p^{L,-}, \hat{\boldsymbol{x}}_{p-1}^{+}, \hat{\boldsymbol{x}}_p^{L,-,p}, \hat{\boldsymbol{x}}_p^{N,-,p}$
```

After both KF and PF have been time updated, the KF is adjusted based on the PF estimates in a mixing step with the cross-covariances of the estimation errors, $\mathcal{P}_p^{\xi L,-}$ and $\mathcal{P}_p^{L\xi,-}$.

In the new formulation from [70], resampling occurs after the measurement update of both PF and KF. Therefore, the quantities computed for the measurement update of the PF can be reused for the KF measurement update. In particular, each particle is only evaluated once in line 1 of each MPF iteration instead of twice as with the previous formulation in [155]. *Adopted from [78].*

3.4.3 Nonlinear recursive B-spline approximation algorithm

□ Nonlinear recursive B-spline approximation (NRBA) adapts a B-spline function $f(s)$ with degree d iteratively to the data set from Section 3.2. Algorithm 4 states the instructions for one iteration of NRBA, which is based on the MPF.

In each iteration p, NRBA modifies f in $I \in \mathbb{N}$ consecutive spline intervals. Each linear particle $\hat{\boldsymbol{x}}_p^{L,p} = (\hat{x}_{p_1}^L, \hat{x}_{p_2}^L, \dots, \hat{x}_{p_J}^L)^\top$ and each nonlinear particle $\hat{\boldsymbol{x}}_p^{N,p} = (\hat{x}_{p_1}^N, \hat{x}_{p_2}^N, \dots, \hat{x}_{p_J}^N)^\top$ contains estimates for $J = d+I$ control points of f. $\boldsymbol{\kappa}_p = (\kappa_{p_1}, \kappa_{p_2}, \dots, \kappa_{p_K})$ denotes the knot vector comprising $K = J + d + 1$ knots. The resulting definition range $\mathcal{D}_p$ of f is given by $\mathcal{D}_p = [\kappa_{p_{d+1}}, \kappa_{p_{J+1}})$. NRBA checks if s_p is in the definition range of the previous time step, $\mathcal{D}_{p-1}$. If not, $\mathcal{D}_{p-1}$ needs to be shifted such that $s_p \in \mathcal{D}_p$. A shift can be conducted in the MPF time update. The result of the time update is the a priori estimate $\hat{\boldsymbol{x}}_p^-$. In the following measurement update, s_p is needed again to compute the measurement matrix $\boldsymbol{C}_p$, and then, to take into account $\boldsymbol{y}_p$. The result of the measurement update is the a posteriori estimate $\hat{\boldsymbol{x}}_p^+$.

Figure 3.5 depicts the allocation of available data points and computed estimates $\hat{\boldsymbol{x}}$ to KF iterations in RBA versus MPF iterations in NRBA. The arrows indicate the needed information for computing the estimates. The KF is initialized with $\hat{\boldsymbol{x}}_0^+$ and conducts in each iteration a time update first and then a measurement update. Therefore, P iterations are required for P data points. In contrast, the MPF performs the measurement update first and is initialized with $\hat{\boldsymbol{x}}_0^-$. Therefore, $\boldsymbol{y}_p$ has to be stored and s_p, s_{p+1}, and $\boldsymbol{y}_p$ have to be provided for iteration $p + 1$. Hence, in order to take into account all data points, an additional iteration compared to the KF is needed. By

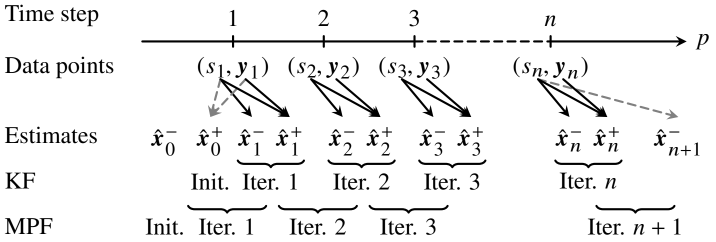

Time step
1
2
3
n
p
Data points
(s_1, y_1) (s_2, y_2) (s_3, y_3) (s_n, y_n)
Estimates
$\hat{x}_0^-$ $\hat{x}_0^+$ $\hat{x}_1^-$ $\hat{x}_1^+$ $\hat{x}_2^-$ $\hat{x}_2^+$ $\hat{x}_3^-$ $\hat{x}_3^+$ $\hat{x}_n^-$ $\hat{x}_n^+$ $\hat{x}_{n+1}^-$
KF
Init. Iter. 1 Iter. 2 Iter. 3 Iter. n
MPF
Init. Iter. 1 Iter. 2 Iter. 3 Iter. n + 1

Figure 3.5: Allocation of availabe data points and computed estimates $\hat{\boldsymbol{x}}$ to KF iterations in RBA versus MPF iterations in NRBA. Arrows indicate the needed information for computing the a priori estimates $\hat{\boldsymbol{x}}^-$ and the a posteriori estimates $\hat{\boldsymbol{x}}^+$. By definition the MPF uses $(s_1, \boldsymbol{y}_1)$ for computing $\hat{\boldsymbol{x}}_0^+$ and s_n for computing $\hat{\boldsymbol{x}}_{n+1}^-$. *Adopted from [78].*

definition, $(s_1, \boldsymbol{y}_1)$ is used for computing $\hat{\boldsymbol{x}}_0^+$ and s_n for $\hat{\boldsymbol{x}}_{n+1}^-$ as indicated by the dashed arrows.

Initialization

Each linear particle $\hat{\boldsymbol{x}}_0^{L,-,p}$ is initialized with $\hat{\boldsymbol{x}}_0^{L,-,p} = \bar{x}^{\text{Init}}\mathbf{1}_{J\times 1}$, and each nonlinear particle $\hat{\boldsymbol{x}}_0^{N,-,p}$ is initialized with

$$\hat{\boldsymbol{x}}_0^{N,-,p} = \bar{x}^{\text{Init}}\mathbf{1}_{J\times 1} + \text{chol}\left(\bar{p}\boldsymbol{I}_{J\times J}\right) \cdot \mathbf{rnd}_{J\times 1}.$$

Hereby, $\mathbf{1}_{J\times 1}$ is a $J \times 1$ matrix of ones and $\bar{x}^{\text{Init}}$ indicates an initial value equal to the scalar measurement $y_{1,v}$ referring to f. chol$(\cdot)$ computes the Cholesky factorization, and $\mathbf{rnd}_{J\times 1}$ is a $J\times 1$ vector of random values drawn from the standard normal distribution. The covariance matrix of a priori estimation error of linear states, $\mathcal{P}^{L,-}$, is initialized with $\mathcal{P}_0^{L,-} = \bar{p}\boldsymbol{I}_{J\times J}$. $\boldsymbol{I}_{J\times J}$ denotes a $J \times J$ identity matrix.

The large scalar value $\bar{p}$ causes $\hat{\boldsymbol{x}}_p$ to quickly change such that f adapts to the data. Provided that the values in $\boldsymbol{Q}_p^L$ are small, the values in $\mathcal{P}_p^{L,-}$ decrease as p grows according to line 8 of Algorithm 3. Small elements in $\mathcal{P}_p^{L,-}$ correspond to certain estimates. Therefore, the particles $\hat{\boldsymbol{x}}_p^{L,-,p}$ and $\hat{\boldsymbol{x}}_p^{N,-,p}$ are slower to be updated using measurements such that f converges. Analogous statements hold for $\mathcal{P}_p^{\xi,-}$ according to line 9 of Algorithm 3.

Algorithm 4: Nonlinear recursive B-spline approximation. *Adopted from [78].*

```
Input: $\boldsymbol{\kappa}_{p-1}, \hat{\boldsymbol{x}}_{p-1}^{L,-,p}, \hat{\boldsymbol{x}}_{p-1}^{N,-,p}, \hat{\boldsymbol{x}}_{p-2}^{+}, \boldsymbol{\mathcal{P}}_{p-1}^{L,-}, \boldsymbol{\mathcal{R}}_{p-1}, s_p, s_{p-1}, \boldsymbol{y}_p, \boldsymbol{y}_{p-1}, \bar{\boldsymbol{\kappa}}_p, \bar{p}, \bar{q}^L, \bar{q}^N, c$
$J \leftarrow$ length $\left(\hat{\boldsymbol{x}}_{p-1}^{L,+,p}\right)$
$K \leftarrow$ length $\left(\boldsymbol{\kappa}_{p-1}\right)$
$d \leftarrow K - J - 1$
$I \leftarrow J - d$
/* Quantities for MPF measurement update */
Compute $\mu$ such that $s_{p-1} \in [\kappa_{p-1_{\mu}}, \kappa_{p-1_{\mu+1}})$
$V_{p-1} \leftarrow$ length$(\boldsymbol{y}_{p-1})$
$\boldsymbol{C}_{p-1} \in \mathbb{R}^{V_{p-1} \times J}$ from (3.47)
/* Quantities for MPF time update */
$\sigma \leftarrow 0$
if $s_p \geq \kappa_{p-1_{J+1}}$ then
    if $s_p \geq \kappa_{p-1_K}$ then
        $\sigma \leftarrow d + 1$
    else
        Compute $\sigma$ such that $s_p \in [\kappa_{p-1_{d+I+1+\sigma}}, \kappa_{p-1_{d+I+2+\sigma}})$
    end
else if $s_p < \kappa_{p-1_{d+1}}$ then
    if $s_p < \kappa_{p-1_1}$ then
        $\sigma \leftarrow -(d + 1)$
    else
        Compute $\sigma$ such that $s_p \in [\kappa_{p-1_{d+1+\sigma}}, \kappa_{p-1_{d+2+\sigma}})$
    end
end
if $\sigma \geq 0$ then
    $\bar{x} \leftarrow$ last element of $\hat{\boldsymbol{x}}_{p-2}^{+}$
    $\boldsymbol{\kappa}_p, \boldsymbol{u}_p^L, \boldsymbol{u}_p^N$ from (3.18), (3.49) and (3.53)
else
    $\bar{x} \leftarrow$ first element of $\hat{\boldsymbol{x}}_{p-2}^{+}$
    $\boldsymbol{\kappa}_p, \boldsymbol{u}_p^L, \boldsymbol{u}_p^N$ from (3.18), (3.55) and (3.56)
end
Compute $\mu$ such that $s_p \in [\kappa_{p_{\mu}}, \kappa_{p_{\mu+1}})$
$\boldsymbol{\mathcal{A}}_p^L, \boldsymbol{Q}_p^L, \boldsymbol{\mathcal{A}}_p^N$ and $\boldsymbol{Q}_p^N$ from (3.48), (3.51), (3.52) and (3.54)
$\left[\boldsymbol{\mathcal{P}}_p^{L,-}, \hat{\boldsymbol{x}}_{p-1}^{+}, \hat{\boldsymbol{x}}_p^{L,-,p}, \hat{\boldsymbol{x}}_p^{N,-,p}\right] \leftarrow$ Algorithm 3 $\Big(\boldsymbol{\mathcal{A}}_p^L, \boldsymbol{\mathcal{A}}_p^N,$
    $\boldsymbol{C}_{p-1}, c, \boldsymbol{\mathcal{P}}_{p-1}^{L,-}, \boldsymbol{Q}_p^L, \boldsymbol{Q}_p^N, \boldsymbol{\mathcal{R}}_{p-1}, \boldsymbol{u}_p^L, \boldsymbol{u}_p^N, \hat{\boldsymbol{x}}_{p-1}^{L,-,p}, \hat{\boldsymbol{x}}_{p-1}^{N,-,p}, \boldsymbol{y}_{p-1}\Big)$
Output: $\boldsymbol{\kappa}_p, \hat{\boldsymbol{x}}_{p-1}^{+}, \hat{\boldsymbol{x}}_p^{L,-,p}, \hat{\boldsymbol{x}}_p^{N,-,p}, \boldsymbol{\mathcal{P}}_p^{L,-}$
```

Hence, the process noises are defined as $\boldsymbol{Q}_p^L = \bar{q}^L \boldsymbol{I}_{J \times J}$ and $\boldsymbol{Q}_p^N = \bar{q}^N \boldsymbol{I}_{J \times J}$ with small positive values $\bar{q}^L$ and $\bar{q}^N$, respectively.

Measurement update

The measurement update from line 1 to line 4 of Algorithm 3 adapts $f(s)$ based on $(s_{p-1}, \boldsymbol{y}_{p-1})$. The v-th dimension of $\boldsymbol{y}_{p-1}$ refers to either f itself or a derivative of f. Therefore, the v-th row of the $V_{p-1} \times J$ measurement matrix $\boldsymbol{C}_{p-1}$ reads

$$\boldsymbol{C}_{p-1_{v;1,\ldots,J}} = \left(\boldsymbol{0}_{1 \times (\mu-(d+1))}, \frac{\partial^r}{\partial s^r} \boldsymbol{b}_{\mu,d}(s_{p-1}), \boldsymbol{0}_{1 \times (J-\mu)} \right), \tag{3.47}$$

whereby $s_{p-1} \in [\kappa_\mu, \kappa_{\mu+1})$ and $r \in \mathbb{N}_0$. Algorithm 4 computes $\boldsymbol{C}_{p-1}$ in line 7 using (3.47).

The value of the nonlinear measurement function c depends on the nonlinear particles $\hat{\boldsymbol{x}}_{p-1}^{N,-,p}$. Furthermore, c can depend on additional quantities that vary with the application and are not stated in Algorithm 3.

The diagonal $V_p \times V_p$ covariance matrix of measurement noise $\mathcal{R}_{p-1}$ enables a relative weighting of the dimensions of $\boldsymbol{y}_{p-1}$ because the influence of the vth dimension of the measurement error $\boldsymbol{e}_p^p = \left(\boldsymbol{y}_{p-1} - \hat{\boldsymbol{y}}^p\right)$ on $\hat{\boldsymbol{x}}_{p-1}^{L,-,p}$ and $\hat{\boldsymbol{x}}_{p-1}^{N,-,p}$ decreases with a growing positive value $\mathcal{R}_{p-1_{v;v}}$.

Time update with shift operation

Based on a comparison between $\boldsymbol{\kappa}_{p-1}$ and s_p, NRBA decides if a shift operation of the B-spline function definition range is required to achieve that $s_p \in \mathcal{D}_p$.

The variable σ calculated from line 8 to line 21 of Algorithm 4 states the shift direction of $\mathcal{D}_{p-1}$ and by how many positions components in $\boldsymbol{\kappa}_{p-1}$, $\hat{\boldsymbol{x}}_{p-1}^{L,-,p}$ and $\hat{\boldsymbol{x}}_{p-1}^{N,-,p}$ need to be moved for that purpose. $\sigma > 0$ indicates a right shift of $\mathcal{D}_{p-1}$, $\sigma < 0$ indicates a left shift, and $\sigma = 0$ means that no shift is conducted because $s_p \in \mathcal{D}_{p-1}$.

Algorithm 4 expects that, for $\sigma > 0$, the $|\sigma|$ additionally needed knots are the σ last entries of the knot vector $\bar{\boldsymbol{\kappa}}_p = (\bar{\kappa}_{p_1}, \bar{\kappa}_{p_2}, \ldots, \bar{\kappa}_{p_K})$ and that they are the $-\sigma$ first entries of $\bar{\boldsymbol{\kappa}}_p$ if $\sigma < 0$.

Case 1: Right shift of definition range ($\sigma \geq 0$)

The new knot vector $\boldsymbol{\kappa}_p$ is given by (3.18) and line 6 of Algorithm 3 updates $\hat{\boldsymbol{x}}_{p-1}^{L,+,p}$ to $\hat{\boldsymbol{x}}_p^{L,-,p}$ using the state transition matrix

$$\mathcal{A}_p^L = \mathcal{A}_p \text{ from (3.19)} \tag{3.48}$$

and the input signal vector

$$\boldsymbol{u}_p^L = \boldsymbol{u}_p \text{ from (3.21).} \tag{3.49}$$

Thereby all entries of $\hat{\boldsymbol{x}}_{p-1}^{L,+,p}$ are moved to the left and the last σ entries of $\hat{\boldsymbol{x}}_p^{L,-,p}$ have an arbitrary initial value $\bar{x}$:

$$\hat{\boldsymbol{x}}_p^{L,-,p} = \left(\hat{x}_{p-1_{\sigma+1}}^L, \hat{x}_{p-1_{\sigma+2}}^L, \ldots, \hat{x}_{p-1_{J-\sigma}}^L, \bar{x}\mathbf{1}_{1\times\sigma}\right)^\top \tag{3.50}$$

During a right shift of the definition range, $\bar{x}$ is set to the last element of $\hat{\boldsymbol{x}}_{p-2}^+$, which is determined during the preceding call of Algorithm 3 in line 5. This is based on the assumption that $\hat{\boldsymbol{x}}_{p-2}^+$ is a good initial value in the magnitude of the data.

Additionally, line 8 of Algorithm 3 updates $\mathcal{P}_{p-1}^{L,+}$ to $\mathcal{P}_p^{L,-}$ using (3.48) and

$$\boldsymbol{Q}_p^L \in \mathbb{R}^{J\times J} \text{ with } \boldsymbol{Q}_{p_{g;h}}^L = \begin{cases} \bar{p}, & \text{if } h = g \wedge Q \\ \bar{q}^L, & \text{if } h = g \wedge \neg Q \\ 0, & \text{otherwise} \end{cases} \tag{3.51}$$

with Q given by (3.24). The update operation moves the elements in $\mathcal{P}_{p-1}^{L,+}$ to the top left and replaces the zeros on the last σ main diagonal elements of $\boldsymbol{Q}_p^L$ with $\bar{p}$ in order to get large values on the last σ main diagonal elements of $\mathcal{P}_p^{L,-}$ and a fast adaption of the initial estimates $\bar{x}$ to the data points.

In line 7 and line 9, Algorithm 3 computes the quantities $\hat{\boldsymbol{\xi}}_p^p$ and $\mathcal{P}_p^{\xi,-}$ that are needed for the PF time update. The calculations of the state transition matrix $\mathcal{A}^N$ with

$$\mathcal{A}_p^N = \mathcal{A}_p \text{ from (3.19)} \tag{3.52}$$

and the input signal vector $\boldsymbol{u}^N$ with

$$\boldsymbol{u}_p^N = \boldsymbol{u}_p \text{ from (3.21)} \tag{3.53}$$

are analogous to those for the linear quantities. $\boldsymbol{Q}^N$ uses $\bar{q}^N$ instead of $\bar{q}^L$:

$$\boldsymbol{Q}_p^N \in \mathbb{R}^{J\times J} \text{ with } \boldsymbol{Q}^N_{p_{g;h}} = \begin{cases} \bar{p}, & \text{if } h = g \wedge Q \\ \bar{q}^N, & \text{if } h = g \wedge \neg Q \\ 0, & \text{otherwise.} \end{cases} \tag{3.54}$$

Case 2: Left shift of definition range ($\sigma < 0$)

The updated knot vector is given by (3.25), the input signal vector for linear states $\boldsymbol{u}^L$ reads

$$\boldsymbol{u}_p^L = \boldsymbol{u}_p \text{ from (3.26)} \tag{3.55}$$

and the input signal vector for nonlinear states $\boldsymbol{u}^N$ is given by

$$\boldsymbol{u}_p^N = \boldsymbol{u}_p \text{ from (3.26).} \tag{3.56}$$

Additionally, $\bar{x}$ is set to the first component of $\hat{\boldsymbol{x}}^+_{p-2}$.

Since $\mathcal{A}_p^L$ and $\mathcal{A}_p^N$ are identical in the chosen state-space model, computational effort can be saved when calculating the covariances and cross-covariances from line 8 to line 11 in Algorithm 3. *Adopted from [78].*

3.4.4 Effectiveness of nonlinear recursive B-spline approximation

□ Numerical experiments conducted in [78] demonstrate the effectiveness of Algorithm 4 in comparison with solutions determined using the LM [115] algorithm. Therein effects of the number of simultaneously adaptable spline intervals and the particle count on the NRBA solution are also investigated. A MATLAB implementation of the experiments is provided in [76].

General experimental setup

The data set $\{(s_p, \boldsymbol{y}_p)\}_{p=1,2,\dots,P}$ is defined according to Section 3.2, whereby

$$\begin{aligned} s_p &= 0.25 + 0.5 \cdot (p-1), \\ y_{p,1} &= \begin{cases} 40, \text{ if } 80 \leq s_p < 120 \\ 30, \text{ otherwise} \end{cases} \\ y_{p,2} &= y_{p,3} = y_{p,4} = 0 \; \forall p \\ P &= 400. \end{aligned} \tag{3.57}$$

A B-spline function $f(s)$ with knot vector $\boldsymbol{\kappa} = (-30, -20, \dots, 230)$ and degree $d = 3$ approximates the data. Thereby, it is supposed that $y_{p,1}$ refers to f, $y_{p,2}$ to the first derivative of f, $y_{p,3}$ to the second derivative of f, and $y_{p,4}$ to the value of the nonlinear measurement function c, which is defined as a quadratic B-spline function with $\boldsymbol{\kappa} = (-5, 0, \dots, 70)$ and $\boldsymbol{x} = (0, 0, 0, 0.25, 1.5, 5, 5, 0, 0, 6, 8, 8, 8)^\top$.

c depends on the value of the approximating function $f(s)$ and is displayed in Figure 3.6. The input variable $f(s)$ of c is restricted to the definition range $[5, 60]$ of c.

The diagonal measurement covariance matrix $\mathcal{R}_p \in \mathbb{R}^{4\times4}$ with $\mathcal{R}_{p_{1;1}} = 1$, $\mathcal{R}_{p_{2;2}} = 5 \cdot 10^{-2}$, $\mathcal{R}_{p_{3;3}} = 5 \cdot 10^{-3}$ and $\mathcal{R}_{p_{4;4}} = 0.8$ or 10^6, respectively, comprises the reciprocal weights of $y_{p,1}$, $y_{p,2}$, $y_{p,3}$ and $y_{p,4}$. The reciprocal weight values for the first three dimensions of $\boldsymbol{y}_p$ avoid that f oscillates and cause that f smooths the jumps in the first dimension of the measurements. With $\mathcal{R}_{p_{4;4}} = 0.8$, the nonlinear target criterion $c\,(f(s)) = 0$ is weighted strongly, whereas it is almost completely neglected with $\mathcal{R}_{p_{4;4}} = 10^6$.

Depending on the applied algorithm, solutions for the former weighting case are denoted by NRBA$^\mathrm{N}$ or LM$^\mathrm{N}$ indicating the nonlinear problem. Conversely, solutions for the latter case of a quasi-linear approximation problem are denoted by NRBA$^\mathrm{L}$ or LM$^\mathrm{L}$.

Solutions for two different numbers of spline intervals I are analysed. For $I = 1$, $\boldsymbol{\kappa}$ is initialized with $\boldsymbol{\kappa}_0 = (-30, 20, \dots, 40)$ which leads to an initial definition range $[0, 10)$ of $f(s)$. For $I = 3$, $\boldsymbol{\kappa}$ is initialized with $\boldsymbol{\kappa}_0 = (-30, 20, \dots, 60)$ and the resulting definition range is $[0, 30)$. In both cases NRBA approximates the data by repeatedly shifting the function definition range to the right. Each time, an additional knot value $\bar{\kappa}_{p_K}$ needs to be provided in the vector $\bar{\boldsymbol{\kappa}}_p$. For $I = 1$ these values are $\bar{\kappa}_{p_K} = 50, 60, \dots, 230$

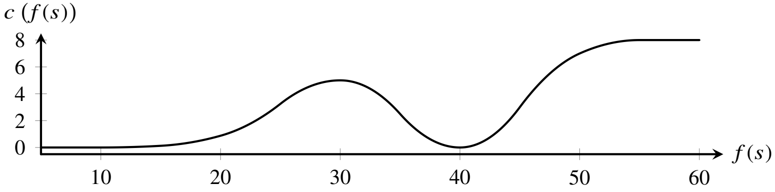


Figure 3.6: Nonlinear measurement function $c\left(f(s)\right)$ depending on the value of the B-spline function $f(s)$ approximating the data. *Adopted from [78].*

and for $I = 3$ they read $\bar{\kappa}_{p_K} = 70, 80, \ldots, 230$. In order to display the NRBA results for the whole data set, all values that are moved out of NRBA matrices and vectors are stored elsewhere.

The remaining NRBA parameters are $\bar{q}^L = 0.005$, $\bar{q}^N = 0.25$ and $\bar{p} = 30$. The LM algorithm uses $\bar{x}^{\text{Init}} = 30$ as the initial value for each control point.

Due to the included PF, NRBA is a nondeterministic method. For each setting, 50 runs are performed and for each run the normalized root mean square error (NRMSE) between the B-spline function determined by NRBA, f_{NRBA}, and the B-spline function according to LM, f_{LM}, is calculated as follows:

$$\begin{aligned}\text{NRMSE} =& \frac{1}{\max_{p=1,\ldots,P}\{f_{\text{LM}}(s_p)\} - \min_{p=1,\ldots,P}\{f_{\text{LM}}(s_p)\}} \\ &\cdot \sqrt{\frac{\sum_{p=1}^{P}\left(f_{\text{NRBA}}(s_p) - f_{\text{LM}}(s_p)\right)^2}{P}}\end{aligned} \tag{3.58}$$

The terms NRBA$^{\text{med}}$ and NRBA$^{\text{max}}$ refer to the NRBA solution with the median or maximum NRMSE, respectively, in each set of 50 runs.

Results

Figure 3.7 shows for both the quasi-linear (L) and the nonlinear (N) problem the approximating functions NRBA$^{\text{med}}$ and NRBA$^{\text{max}}$ compared to the LM solutions. Black dots depict the first component $y_{p,1}$ of 40 of the 400 data points $(s_p, \boldsymbol{y}_p)$. Dashed vertical lines indicate knots, whereby the first and last knots are not shown.

Figure 3.7a shows NRBA approximations for $I = 1$ and $P = 6561 = 9^4$. In this case the MPF state vector comprises four linear and four nonlinear components and the PF creates nine samples per nonlinear state dimension.

At $f(s) = 30$, the deviation between the value of c and its target value $y_{p,4} = 0$ has a local maximum (c.f. Figure 3.6). The nonlinear problem penalizes this deviation strongly; hence, NRBA$^{\mathrm{N}}$ and LM$^{\mathrm{N}}$ avoid $f(s) = 30$. In contrast, NRBA$^{\mathrm{L}}$ and LM$^{\mathrm{L}}$ approximate data with $y_{p,1} = 30$ closely.

Data and knot vector are symmetrical to the straight line given by $s = 100$. Since the LM algorithm processes all data simultaneously in each iteration, the solutions LM$^{\mathrm{L}}$ and LM$^{\mathrm{N}}$ in Figure 3.7a reflect this symmetry.

In contrast, NRBA processes the data from left to right and can only adapt some control points at a time. For $I = 1$, these are the four control points that influence the B-spline function in the spline interval in which the current data point lies and for $I = 3$ additionally the two control points that affect the two spline intervals to the left.

NRBA$^{\mathrm{L}}$ and NRBA$^{\mathrm{N}}$ are both asymmetrical and mostly delayed with respect to LM$^{\mathrm{L}}$ and LM$^{\mathrm{N}}$. For NRBA$^{\mathrm{N}}$, the asymmetry is less distinct because the PF removes states that translate to a large delay more quickly from the particle set because they create a larger error. Additionally, the range of values in NRBA$^{\mathrm{N}}$ is smaller than in NRBA$^{\mathrm{L}}$ so that a present lag is less obvious.

For the same weighting NRBA$^{\mathrm{med}}$ and NRBA$^{\mathrm{max}}$ differ only slightly, which suggests that, for the given setup, $P = 6561$ suffices for a convergence of NRBA solutions.

With RBA for linear weighted least squares approximation similar numerical experiments but without any nonlinear approximation criterion were performed in Subsection 3.3.4 and [80]. For $I = 1$, a strong asymmetry and delay are observed with RBA, analogous to NRBA$^{\mathrm{L}}$ in Figure 3.7a. The delay decreased as I was increased because the filter was able to update more control point estimates with hindsight based on $\mathcal{P}^{L,+}$.

By increasing I to three with NRBA, the dimension of the state space also increases to six linear and six nonlinear components. The PF in NRBA samples the state space less densely, unless the particle count is increased exponentially with I.

Figure 3.7b displays the results for the quasi-linear approximation problem for the case of keeping the sampling density per nonlinear state space dimension constant by choosing $P = 625 = 5^4$ for $I = 1$ and $P = 15625 = 5^6$ for $I = 3$. Overall the NRBA solution is more symmetrical with $I = 3$ but

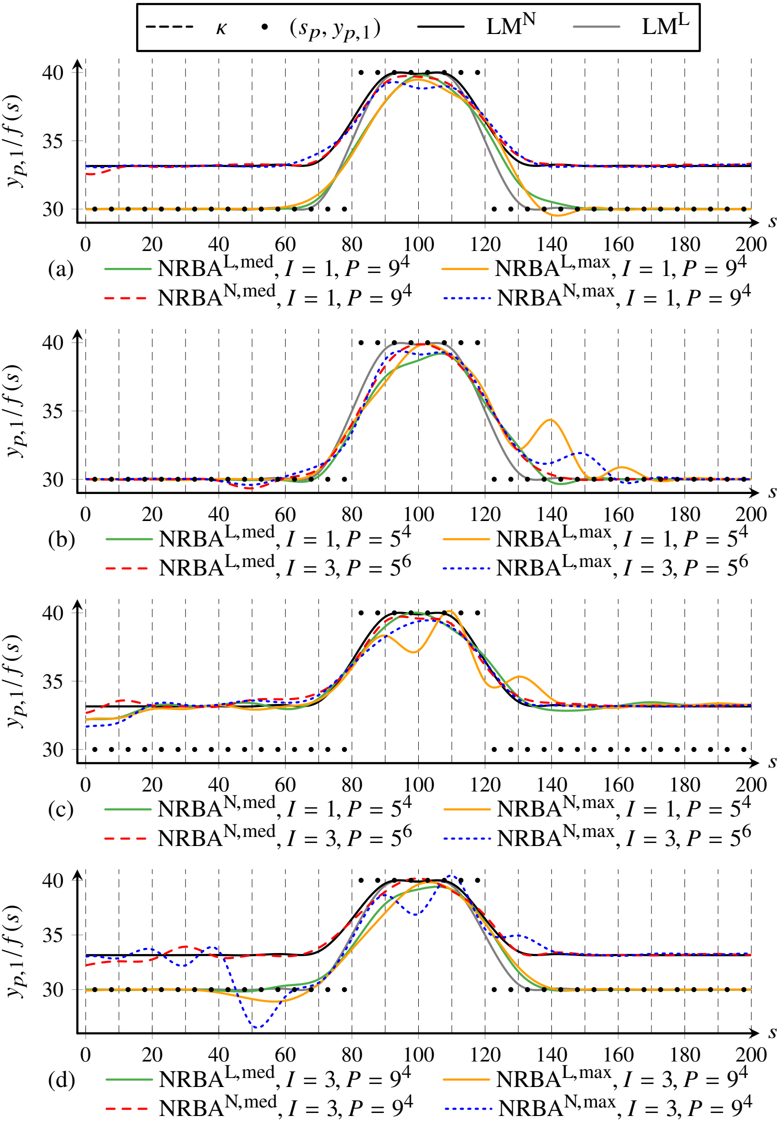


Figure 3.7: B-spline functions f with NRBA for interval counts I and particle counts P and with median (med) and maximum (max) NRMSE compared to the LM solution for quasi-linear (L) and nonlinear (N) problems. Only 40 data points $(s_p, y_{p,1})$ and a subset of knots κ is shown. *Adopted from [78].*

a comparison of NRBA^{med} for $I = 1$ and $I = 3$ indicates that the delay for $s \geq 120$ is not reduced. Instead, the ability to adapt more control point estimates simultaneously sometimes leads to undesirable results, e.g. the too low course between $s = 40$ and $s = 60$ as well as the overcompensation of the delay between $s = 60$ and $s = 75$.

For $I = 1$, NRBA^{max} differs more from NRBA^{med} and shows larger oscillation amplitudes than for $I = 3$. This suggests that $P = 625$ is not sufficient for a convergence of NRBA for $I = 1$. However, even with 625 particles for $I = 1$, the required increase to $P = 15625$ for $I = 3$ is already quite strong. Keeping the sampling density constant quickly becomes infeasible, especially if computation time constraints are present [70].

All other factors held constant Figure 3.7c shows the results for the nonlinear approximation problem, which support the previously drawn conclusions. Additionally, the conflicting target criteria in the nonlinear approximation problem cause a larger stabilization period at $s \leq 20$.

Figure 3.7d depicts the effect of increasing I from one to three while maintaining the particle count of Figure 3.7a. For $I = 3$ NRBA^{max} differ much more from the corresponding NRBA^{med}. This indicates that more particles are needed for convergence for $I = 3$. Furthermore, for $I = 3$ these differences are much larger for NRBA^{N} than for NRBA^{L}.

A comparison between corresponding NRBA^{med} solutions in both figures shows only a small approximation improvement from increasing I for the chosen setup. In Figure 3.7d NRBA^{N} temporarily decreases below $f(s) = 30$, the position of the maximum of c (c.f. Figure 3.6). This is a case in which NRBA chooses a locally but not globally good solution. More detailed investigations on the effects of particle count P on convergence can be found in the cited prepublication. *Adopted from [78].*

3.5 Scientific contribution

This chapter presented two novel algorithms for the weighted least squares (WLS) approximation of a set of data points with a B-spline function. These are the algorithm recursive B-spline approximation (RBA) for the case of a linear WLS problem and the algorithm nonlinear recursive B-spline approximation (NRBA) for the case of a nonlinear WLS problem.

Figure 3.8 subdivides the unconstrained WLS B-spline approximation problem and illustrates the problem types to which RBA, NRBA and the well-known algorithms WLS, RLS, KF and LM can be applied.

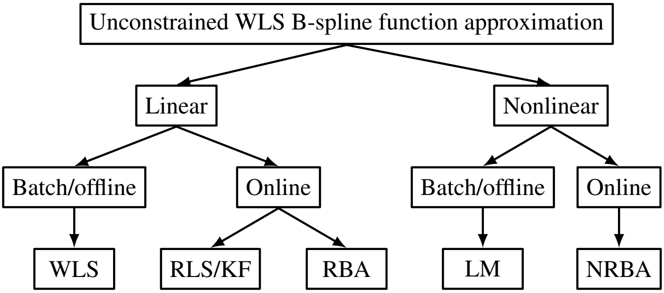


Figure 3.8: Classification of unconstrained weighted least squares (WLS) B-spline approximation problems and suitable algorithms.

Table 3.1: Comparison of B-spline approximation methods. *: Only applies to WLS. **: Also applies to LM if the sliding window method in [38] is used with a window size very small compared to P. *Adopted from [80].*

Feature	WLS/LM (single call)	WLS/LM (multiple calls)	RLS/KF	RBA/NRBA
Number of processable data points P	Bounded	Unbounded	Unbounded	Unbounded
Time complexity	$O(P)^*$	$O(P)^{**}$	$O(P)$	$O(P)$
Approximation interval	Fixed	Variable	Fixed	Variable
Determination of total number of control points being estimated	At beginning	During run-time	At beginning	During run-time

Table 3.1 compares features of the mentioned approximation methods. With LM the computational effort in each iteration depends on the number of data points P. However, the number of iterations performed depends on the specified tolerance and cannot be deduced solely from P. Therefore the time complexity statement in Table 3.1 does not apply to LM in the general case. However, the sliding window LM approach in [38] has time compexity $O(P)$ if the window size is very small compared to P.

The design of RBA, NRBA in this chapter as well as their applications in the following chapters are limited to the B-spline representation and functions. However, the limitation to functions and B-splines in this work did not lead to a loss of generality and the approach of RBA and NRBA can be generalized to curves and surfaces of all spline representations that offer local control, so that the benefits of RBA and NRBA stated in the following can be directly transferred to them as well.

This is because according to Section 2.3, curves and surfaces are direct extensions of functions. Moreover, spline representations with local control only differ in the geometry matrix $\boldsymbol{M}$ and therefore in the blending functions, but they all share the same structure of (2.17) for curves and of (2.19) for surfaces. Their local control results from the bounded interval, in which a blending function is nonzero. As a result, other local spline representations suffer from the same identified research gap.

Recursive B-spline approximation

□ The RBA algorithm prepublished in [80] solves a linear WLS approximation problem iteratively using a KF which estimates the control points of the B-spline function sequentially. Therefore the total computational effort increases linearly with the number of approximated data points.

The main contribution is to use the time update of the KF for a shift of estimated B-spline control points in the KF state vector in combination with a shift in the B-spline knot vector. The shift operation enables to shift the definition range such that it is always possible to take into account the latest data point for the approximation.

Thereby RBA overcomes the limitation of other recursive algorithms based on the KF that can only approximate data points within the initially chosen and fixed approximation interval.

RBA is especially advantageous in online applications in which the magnitude of the data points is not exactly known or changes over time because then data points outside the initially chosen bounded B-spline function definition range can occur.

RBA is also beneficial when a tradeoff between low computational effort and high approximation quality is needed because the shift operation of RBA allows to reduce the size of the KF state vector in both online and offline applications. A smaller state vector causes less computational effort.

Numerical experiments in [80] and Subsection 3.3.4 show that the RBA result converges to the WLS solution as the size of the state vector is increased. Additionally the experimental results reveal that few simultaneously adaptable spline intervals suffice for good approximation results.

The number of required shift operations can increase when the size of the state vector is reduced to lower the computational effort. The experiments indicate that shift operations influence the control point values of the resulting approximation function only in a magnitude that is close to numerical accuracy. Each shift operation comes at the expense that a part of the approximation result is forgotten in order to keep the sizes of matrices and vectors constant. A growing approximation interval can be realized by storing matrix and vector elements separately from RBA before they are overwritten. *Adopted from [80].*

Nonlinear recursive B-spline approximation

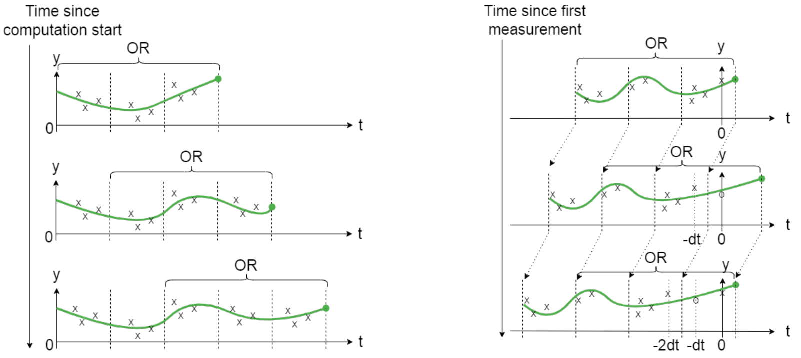
□ The NRBA algorithm prepublished in [78] is a generalization of RBA for NWLS problems which result from target criteria that depend on the control points nonlinearly.

NRBA determines a B-spline function such that it approximates an unbounded number of data points with respect to both linear and nonlinear target criteria. The approach uses a MPF for solving the approximation problem iteratively.

In the MPF, a PF takes into account target criteria that do not relate to the control points in a linear fashion whereas a KF solves any linear subproblem optimally for each particle [70]. As the values of the B-spline function and its derivatives depend linearly on the control point values, linear target criteria will occur in most approximation applications.

The MPF can take into account the exactly known values of the B-spline basis functions and does not need to estimate them like most other nonlinear filters do. Taking advantage of the linear substructure of the problem allows to reduce computational effort and achieve better results compared to purely nonlinear filters like a PF [155].

The features and benefits of RBA that result from the shift operation also apply to NRBA. Numerical experiments in [78] and Subsection 3.4.4 investigated the effectiveness of the approach in comparison to the LM algorithm and illustrated the effects of selected NRBA parameter values on the approximation result. Provided that the parameters are chosen appropriately, the solution of the proposed method is close to the LM solution apart from a slight filter-typical delay.



(a) Trajectory optimization. Planned trajectory time *t*.

(b) Signal representation. Real time *t* measured from present point in time, interval *dt* between consecutive measurements.

Figure 3.9: Exemplary use cases for different operating modes of RBA and NRBA. Optimization range (OR).

NRBA use cases are NWLS problems in which a linearization of nonlinear criteria is not desired or promising, for example because of distinct nonlinearities.

For linear WLS problems RBA should be used instead of NRBA. RBA is based on the KF, which computes an optimal solution [205]. For linear problems NRBA can at best reach the same approximation quality provided that the particle count is large enough, which requires more computational effort.

Furthermore, with NRBA the approximation depends more strongly on the parameterization of the underlying filter algorithm than with RBA.

Increasing the number of control points that NRBA can adapt simultaneously is not as unambiguously beneficial for the approximation result as with RBA. Moreover, with NRBA the increase of control point count usually needs to be combined with an exponential increase of the particle count in the PF for an improvement of the approximation. This is known as curse of dimensionality and a general problem of sampling-based nonlinear filters. *Adopted from [78].*

Use cases in terms of operating mode

Use cases of RBA and NRBA can be classified according to whether the knot vector changes only with a shift operation or with each iteration:

- Knots change only with each shift operation: This mode assumes that the independent variable will not take values large enough to create an arithmetic overflow during operation. Therefore the knot vector changes only with a shift operation. Figure 3.9a illustrates this mode at the example of trajectory optimization, whereby the independent variable t refers to the planned trajectory time. From the top plot to the bottom plot the computation time progresses and the number of processed data points increases. In the example, t grows with the data, but a limited planning horizon, that does not lead to arithmetic overflows, can be assumed. Ideally, t increases much faster than real time so that there is no need to evaluate the trajectory in the optimization range (OR), in which it can change. Chapter 5 will investigate this use case in more detail. The examples in Subsection 3.3.4 and Subsection 3.4.4 also belong to this operating mode.

- Knots change with each iteration: This operating mode is required in case of an unbounded independent variable, that eventually will take a value that causes an arithmetic overflow. Figure 3.9b illustrates this mode at the example of an analytic representation of a signal or its derivative or its integral over the recent past up to the present point in time. From the top plot to the bottom plot a new measurement is received with each plot. If the independent variable t were assigned to the observation number, eventually an overflow in the knot vector would be encountered. Let dt denote the time interval between the two latest measurements. For simplicity, let dt be constant. An overflow can be avoided by subtracting dt from the knot vector after each iteration. Thereby the latest measurement is always assigned to $t = 0$ and the knots are continuously shifted to the left in Figure 3.9b as their values keep decreasing until they are removed from the knot vector and the OR of RBA or NRBA. The signal representation application requires to evaluate the function within the OR unless a time delay of the magnitude of the OR is acceptable and a some knots and control points can be saved when they leave the OR. If so, the function can be evaluated outside of the OR and further into the past.

Differentation from previous works

In the previous project *e-generation* [22, pp. 29-35] an iterative data approximation by a functional representation was developed as well and was patented by F. Bleimund and S. Rhode [21]. This previous approach has in common with RBA that it also solves the linear WLS optimization problem (3.13) iteratively, whereby the target criteria refer to the value of the functional representation and its first two derivatives. Moreover, the optimization problem is solved iteratively using a KF and a shift matrix.

However, in the previous project, a polynomial as in (2.8) is adapted to the data. This polynomial exhibits the features discussed in Section 2.3 along with its flaws compared to a spline representation, such as coupling between degrees of freedom and polynomial degree, risk of oscillations from the Runge's phenomenon, global effect of coefficient value changes and geometrically difficult to interpret coefficients.

Due to the global control of the polynomial, a shift operation and therefore the KF would not be needed if the approximation were performed analogously to the case depicted by Figure 3.9a, in which the value of the independent variable can take large values. For this case, RLS would suffice.

However, the approach assigns the latest oberserved data point always to the same value of the independent variable, analogous to Figure 3.9b. This is achieved by defining the shift operation via an upper triangular matrix such that a coefficient vector is calculated for a polynomial that is shifted by dt to the left with respect to the current polynomial during the time update of the KF. This shift operation for the polynomial corresponds to the subtraction of dt from the spline knot vector as described above for the second operating mode.

In contrast, the shifting operation stated in the previous sections of this work is needed in general for both of the above described operating modes. It modifies both the knot vector and the control point vector and occurs depending on the knot vector and the independent variable of the latest data point.

The algorithms presented in this work are superior to the one developed in *e-generation* in that they allow to benefit from the advantages of spline representations with local control during an iterative data approximation. The NWLS problem adressed by NRBA was not considered in the previous project.

4 Models of research vehicle and reference route

This chapter describes models of the research vehicle and the reference route. The content of this chapter and the previous one will then be combined for trajectory optimization in Chapter 5 and ALC in Chapter 6.

The research vehicle is an all-wheel-driven BEV based on the Porsche Boxster (type 981) which was developed during the research project *e-generation*. Further information about the vehicle is provided in the remainder of this chapter as well as in [14, 16, 207].

Detailed nonlinear models allow a representation of the dynamic vehicle behavior that is close to reality. Such models take into account the kinematics of the vehicle and of its subsystems like the wheel suspension. They also comprise models of tire forces. However, a model should also focus on the aspects that are important for the actual application and simplify where possible. Depending on the desired level of detail and the application, different vehicle models are known in literature [157, pp. 5-6].

The purpose of the vehicle models in this work is to investigate the energy consumption of the research vehicle under ALC on a reference route. According to [187, p. 28] with a normally experienced driver the absolute value of the lateral acceleration is below 3.5 $\mathrm{m/s^2}$ and the longitudinal acceleration ranges roughly from $-2.4\ \mathrm{m/s^2}$ to $1.8\ \mathrm{m/s^2}$. For comfort and acceptance reasons, an ALC should not exceed these limits most of the time, therefore models for low dynamics suffice. Simplifying assumptions of this work for longitudinal dynamics include:

- Lifting, rolling and pitching motions of the vehicle are neglected.
- The vehicle mass is concentrated in the center of gravity (COG).
- The vertical wheel force, also known as *wheel load*, is constant.
- Longitudinal tire slip is neglected.

For describing the lateral dynamics the linear single track model is used.

Section 4.1 states the coordinate system. Section 4.2 describes relevant driving resistances and Section 4.3 the power train of the research vehicle. Section 4.4 explains how the resulting energy consumption for a given route can be derived and reasons the approach for optimization of energy con-

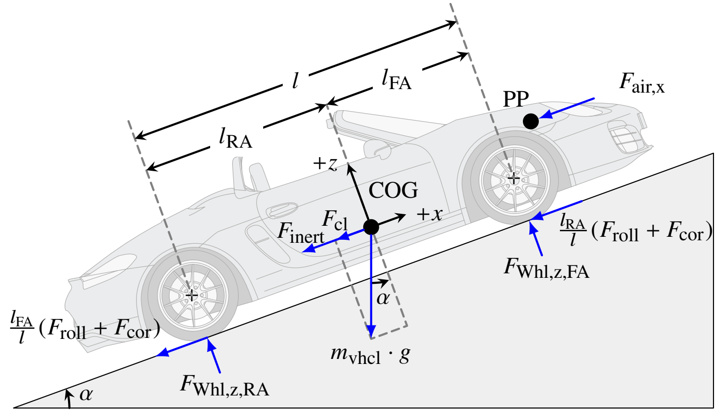


Figure 4.1: Lateral view of research vehicle at slope of angle α with acting climbing force F_{cl}, rolling resistance F_{roll}, cornering resistance F_{cor}, air resistance F_{air} and inertial force F_{inert}. Vehicle drawing from [46].

sumption that the ALC takes. Section 4.5 describes vehicle models for ALC tests in a simulation environment and for investigations of the resulting energy consumption of the research vehicle. The reference route is mentioned in Section 4.6 and Section 4.7 derives two simplified adaptive vehicle models for use within the ALC.

4.1 Coordinate system

Several coordinate systems are used to describe vehicle position and orientation [157, pp. 17-31]. This work uses a body-fixed right-hand vehicle coordinate system with origin in the vehicle COG and axes x, y and z. The axes are perpendicular to each other and shown in Figure 4.1 and Figure 4.2. The x axis is the vehicle longitudinal axis and points towards the vehicle front, whereas the y axis is the vehicle lateral axis that points towards the left vehicle side. Since rolling and pitching movements of the vehicle with respect to the road surface are neglected, the plane that x and y axes span is parallel to the road surface. The z axis is the vertical axis that points upwards and is perpendicular to the road surface.

4.2 Driving resistances

This section states the main forces acting on a vehicle and the force equilibrium that they form at the wheels. Figure 4.1 and Figure 4.2 illustrate these forces.

4.2.1 Climbing force

□ The climbing force F_{cl} results from the gravitational force acting on the COG and is given by

$$F_{\mathrm{cl}} = m_{\mathrm{vhcl}} \cdot g \cdot \sin(\alpha). \tag{4.1}$$

F_{cl} depends on the vehicle mass m_{vhcl} and road slope angle α which is measured between the road surface and the plane that is perpendicular to the direction of gravitational force $m_{\mathrm{vhcl}} \cdot g$ with gravitational constant g. *Adopted from [79].* The road slope angle α can be computed from the road slope γ stated in percent using

$$\alpha = \arctan(\gamma/100) \text{ [119, p. 95].} \tag{4.2}$$

4.2.2 Wheel resistance

The wheel resistance consists of several components, of which only rolling resistance and cornering resistance will be considered. Resistance components neglected in this work include the toe-in resistance and flood resistance. The toe-in resistance is caused by not perfectly parallel wheels and the flood resistance occurs on wet roads when the tire evacuates water [119, pp. 11-19]. Moreover, the longitudinal tire slip resistance [165] and the ventilation resistance of the rotating wheel [182] are not considered.

Bearing friction as well as residual braking torque from the hydraulic brakes can also be assigned to wheel resistance but will be considered as resistances of the power train in this work.

Rolling resistance

□ Rolling resistance results from damping forces of the deformed tire rubber. When driving straight on a dry road, the wheel resistance almost entirely comes from the rolling resistance F_{roll} [119, p. 11] with

$$F_{\mathrm{roll}} = f_{\mathrm{r}} \cdot m_{\mathrm{vhcl}} \cdot g \cdot \cos(\alpha). \tag{4.3}$$

Adopted from [79]. The normal force $m_{\text{vhcl}} \cdot g \cdot \cos(\alpha)$ equals the sum of vertical front axle wheel force $F_{\text{Whl,z,FA}}$ and vertical rear axle wheel force $F_{\text{Whl,z,RA}}$. In the static case the distribution of the vertical forces between front axle (FA) and rear axle (RA) can be determined from the wheel base (l), the distance between FA and COG (l_{FA}) and the distance between RA and COG (l_{RA}). The road slope influences the static load distribution between the axles as well but since the distance between COG of the research vehicle and road surface is small, this effect of road slope is neglected.

The rolling resistance coefficient f_{r} depends on the vertical wheel force, tire pressure and slightly increases with velocity [24, pp. 50, 52-53]. For most passenger vehicles f_{r} can be assumed between 0.007 and 0.014 [10].

Cornering resistance

In curves the centrifugal force F_{centr} with

$$F_{\text{centr}} = m_{\text{vhcl}} \cdot \dot{v}_{\text{Vhcl,y}} \tag{4.4}$$

acts on the COG. The lateral acceleration $\dot{v}_{\text{Vhcl,y}}$ is given by

$$\dot{v}_{\text{Vhcl,y}} = v_{\text{Vhcl}}^2 \cdot \kappa \tag{4.5}$$

for small side slip angles and constant velocity [157, p. 228]. The road curvature κ is the reciprocal of the curve radius.

The single track vehicle model represents the fundamental driving dynamics for lateral acceleration smaller than $\dot{v}_{\text{Vhcl,y}} = 4\ \text{m/s}^2$ on dry roads. The model is very idealized and uses the following simplifications:

- Lifting, rolling and pitching motions of the vehicle are neglected.
- The vehicle mass is concentrated in the COG.
- The vehicle velocity is assumed to be roughly constant, i.e. quasi-stationary.
- Wheels on the same axle are represented by a single wheel in the center of the corresponding axle.
- The vertical wheel force is constant.
- Pneumatic trail and aligning torque resulting from the slip angle of the tire are neglected.
- Longitudinal tire forces are neglected [157, pp. 225-226].

In order to stay on the road, the tires need to provide lateral forces that counteract the centrifugal force. These lateral forces are distributed among

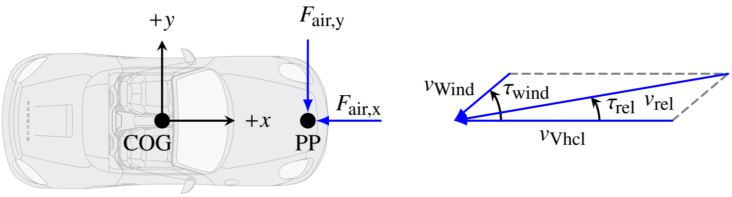


Figure 4.2: Top view of research vehicle with acting air forces. Environmental wind with velocity v_{Wind} and angle τ_{wind} as well as vehicle velocity v_{Vhcl} result in relative air flow with velocity v_{rel} and angle τ_{rel}. This causes longitudinal air resistance $F_{\text{air,x}}$ and lateral air resistance $F_{\text{air,y}}$ acting on the pressure point (PP). COG denotes the center of gravity. Vehicle drawing from [46].

the axles depending on the position of the COG and created using tire slip angles.

These slip angles cause the cornering resistance F_{cor} acting in longitudinal direction. Under the assumption of small road curvature, small steering angle, small slip angles and identical tires, F_{cor} according to the single track model reads

$$F_{\text{cor}} = \frac{\kappa^2 \cdot v_{\text{Vhcl}}^4}{2 \cdot c} \cdot \left[\left(\frac{l_{\text{FA}}}{l} \cdot m_{\text{vhcl}} \right)^2 + \left(\frac{l_{\text{RA}}}{l} \cdot m_{\text{vhcl}} \right)^2 \right] \tag{4.6}$$

whereby c denotes the tire cornering stiffness [64, p. 147].

4.2.3 Air resistance

The air resistance F_{air} acts on the pressure point (PP) of the vehicle and results from the turbulences of air stream and friction of the air flow. The relative air velocity v_{rel} and relative air flow angle τ_{rel} both depend on the vehicle velocity v_{Vhcl}, the environmental wind velocity v_{Wind} and the environmental wind angle τ_{wind} as Figure 4.2 illustrates:

$$\begin{aligned} v_{\text{rel}} &= \sqrt{v_{\text{Vhcl}}^2 + v_{\text{Wind}}^2 + 2 \cdot v_{\text{Vhcl}} \cdot v_{\text{Wind}} \cdot \cos(\tau_{\text{wind}})} \\ \tau_{\text{rel}} &= \arcsin\left(\frac{v_{\text{Wind}}}{v_{\text{rel}}} \right) \cdot \sin(\tau_{\text{wind}}) \end{aligned} \tag{4.7}$$

The angles are measured anticlockwise starting from negative direction of the x axis. □ The longitudinal air resistance $F_{\mathrm{air,x}}$ is given by

$$F_{\mathrm{air,x}} = c_{\mathrm{x}}\left(\tau_{\mathrm{rel}}\right) A \frac{\rho}{2} v_{\mathrm{rel}}^{2}. \tag{4.8}$$

c_{x} is the dimensionless longitudinal drag coefficient that describes the shape of the vehicle and depends on the relative air flow angle τ_{rel}. A denotes the effective vehicle cross-sectional area perpendicular to the vehicle longitudinal axis and ρ is the air density. *Adopted from [79].* Analogous to the centrifugal force, the lateral air resistance $F_{\mathrm{air,y}}$ needs to be balanced by lateral tire forces. The required slip angles in turn cause a driving resistance component in longitudinal direction.

Neglecting environmental wind leads to $v_{\mathrm{rel}} = v_{\mathrm{Vhcl}}$, $\tau_{\mathrm{rel}} = 0$ and $F_{\mathrm{air,y}} = 0$. $c_{\mathrm{x}}(\tau_{\mathrm{rel}} = 0)$ is also denoted c_{w} [157, pp. 212-214]. For passenger vehicles c_{w} usually lies between 0.20 and 0.40 and A ranges from 1.5 m^2 to 2.5 m^2. With increasing vehicle velocity $F_{\mathrm{air,x}}$ becomes the largest driving resistance component [24, pp. 50-51].

4.2.4 Inertial force

Inertial forces arise when vehicle velocity changes. Analogously variations of the angular velocity of power train components cause inertial torque. Each axle of the research vehicle is driven independently as Figure 4.3 illustrates. Due to the fixed gear ratio of each gear box and the assumption of negligible longitudinal tire slip, angular velocities are proportional to the vehicle velocity. Computing the equivalent mass of the power train allows to summarize both inertial force and inertial torque in the inertial force F_{inert} with respect to the vehicle longitudinal acceleration $\dot{v}_{\mathrm{Vhcl,x}}$ [62, pp. 16-17]:

$$F_{\mathrm{inert}} = \left(m_{\mathrm{vhcl}} + m_{\mathrm{eq},FA} + m_{\mathrm{eq,RA}}\right) \cdot \dot{v}_{\mathrm{Vhcl,x}} \tag{4.9}$$

$m_{\mathrm{eq},FA}$ is the equivalent mass of the power train components that drive the front axle FA and $m_{\mathrm{eq,RA}}$ denotes the equivalent mass of components that drive the rear axle RA:

$$\begin{aligned} m_{\mathrm{eq,FA}} &= \frac{J_{\mathrm{EM,FA}} \cdot i_{\mathrm{G}}^2 + J_{\mathrm{GB}} + 2 \cdot J_{\mathrm{Wheel,FA}} + 2 \cdot J_{\mathrm{Brake,FA}}}{r_{\mathrm{dyn,FA}}^2} \\ m_{\mathrm{eq,RA}} &= \frac{J_{\mathrm{EM,RA}} \cdot i_{\mathrm{G}}^2 + J_{\mathrm{GB}} + 2 \cdot J_{\mathrm{Wheel,RA}} + 2 \cdot J_{\mathrm{Brake,RA}}}{r_{\mathrm{dyn,RA}}^2} \end{aligned} \tag{4.10}$$

J_{EM} is the mass moment of inertia of the corresponding electric motor (EM) that is connected via a gear box (GB) with integrated differential to the wheels. Both gear boxes have the same gear ratio $i_{\mathrm{G}} = 9.59$ [14]. J_{GB} is the mass moment of inertia of a GB, J_{Wheel} the mass moment of inertia of a wheel, J_{Brake} the mass moment of inertia of a brake and r_{dyn} the dynamic wheel radius.

4.2.5 Force equilibrium

□ The sum of the driving resistances from (4.1), (4.3), (4.6), (4.8) and (4.9) equals the traction force F_{trac} between all tires and the road surface:

$$F_{\mathrm{trac}} = F_{\mathrm{cl}} + F_{\mathrm{roll}} + F_{\mathrm{air}} + F_{\mathrm{cor}} + F_{\mathrm{inert}} \tag{4.11}$$

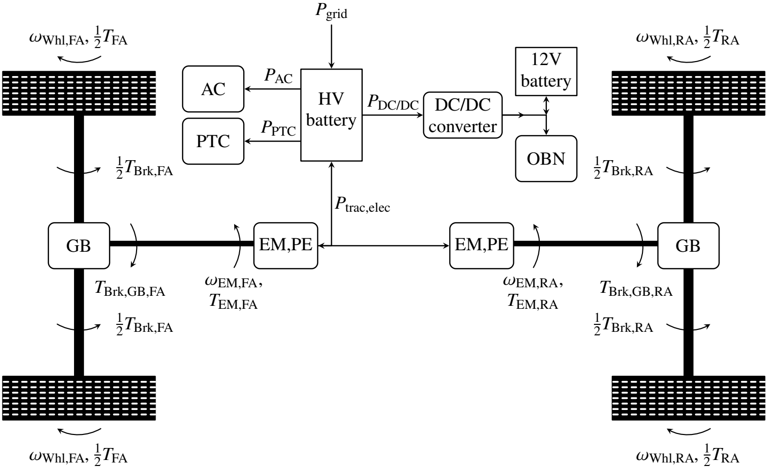
The mechanical traction power $P_{\mathrm{trac,mech}}$ corresponding to F_{trac} reads

$$P_{\mathrm{trac,mech}} = F_{\mathrm{trac}} \cdot v_{\mathrm{Vhcl}}. \tag{4.12}$$

Providing F_{trac} requires longitudinal tire slip. Under the assumption of negligible slip losses, $P_{\mathrm{trac,mech}}$ equals the wheel power P_{wheel} provided by the power train [157, pp. 152-153]. *Adopted from [79].*

4.3 Power train

Figure 4.3 depicts the power train of the research vehicle. In various components of the power train power losses occur. These power losses are measured during component and vehicle tests on test benches and are stored in look-up tables, which have been provided for the research vehicle. Depend-



P_{grid}
$\omega_{\mathrm{Whl,FA}}, \frac{1}{2}T_{\mathrm{FA}}$
$\omega_{\mathrm{Whl,RA}}, \frac{1}{2}T_{\mathrm{RA}}$
AC
P_{AC}
HV battery
$P_{\mathrm{DC/DC}}$
DC/DC converter
12V battery
PTC
P_{PTC}
OBN
$\frac{1}{2}T_{\mathrm{Brk,FA}}$
$\frac{1}{2}T_{\mathrm{Brk,RA}}$
$P_{\mathrm{trac,elec}}$
GB
EM,PE
EM,PE
GB
$T_{\mathrm{Brk,GB,FA}}$
$\omega_{\mathrm{EM,FA}}$, $T_{\mathrm{EM,FA}}$
$\omega_{\mathrm{EM,RA}}$, $T_{\mathrm{EM,RA}}$
$T_{\mathrm{Brk,GB,RA}}$
$\frac{1}{2}T_{\mathrm{Brk,FA}}$
$\frac{1}{2}T_{\mathrm{Brk,RA}}$
$\omega_{\mathrm{Whl,FA}}, \frac{1}{2}T_{\mathrm{FA}}$
$\omega_{\mathrm{Whl,RA}}, \frac{1}{2}T_{\mathrm{RA}}$

Figure 4.3: Model of power train of research vehicle.

ing on the kind of input power, components of the power train can be divided into mechanical components and electrical components.

4.3.1 Mechanical components

The wheel power P_{wheel} provided by the power train reads

$$P_{\mathrm{wheel}} = T_{\mathrm{FA}} \cdot \omega_{\mathrm{Whl,FA}} + T_{\mathrm{RA}} \cdot \omega_{\mathrm{Whl,RA}} \tag{4.13}$$

with front axle torque T_{FA}, rear axle torque T_{RA}, front wheel angular velocity $\omega_{\mathrm{Whl,FA}}$ and rear wheel angular velocity $\omega_{\mathrm{Whl,RA}}$.

The torque distribution between T_{FA} and T_{RA} is computed by the motor electronic control unit (ECU) as a function of F_{trac}. At each axle a gear box (GB) with open differential distributes the torque equally among the left and right wheel.

When driving straight, $\omega_{\mathrm{Whl,FA}}$ and $\omega_{\mathrm{Whl,RA}}$ are given by

$$\omega_{\mathrm{Whl,FA}} = \frac{v_{\mathrm{Vhcl}}}{r_{\mathrm{dyn,FA}}}, \quad \omega_{\mathrm{Whl,RA}} = \frac{v_{\mathrm{Vhcl}}}{r_{\mathrm{dyn,RA}}} \tag{4.14}$$

with dynamic front wheel radius $r_{\text{dyn,FA}}$ and dynamic rear wheel radius $r_{\text{dyn,RA}}$. Between wheel and gear box undesired friction occurs in the wheel bearing, which results in braking torque. There is also residual brake torque caused by permanent residual friction between brake pad and brake disc. The model takes into account the front axle brake torque $T_{\text{Brk,FA}}$ and the rear axle brake torque $T_{\text{Brk,RA}}$. The values are derived by interpolating linearly with respect to the wheel angular velocities in look-up tables.

At each axle a 1-speed gear box with gear ratio i_{G} increases the motor angular velocity and reduces the motor torque. Front axle motor angular velocity $\omega_{\text{EM,FA}}$ and rear axle motor angular velocity $\omega_{\text{EM,RA}}$ are given by

$$\omega_{\text{EM,FA}} = i_{\text{G}} \cdot \omega_{\text{Whl,FA}}, \quad \omega_{\text{EM,RA}} = i_{\text{G}} \cdot \omega_{\text{Whl,RA}}. \tag{4.15}$$

Front axle motor torque $T_{\text{EM,FA}}$ and rear axle motor torque $T_{\text{EM,RA}}$ read

$$\begin{aligned} T_{\text{EM,FA}} &= \frac{T_{\text{FA}}}{i_{\text{G}}} + T_{\text{Brk,GB,FA}}\left(T_{\text{EM,FA}}, \omega_{\text{EM,FA}}\right), \\ T_{\text{EM,RA}} &= \frac{T_{\text{RA}}}{i_{\text{G}}} + T_{\text{Brk,GB,RA}}\left(T_{\text{EM,RA}}, \omega_{\text{EM,RA}}\right), \end{aligned} \tag{4.16}$$

whereby $T_{\text{Brk,GB,FA}}$ and $T_{\text{Brk,GB,RA}}$ denote the braking torques occurring in the gear boxes during the not lossless power conversion. The absolute values of these quantities increase with the absolute values of motor torque and angular velocity and are determined using two-dimensional linear interpolation in look-up tables.

4.3.2 Electrical components

A permanently excited synchronous motor (PSM) with 120 kW drives the front axle and an asynchronous motor (ASM) with 144 kW drives the rear axle. The PSM has a higher efficiency whereas the ASM has lower drag losses. Therefore, mainly the PSM propels the vehicle and is supported by the ASM only in case of high power demands [14]. The motors are connected to the high voltage (HV) battery via power electronics (PE). The losses of both motors including the losses in the power electronics are modelled by two-dimensional linear interpolation with respect to motor torque T_{EM} and motor angular velocity ω_{EM} in look-up tables. Ohmic resistance and magnetic resistance as well as switching losses contribute to the losses of EM and PE. $P_{\text{trac,elec}}$ is the electrical traction power that results from the mechanical traction power $P_{\text{trac,mech}}$ and includes the power losses between power electronics and wheels.

While braking, the motors act as generators and convert mechanical traction power $P_{\text{trac,mech}}$ into electrical traction power $P_{\text{trac,elec}}$ that is recuperated into the HV battery. For driving stability reasons, the majority of the recuperation power comes from the front wheels, which are connected to the more efficient PSM. When the braking power demand exceeds the recuperation capabilites, hydraulic wheel brakes are activated in order to fulfill the demand. Hydraulic wheel brakes convert mechanical traction power into thermal power that cannot be reused for vehicle propulsion anymore. Additional hydraulic braking power is usually needed when braking at high velocity or when high deceleration is requested. Furthermore, during a standstill and at very low speeds hydraulic brakes are solely used.

Due to the powerful drive train, the vehicle has a high recuperation power limit. Furthermore, simulations of the ALC for determining the energy consumption will not include standstills, high dynamic driving maneuvers and driving at high speeds. Therefore this work assumes that the vehicle can brake solely using recuperation, also denoted perfect recuperation by [62, p. 16], and does not consider hydraulic braking power in the vehicle model.

Apart from the power electronics, the motors and the HV battery, a positive temperature coefficient (PTC) battery heating, an air conditioning (AC) compressor and a DC/DC converter for the 12 V circuit belong to the HV system. The on-board network (OBN) is fed via the 12 V battery or directly from the DC/DC converter. The on-board network (OBN) also powers a thermoelectric heat pump for heating the passenger cabin. Arrows in Figure 4.3 indicate possible power flow directions between electrical components.

As a result the PTC power consumption P_{PTC}, the AC power consumption P_{AC} and the DC/DC converter power consumption $P_{\text{DC/DC}}$ also contribute to the HV battery power demand $P_{\text{Batt,dem}}$:

$$P_{\text{Batt,dem}} = P_{\text{trac,elec}} + P_{\text{PTC}} + P_{\text{AC}} + P_{\text{DC/DC}} \tag{4.17}$$

In case of recuperation $P_{\text{Batt,dem}}$ can become negative indicating that power is transferred to the battery.

The HV battery consists of four cell strings arranged in parallel. Each cell string is composed of 100 lithium nickel-manganese-cobalt oxide (LiNMC) cells connected in series [207]. The cells are liquid cooled and can be heated using PTC elements. The battery can provide an electrical power of 300 kW at a nominal voltage of 370 V [14]. Its nominal capacity of 38.3 kWh enables a vehicle range of more than 200 km [16]. The grid charging power P_{grid} is up to 22 kW with AC via the battery charger inside the vehicle and up to 100 kW with DC [14].

According to [71] the terminal voltage U_{TV} of a LiNMC battery can be described using the following model stated in [134]:

$$\begin{aligned} U_{\text{TV}} &= U_{\text{OCV}}(SOC) - R \cdot I \\ R &= \begin{cases} R^-, & I \geq 0 \\ R^+, & I < 0 \end{cases} \end{aligned} \tag{4.18}$$

Hereby U_{OCV} is the open-circuit voltage as a function of state of charge (SOC). The internal ohmic resistance R can be separated into charge resistance R^+ and discharge resistance R^-. The battery current I is assumed positive for discharge and negative for charge. U_{OCV}, R^+ and R^- are determined by cell testing [134] and have been provided for the HV battery of the research vehicle in look-up tables.

Figure 4.4 shows the equivalent circuit from [134] and the approximate open-circuit voltage derived from [71]. A comprehensive literature review of battery models and their application to LiNMC batteries is given in [71].

According to (4.18) the internal ohmic resistance of the battery causes a HV battery power loss $P_{\text{Batt,Loss}}$, during a HV battery power demand $P_{\text{Batt,dem}}$:

$$P_{\text{Batt,dem}} = U_{\text{TV}} \cdot I = \begin{cases} \underbrace{U_{\text{OCV}} \cdot I}_{=:P_{\text{Batt}}>0} - \underbrace{R^- \cdot I^2}_{=:P_{\text{Batt,Loss}}>0}, & P_{\text{Batt,dem}} > 0 \\ \underbrace{U_{\text{OCV}} \cdot I}_{=:P_{\text{Batt}}<0} - \underbrace{R^+ \cdot I^2}_{=:P_{\text{Batt,Loss}}>0}, & P_{\text{Batt,dem}} < 0 \end{cases} \tag{4.19}$$

This means that if the HV battery power P_{Batt} is positive, i.e. the battery is discharged, the battery must provide $P_{\text{Batt,Loss}}$ in addition to $P_{\text{Batt,dem}}$ in order to compensate for its ohmic losses. As a result, the SOC will decrease faster. Conversely, if P_{Batt} is negative because of recuperation, the SOC will increase slower because the absolute value of the charging power is reduced by $P_{\text{Batt,Loss}}$.

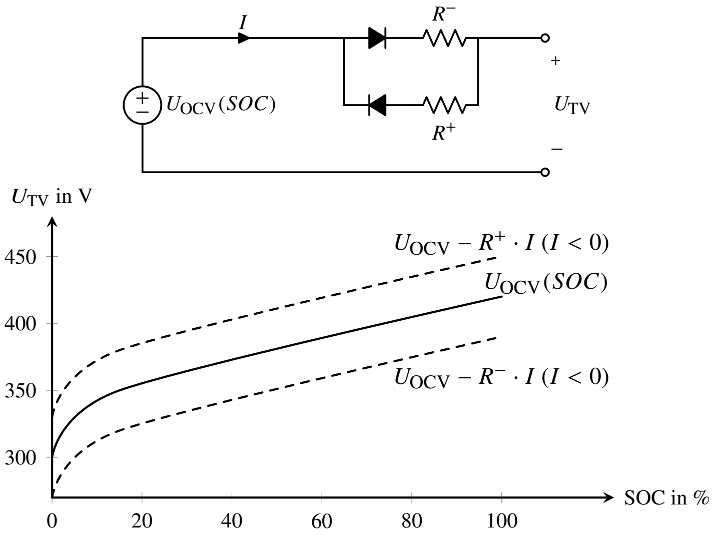


Figure 4.4: High voltage battery characteristics. Top: Equivalent circuit according to the "simple model" in [134] with state of charge (SOC), open-circuit voltage U_{OCV}, charge resistance R^{+}, discharge resistance R^{-}, battery current I and terminal voltage U_{TV}. The diodes are assumed ideal. Bottom: Exemplary U_{TV} of the LiNMC battery in the research vehicle depending on SOC and voltage across the internal ohmic resistance. U_{OCV} was derived from [105].

4.4 Energy consumption and optimization approach

□ Coming from a temporal representation, the HV battery energy E_{Batt} needed for a route results from integrating P_{Batt} over time t, whereby t_{Trip} denotes the trip time:

$$E_{\text{Batt}} = \int_{t=0}^{t_{\text{Trip}}} P_{\text{Batt}}(t)\mathrm{d}t \tag{4.20}$$

Adopted from [79]. One goal of the ALC is to compute an energy-efficient velocity trajectory. For reducing the energy consumption of a BEV three different cost functions in time domain are compared in [81]:

First, motivated by reports in literature that a velocity profile with low acceleration and deceleration is beneficial for energy savings, a penalization of the absolute value of acceleration is investigated. Due to the resulting reluctance to change velocity limits, this approach is suitable for the considered ACC example only to a limited extent. The approach offers the least energy savings in the comparison (8.4 % and 4.1 % depending on the scenario). This approach is also used in [107] and has the benefit that no power consumption model is needed.

Second, the difference of P_{Batt} from its minimum value is penalized, which means that the recuperation maximum is seen as the optimal state. This approach leads to unnecessary strong and frequent recuperations, that are followed by corresponding accelerations. As losses occur in both recuperation and traction mode, this approach is the second least effective (10.6 % and 5.2 % on average).

The highest effectiveness (11.9 % and 5.6 % on average) comes from avoiding deviations of P_{Batt} from zero only for the case of traction and to not consider recuperation.

□ The rationale of the energy consumption optimization approach in this work is to avoid power losses between battery and wheels denoted by P_{Loss}:

$$P_{\mathrm{Loss}} = P_{\mathrm{Batt}} - P_{\mathrm{trac,mech}} \tag{4.21}$$

The corresponding efficiency measure $P_{\mathrm{trac,mech}}/P_{\mathrm{Batt}}$ is known as *energy efficiency of driving* and *tank to wheel efficiency* [10].

P_{Loss} can be expressed with respect to $P_{\mathrm{trac,elec}}$ as in Figure 4.5. The data points result from computing the driving resistances and power train quantitites for various combinations of vehicle velocity v_{Vhcl} and vehicle longitudinal acceleration $\dot{v}_{\mathrm{Vhcl,x}}$ with road curvature and slope both equal zero.

P_{Loss} increases with the absolute value of v_{Vhcl} and $\dot{v}_{\mathrm{Vhcl,x}}$. Arrows indicate how the operating points are shifted when v_{Vhcl} and $\dot{v}_{\mathrm{Vhcl,x}}$ are increased. The P_{Loss} minima for each $P_{\mathrm{trac,elec}}$ can be approximated by a parabola through the origin. The slope of the parabola increases with the absolute value of $P_{\mathrm{trac,elec}}$ indicating that the efficiency of the power transmission decreases with increasing amount of transferred power. *Adopted from [79].*

Sensitivity analyses by [10] reveal that $P_{\mathrm{trac,mech}}/P_{\mathrm{Batt}}$ is the parameter with the highest impact on the energy consumption in a BEV. Frequently,

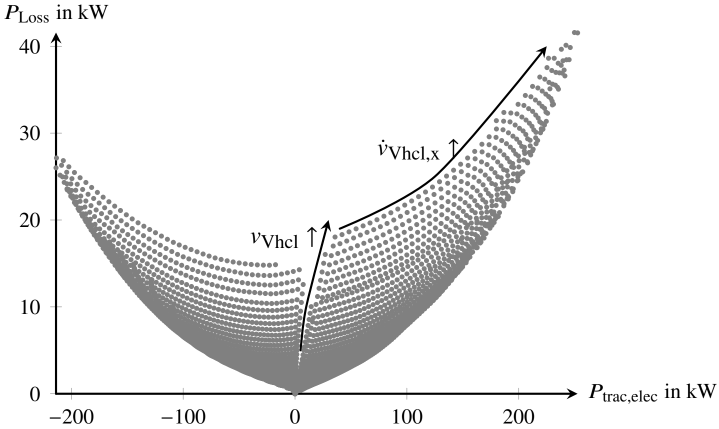


Figure 4.5: Power loss between HV battery and wheels P_{Loss} versus electrical traction power $P_{\mathrm{trac,elec}}$ for various combinations of vehicle velocity v_{Vhcl} and vehicle longitudinal acceleration $\dot{v}_{\mathrm{Vhcl,x}}$. *Adopted from [79].*

consumption models that require detailed knowledge of these usually only roughly known parameter values are used at the risk of inaccurate results [10]. To overcome this problem, this work applies data-driven models that learn such relevant parameters in aggregated form during vehicle operation.

□ Due to lack of sensors P_{Loss} is not known during vehicle operation and therefore cannot be used for a data-driven model. However, $P_{\mathrm{trac,elec}}$ can be derived from voltage and current sensor data of the power electronics and can serve as a proxy for P_{Loss} according to the above considerations. Therefore the energy optimization criterion in the cost function of the ALC is designed to penalize the absolute value of $P_{\mathrm{trac,elec}}$. *Adopted from [79].*

In comparison, the approach of this work comes closest to the third cited method but goes beyond in that it additionally penalizes the also imperfect recuperation mode. Furthermore, using $P_{\mathrm{trac,elec}}$ as optimization criterion is assumed to be more effective than using acceleration and deceleration as the first cited method does. This is because for the same requested acceleration or deceleration the required $P_{\mathrm{trac,elec}}$ can differ significantly depending on road slope and vehicle velocity.

□ The power required for auxilaries such as light, ventilation, heating, cooling and radio can contribute to the total energy consumption for a trip significantly, in case of low velocity $v_{\mathrm{Vhcl}} < 30$ km/h. For $v_{\mathrm{Vhcl}} > 80$ km/h the auxiliary power demand is negligible. It also strongly depends on environmental conditions and individual comfort preferences. Its 5 % and 95 % quantils are given by 0.2 kW and 1.3 kW [10]. *Adopted from [79].*

Furthermore, some components such as the AC compressor are only temporarily active and their activation can be scheduled. Therefore a predictive operating strategy that consideres both traction power demand and cooling power demand has much more energy-saving potential compared to for instance penalizing the sum of instantaneous AC power consumption and electrical traction power. A longitudinal control that includes operating strategies of auxiliaries is presented in [183].

□ For these reasons this work focuses on optimization of traction consumption and neglects auxiliaries by assuming $P_{\mathrm{Batt,dem}} = P_{\mathrm{trac,elec}}$. Even without explicit consideration of power demand of auxiliaries, the penalization of $P_{\mathrm{trac,elec}}$ has a beneficial effect on efficiency because any additional consumption of auxiliaries further increases the HV battery power loss.

The goal is to penalize $P_{\mathrm{trac,elec}}$ only to such an extent, that the ALC avoids inefficient power peaks that have negligible effect on the trip time. *Adopted from [79].* With large penalty for $P_{\mathrm{trac,elec}}$, the vehicle will drive very slow or even come to a standstill because of lack of $P_{\mathrm{trac,mech}}$ that is needed to overcome the driving resistance. Then the resulting trip time will strongly increase as well as the energy consumption because of the power demand of auxiliaries. However such large penalty for $P_{\mathrm{trac,elec}}$ is far beyond the driver acceptance.

4.5 Vehicle models for simulation environment

The preceding sections investigated the driving resistances that result from a certain vehicle velocity v_{Vhcl}, vehicle longitudinal acceleration $\dot{v}_{\mathrm{Vhcl,x}}$, road slope angle α and road curvature κ and stated how the corresponding HV battery power P_{Batt} can be derived from the mechanical traction power $P_{\mathrm{trac,mech}}$. The upper diagram in Figure 4.6 depicts this approach, which is referred to as *backward simulation*. Starting from the desired driving state backward simulation computes the state of components such as gear box against the direction of the effect chain.

Backward simulation assumes that the power train can provide the power needed for the given driving situation. Due to the missing consideration of physical root causes, this approach is also denoted *non-causal modelling*

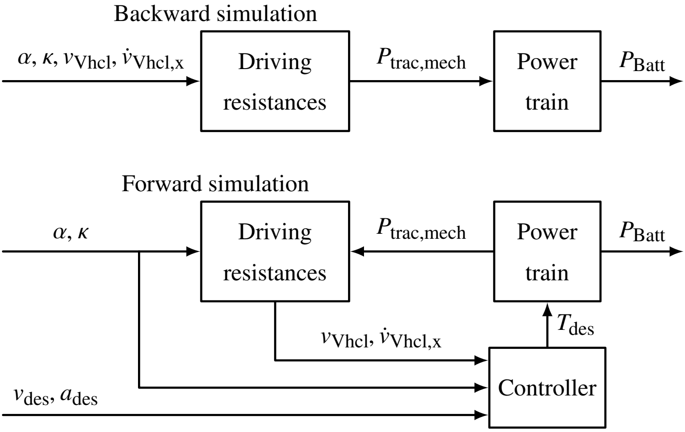


Figure 4.6: Sequences of forward (top) and backward simulation (bottom).

[45]. Non-causal models include no control loop, therefore no controller is needed. However, backward simulations are only suitable for quasi-stationary simulations that do not take into account transient effects. Quasi-stationary states of components are usually derived by interpolation between values of look-up tables as mentioned in Section 4.3.

Backward simulation will be applied for computing the resulting energy consumption for planned trajectories in Chapter 5 and for recorded drives in Chapter 6. The model will be referred to as *open-loop reference model*.

The equations from the previous sections can also be rearranged for a *forward simulation*. The lower diagram in Figure 4.6 depicts this case. Forward simulation computes component states starting from the cause in accordance with the cause-and-effect chain. Therefore this approach is denoted *causal modelling*.

Forward simulations enable dynamic simulations that incorporate dynamic component behavior, e.g. transient effects in the motors and battery. Components can be described using differential equations which are connected to each other according to the cause-and-effect chain. In this work the cause is the desired motor torque T_{des} while $\dot{v}_{\mathrm{Vhcl,x}}$ is at the end of the cause-and-effect chain. $\dot{v}_{\mathrm{Vhcl,x}}$ is derived by rearranging (4.9) and used for computing v_{Vhcl} for the next simulation time step. Causal models include a closed control loop and therefore require a controller.

Forward simulations will be used for testing the interaction between ALC and vehicle. The model will be referred to as *closed-loop reference model*. The controller inside the ALC chooses T_{des} such that deviations of v_{Vhcl} and $\dot{v}_{\mathrm{Vhcl,x}}$ from the desired velocity v_{des} and the desired acceleration a_{des}, respectively, ideally vanish. v_{des} and a_{des} are specified by the planned trajectory of the ALC [48], [62, pp. 37-41].

4.6 Model of reference route

□ The reference route is a roughly 23 km long circuit around the village Weissach in Southwest Germany, hence denoted Weissach route (WR). Figure 4.7 depicts on a map the WR, that comprises urban sections as well as country roads. The legal speed limit varies between 30 km/h and 100 km/h and the road slope ranges from -8 % to 10 %. *Adopted from [79].*

Figure 4.8 displays the corresponding map data consisting of legal speed limit, road curvature and road slope as well as the therewith derived elevation profile. The map data was extracted from a navigation system and will be used in backward and forward simulations with the closed-loop and open-loop reference model, respectively.

4.7 Vehicle models for automated longitudinal control

□ This section presents two models for use within the ALC system. For a given driving situation, the first model provides the required traction force F_{trac} and the second one the required electrical traction power $P_{\mathrm{trac,elec}}$. Both quantities depend on the driving resistances, which in turn are affected by the given driving situation as well as parameters. *Adopted from [79].*

According to sensitivity analyses in [10], the second most relevant parameter for the accuracy of a vehicle model is f_{r}. In the comparison, the model is only affected by ρ for $v_{\mathrm{Vhcl}} > 100$ km/h, whereas c_{w} and A are negligible in the investigated situations. m_{vhcl} is only relevant for hilly trips and trips that lead to higher elevation.

□ During vehicle operation these parameters are not exactly known. m_{vhcl} changes with additional passengers or luggage, c_{w} and A when the convertible top is being opened and f_{r} with tire temperature.

Therefore adaptive models are applied, which estimate these parameters during vehicle operation either explicitly or in aggregated form. In order to be able to adapt a model, its inputs and outputs must be quantities that can be

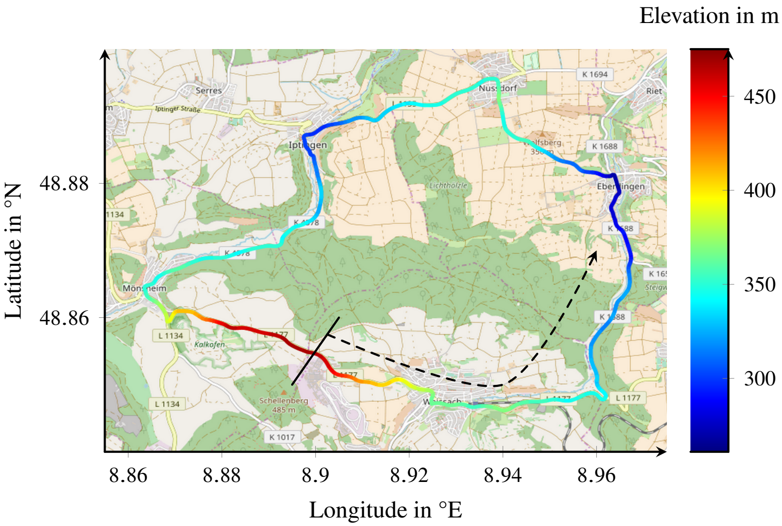


Figure 4.7: Course of the Weissach route (WR) that is used as reference route for the evaluation of the automated longitudinal control. The dotted arrow indicates the driving direction and the straight line the start and end point of the WR.

derived from signals on the Controller Area Network (CAN) bus. *Adopted from [79].* However, only few quantities of the power train that are relevant for this purpose, are measured. Therefore the adaptive models represent the power train properties only on an aggregated level and cannot state losses in individual components, e.g. gear box losses.

Compared to the vehicle reference models for the simulation environment both vehicle models for ALC are simplified in that they first do not take into account road curvature explicitly and in that they second do not distinguish between vehicle mass and rotational masses of the power train as (4.9) does.

Some publications take into account the inertia of rotating parts of the power train [96, 188] for more accuracy but the majority does not [12, 91, 102, 127, 195, 204]. Due to lack of a fly wheel, less complicated gear boxes and smaller motors, the inertia of rotating parts are usually less in a BEV than in a conventionally driven vehicle. As the inertia of rotating parts is usually not known and elaborate to determine, publications that consider

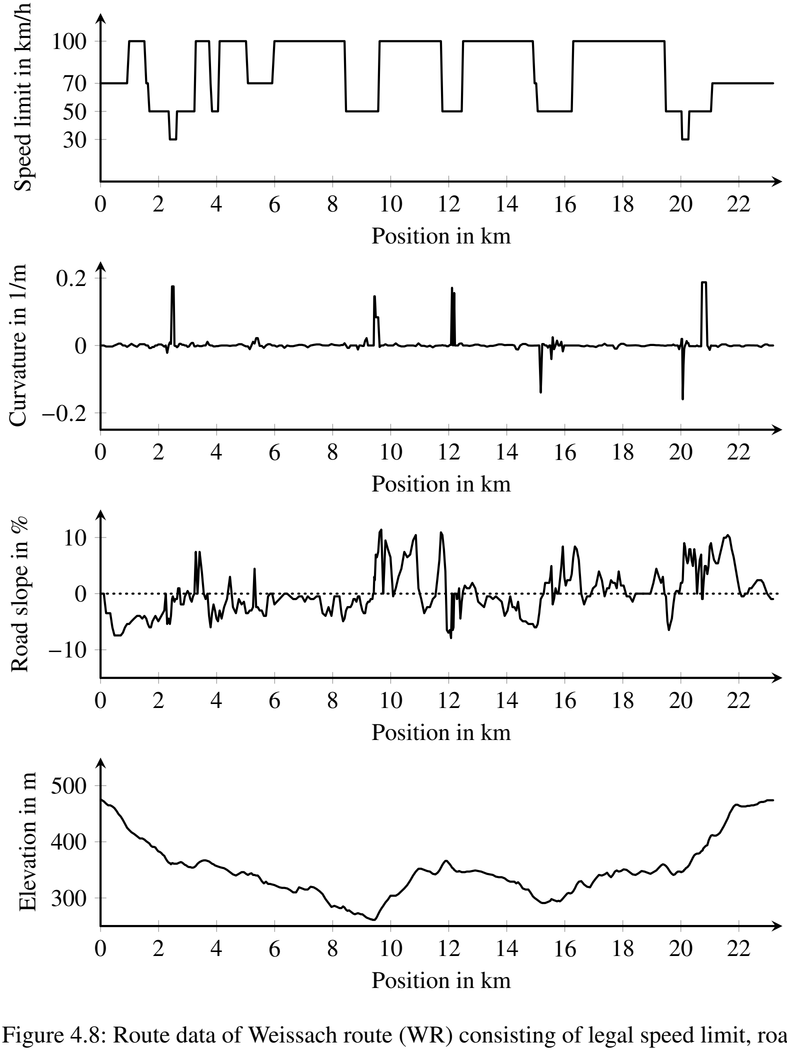


Figure 4.8: Route data of Weissach route (WR) consisting of legal speed limit, road curvature and road slope used for simulation as well as the therewith derived elevation profile. The position is measured from the start of the WR (c.f. Figure 4.7).

it, mostly increase the vehicle mass by 5%. In the research vehicle, the equivalent masses add up to roughly 5.6 % of the vehicle mass. Varying the mass factor between 0 % and 5 % had almost no impact on the results in the sensitivity analyses for a BEV in [10].

□ The simplification regarding rotational masses allows using the sensor longitudinal acceleration a_{x} as a model input. a_{x} is measured by the acceleration sensor and is influenced by both change of velocity and road slope angle:

$$a_{\mathrm{x}} = \dot{v}_{\mathrm{Vhcl,x}} + g \cdot \sin(\alpha) \tag{4.22}$$

Adopted from [79]. The required traction force and the required electrical traction power can differ for the same a_{x} even if all other influence factors are kept constant. This is because in the case that a_{x} results from driving at a slope with constant velocity, no rotational acceleration of the power train occurs but it does if a_{x} results from change of velocity at zero slope. The adaptive models cannot distinguish these cases. The simplification using a_{x} is frequently used for driving resistances parameter estimation [140, 170, 178].

A comprehensive review of vehicle energy consumption models including an analysis of influence factors on the energy consumption is given in [118]. Furthermore, [118] classifies consumption models into white-box, gray-box and black-box models. White-box models are based on high knowledge of the underlying system and incorporate detailed descriptions of subsystems. In contrast, black-box models only learn an input-output pattern from provided data and require no system knowlege. Gray-box models partly use system knowlege and are partly driven by data.

Based on the inputs to a black-box consumption model, [118] distinguishes engine-based consumption models with engine torque and engine speed as inputs, vehicle-based consumption models with instantaneous vehicle speed and acceleration as inputs and modal-based consumption models which use operating modes such as idling, accelerating or cruising as inputs.

Based on the criteria in [118], the proposed traction force model can be regarded as a gray-box model and the proposed model for the electrical traction power as a vehicle-based black-box consumption model.

Similar models can be found in literature. For example, [48] models the tractive force without cornering resistance and the mechanical traction power of a BEV similar to (4.11) and (4.12) depending on instantaneous speed, acceleration and road grade information. The driving resistance parameters are assumed to be known and constant. A parameter that describes the efficiency of recuperation is estimated with the LS method.

In contrast, [51] takes a more aggregated approach and models the traction force with a quadratic polynomial as a function of vehicle velocity. This traction force model is then used for a multivariate model of power demand and energy consumption of a BEV that depends on vehicle velocity and acceleration. In [194] the polynomial regression model additionally uses the SOC as an input. Two KAF algorithms are compared to RLS for representing $P_{\mathrm{trac,elec}}$ depending on a_{x} and velocity in [146].

4.7.1 Adaptive traction force model

□ The ATFM answers the question how much traction force is needed in order to fulfill the driving demand of the longitudinal control in the current situation. The ATFM is used as a pilot control in the controller depicted in the lower diagram in Figure 4.6. The controller computes a motor torque demand such that the vehicle tracks the planned velocity trajectory. The ATFM is based on the following simplification of the longitudinal traction force equation (4.11) that neglects the cornering resistance (4.6) and the equivalent masses of the power train in the inertial force (4.9):

$$\begin{aligned} F_{\mathrm{trac}} &= F_{\mathrm{roll}} + F_{\mathrm{inert,simplified}} + F_{\mathrm{cl}} + F_{\mathrm{air}} \\ &= m_{\mathrm{vhcl}} \cdot g \cdot f_{\mathrm{r}} \cdot \cos(\alpha) + m_{\mathrm{vhcl}} \cdot \dot{v}_{\mathrm{Vhcl,x}} \\ &+ m_{\mathrm{vhcl}} \cdot g \cdot \sin(\alpha) + \frac{\rho}{2} c_{\mathrm{w}} \cdot A \cdot v_{\mathrm{Vhcl}}{}^{2} \end{aligned} \tag{4.23}$$

For small slope angles $\cos(\alpha) \approx 1$ applies. Therefore the first summand can be approximated by the rolling resistance constant F_0 when additionally the vehicle mass in this summand is assumed to change only slowly. Moreover, the second and third summand are merged using (4.22). The traction force F_{trac} is not measured but it can approximately be computed as

$$F_{\mathrm{trac}} = T_{\mathrm{EM,FA}} \cdot \frac{i_{\mathrm{G}}}{r_{\mathrm{dyn,FA}}} + T_{\mathrm{EM,RA}} \cdot \frac{i_{\mathrm{G}}}{r_{\mathrm{dyn,RA}}}, \tag{4.24}$$

whereby the braking torques in the power train are neglected in contrast to (4.16). $T_{\mathrm{EM,FA}}$ and $T_{\mathrm{EM,RA}}$ are no measured quantities either. These signals are computed by the motor ECU from the measured motor voltages and currents using look up tables.

With these adaptions (4.23) in matrix form reads

$$F_{\text{trac}} = \underbrace{(1, a_{\text{x}}, v_{\text{Vhcl}}{}^{2})}_{=:\boldsymbol{C}_{\text{Vhcl}}} \cdot \underbrace{(F_0, m_{\text{vhcl}}, (\frac{\rho}{2} c_{\text{w}} A))^{\top}}_{=:\boldsymbol{x}_{\text{Vhcl}}}, \qquad (4.25)$$

whereby $\boldsymbol{C}_{\text{Vhcl}}$ is the vehicle motion vector. The vehicle parameter vector $\boldsymbol{x}_{\text{Vhcl}}$ summarizes the driving resistance parameters. These parameters are generally not fully known. Therefore $\boldsymbol{x}_{\text{Vhcl}}$ needs to be estimated such that the estimation minimizes the residual between the traction force computed with (4.24) and the traction force according to the model output from (4.25). *Adopted from [79].*

This optimization problem is solved with a KF (c.f. Subsection 3.3.2) and the following model equations:

$$\boldsymbol{x}_{\text{Vhcl},p} = \boldsymbol{x}_{\text{Vhcl},p-1} + \omega_p \qquad \text{(State equation)} \qquad (4.26)$$

$$F_{\text{trac},p} = \boldsymbol{C}_{\text{Vhcl}} \cdot \boldsymbol{x}_{\text{Vhcl},p} + \upsilon_p \qquad \text{(Measurement equation)} \qquad (4.27)$$

p denotes the time step, ω the process noise and υ the measurement noise. Frequently, algorithms such as RLS, KF, EKF are used to determine the driving resistance parameters [140, 170, 178]. □ For the ATFM a Stenlund-Gustafsson M-Kalman filter described in [145] is applied. It uses a regularization from Stenlund and Gustafsson [168]. This regularization method sets lower bounds for the covariance matrix in such a way that the estimated parameters are kept constant in phases of low excitation, i.e. when F_{trac} and $\boldsymbol{C}_{\text{Vhcl}}$ temporally remain constant, such as while driving with roughly constant velocity and road slope. Low excitation also applies to vehicle standstill but for this phase a simple criterion for pausing the adaption can be defined. *Adopted from [79].* The ATFM was created by F. Bleimund [22, pp. 24-29] within *e-generation* and reused by the author of this work without modifications.

Literature also proposes methods that focus on fusing the information of different sensors [74, 198] as well as methods that divide the estimation into several stages [101, 199].

4.7.2 Adaptive electrical power model

Model interfaces and features

□ The AEPM answers the question how much electrical traction power $P_{\text{trac,elec}}$ is required to fulfill the driving demand. The ALC uses the AEPM during trajectory optimization in order to derive trajectories that require few electrical traction power. As shown in Section 4.4, this corresponds to a low power loss between HV battery and wheels.

$P_{\text{trac,elec}}$ can be determined from the measured voltages and currents of the electric motors that are available on the CAN bus. Apart from the power losses in the drive train, $P_{\text{trac,elec}}$ equals the mechanical traction power $P_{\text{trac,mech}}$ which can be computed as the product of traction force F_{trac} and vehicle velocity v_{Vhcl} (c.f. (4.12)). According to Subsection 4.7.1, F_{trac} can be modelled as a function of the CAN signals v_{Vhcl} and a_{x}. Therefore the AEPM relates v_{Vhcl} and a_{x} to $P_{\text{trac,elec}}$:

$$P_{\text{AEPM}} = \text{AEPM}(v_{\text{Vhcl}}, a_{\text{x}}) \tag{4.28}$$

Features of power consumption representation

A substantial fraction of $P_{\text{trac,mech}}$ is needed to overcome the acceleration resistance $m_{\text{vhcl}} \cdot \dot{v}_{\text{Vhcl,x}}$ from (4.9). The corresponding acceleration power $m_{\text{vhcl}} \cdot \dot{v}_{\text{Vhcl,x}} \cdot v_{\text{Vhcl}}$ depends on an unseparable product of $\dot{v}_{\text{Vhcl,x}}$ and v_{Vhcl}. For negligible road slope angle α these quantities are similar to the model inputs because of (4.22) (c.f. [51]). Therefore the linear model structure used in Subsection 4.7.1 for the ATFM is not suitable for the AEPM. *Adopted from [79].*

The power consumption representation in time domain is highly nonconvex, as stated at the example of a $P_{\text{Batt}} = f(F_{\text{trac}}, v_{\text{Vhcl}})$ representation in [81]. An approximation of P_{Batt} with a convex function can be used as a cost function in an convex optimization problem that can be solved more easily than a nonlinear optimization problem. The quadratic approximation accuracy for various representations is also investigated in [81]. Due to the less prominent nonconvexity of the representation in space domain, the highest accuracy is observed for the $(P_{\text{Batt}}/v_{\text{Vhcl}}) = f(F_{\text{trac}}, {v_{\text{Vhcl}}}^2)$ representation, which is the energy consumption in Watt seconds per meter. In space domain, however, vehicle dynamics and the vehicle control problem are more nonlinear [81].

Algorithm for model adaption

□ The AEPM is based on the Fixed-Budget KRLS (FB-KRLS) algorithm described in [180], which is a representative of KAFs introduced in Section 2.4. The source code of FB-KRLS is available at [181]. With each new data point, FB-KRLS adds a new kernel support vector in the $v_{\text{Vhcl}} - a_{\text{x}}$ plane, updates the approximation and discards the least important kernel support vector to keep the required memory constant. During model evaluation the model output P_{AEPM} results from computing the sum of all kernels at $(v_{\text{Vhcl}}, a_{\text{x}})$, weighted with their corresponding control points. *Adopted from [79].*

The model captures aggregated power train losses that are constant or depend on v_{Vhcl} or a_{x}. If vehicle parameters or power train properties change, the AEPM will adapt itself accordingly.

While [81] approximates the nonlinear structure by a convex function to get a convex optimization problem, a KAF approximates the nonlinear structure using a nonlinear function and still offers a convex optimization problem for model adaption. For this kind of convex problems there are solvers that converge in a defined time period, which is important for real-time model adaption.

RLS and two fixed-budget KAFs, called KRLS Tracker (KRLS-T) and QKLMS-FB, are applied to (4.28) in a very similar approach in [146]. KRLS-T achieves the highest accuracy, followed by QKLMS-FB. As expected, the linear RLS, proves not suitable for strong nonlinearity and achieves the lowest accuracy.

All three mentioned KAF use the Gaussian kernel defined in (2.21). The KRLS-T algorithm published in [179] in 2012 is a KAF algorithm for time-varying regression that includes a forgetting factor that can handle non-stationary scenarios. In contrast, the FB-KRLS published in 2010 stems from a sliding-window approach with the improvement that it maintains not the kernels for the latest but for the most important data. Its forgetting mechanism is not optimized towards non-stationary scenarios [181].

The idea to model the power train characteristics with a FB-KRLS originates from S. Rhode [146], who also provided a first script in MATLAB that performs this task. Based on this, both researchers continued the investigations independently from each other, so details differ. The content of the remainder of this subsection, except from the normalization described below, stems from the author of this work. The novelity of this work regarding FB-KRLS is limited to the application of this kind of black-box model for trajectory optimization and within the ALC along with required problem-specific adaptions such as the data shift as well as the determination of model hyperparameters, both also mentioned below.

Data transformation

□ For various $(v_{\mathrm{Vhcl}}, a_{\mathrm{x}})$ combinations the trajectory optimization algorithm evaluates the AEPM and takes decisions based on the AEPM outputs. The standard FB-KRLS output is zero in border areas where no data points have occured yet. This means that if the trajectory optimization evaluates the model for high velocities or accelerations that are beyond the capabilities of the vehicle, the AEPM states $P_{\mathrm{trac,elec}} \approx 0$. To avoid that results of evaluations of the model in its border areas appear as efficient and therefore favorable driving states to the trajectory algorithm, the model outputs must have large absolute values in border areas. This can be achieved by shifting $P_{\mathrm{trac,elec}}$ in training data by subtracting a large offset value.

After the shift, all components of training data $(v_{\mathrm{Vhcl}}, a_{\mathrm{x}}, P_{\mathrm{trac,elec}})$ and evaluation data $(v_{\mathrm{Vhcl}}, a_{\mathrm{x}})$ are additionally scaled to a ranges between -1 and 1 by multiplying each of them with a constant between 0 and 1. This process is called normalization. Frequently better aproximation quality is observed for models that work with normalized data.

Shift and normalization form the transformation. Results of model evaluations are retransformed by a denormalization and backshift operation before they are provided to the trajectory optimization algorithm.

Figure 4.9 depicts the retransformed model output as a function of a_{x} for various nonnegative vehicle velocities. Negative velocities are not shown for simplicity and because with ALC the vehicle only drives forward. For medium negative a_{x} the $P_{\mathrm{trac,elec}}$ is negative indicating that power is recuperated into the battery. The gray shaded area indicates possible operating points of the vehicle while driving with activated ALC under the assumption of no slope. As the ALC limits are defined with respect to $\dot{v}_{\mathrm{Vhcl,x}}$, the width of the area will change in presence of slope according to (4.22).

In border areas that exceed the vehicle capabilities the shift leads to very large AEPM outputs of up to three times the maximum $P_{\mathrm{trac,elec}}$. As a result, the trajectory optimization algorithm avoids trajectories that include such unreachable $(v_{\mathrm{Vhcl}}, a_{\mathrm{x}})$ combinations. *Adopted from [79].*

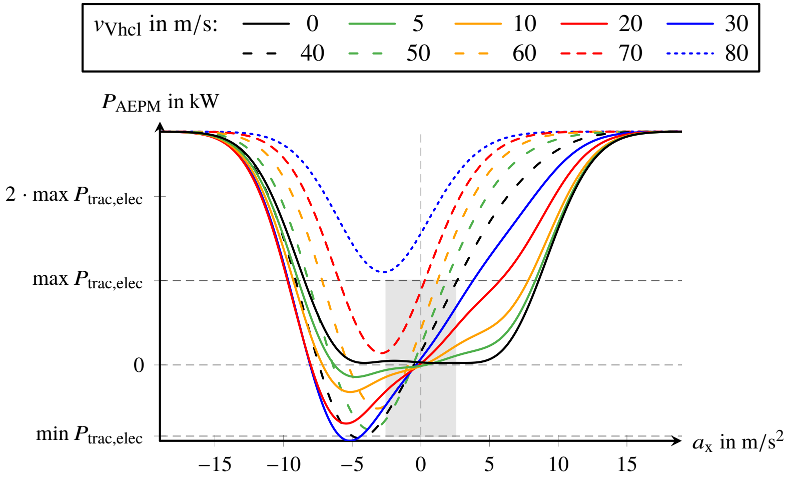


Figure 4.9: Electrical power P_{AEPM} according to the adaptive electrical power model (AEPM) depending on sensor longitudinal acceleration a_{x} for various vehicle velocities v_{Vhcl}. The gray shaded area is the operating area of the ALC, $\max P_{\text{trac,elec}}$ the maximum traction power and $\min P_{\text{trac,elec}}$ the maximum recuperation power. *Adopted from [79].*

In case of negative slope peaks or strong braking, the simple shift leads to a zero-crossing of $P_{\text{trac,elec}}$ at roughly $-7\ \text{m/s}^2 \leq a_{\text{x}} \leq -5\ \text{m/s}^2$ before $P_{\text{trac,elec}}$ increases to large values. This case can still mislead the trajectory optimization algorithm but for the given setup it did not present a problem. As Figure 4.8 indicates, on the WR peaks in the course of the road slope are stronger in the positive direction than in the negative direction. This asymmetrical road slope distribution causes that in AEPM evaluations the maximum absolute value of a_{x} is greater for $a_{\text{x}} > 0$ than it is for $a_{\text{x}} < 0$ and that border areas on the left hand side of Figure 4.9 are not reached.

To prevent that negative slope peaks cause issues on other hilly routes, the shift should be improved by subtracting a plane from the model input data instead of a constant. This plane can be designed so that it goes through the origin and is tilted upwards in the directions of the v_{Vhcl} and a_{x} axes, e.g. $-c_1 \cdot v_{\text{Vhcl}} - c_2 \cdot a_{\text{x}} + c_3 \cdot P_{\text{trac,elec}} = 0$ with constants $c_1, c_2, c_3 \geq 0$. By choosing these constants appropriately, physically unreasonable zero-crossings of P_{AEPM} can be avoided.

Determination of model hyperparameters

The question is how kernel variance σ^2 in (2.21) and kernel count M should be chosen for the AEPM. For better differentiation from other parameters like control points, σ^2 and M are called hyperparameters [146] and determining them is part of the model selection process [72, pp. 128-130]. A model with many parameters usually provides more accurate results at the cost of increased effort for parameterization and evaluation.

Model selection criteria, also called *information criteria*, assess models and thereby can help to find a trade-off between model complexity and model quality. The general structure of an information criterion is:

$$\text{criterion} = P \cdot \ln\left(\frac{SSE}{P}\right) + \text{penalty term} \tag{4.29}$$

$\ln(\cdot)$ is the natural logarithm, P the number of observations and SSE is the sum of squared errors with

$$SSE = \sum_{p=1}^{P} \left(y_p - \hat{y}_p\right)^2 . \tag{4.30}$$

y denotes the measurement and $\hat{y}$ the model output. The selected criterion is computed for a set of trained models and the model with lowest criterion value is chosen. Several information criteria have been introduced, which are based on different principles and differ in the penalty term. In a comparison in [89] the Hannan-Quinn information criterion (HQC) performs best:

$$\text{HQC} = P \cdot \ln\left(\frac{SSE}{P}\right) + M \cdot \ln\left(\ln(P)\right) \tag{4.31}$$

Number of model parameters M and kernel variance σ^2 from (2.21) span the hyperparameter space for the search for the most suitable AEPM. For each combination of $M = 10, 20, \ldots, 100$ and $\sigma^2 = 1/1, 1/2, \ldots, 1/30$ a model is trained and tested to determine the model with the lowest HQC.

Data for model training and test was created using the open-loop reference model. Thereby the vehicle longitudinal acceleration was sampled with $\dot{v}_{\text{Vhcl,x}} = -10, -9.8, \ldots, 10\ \text{m/s}^2$ and the vehicle velocity with $v_{\text{Vhcl}} = 0, 0.5, \ldots, 56\ \text{m/s}$. Slope and curvature were assumed equal zero. Operating points that are beyond the vehicle capabilities according to the open-loop reference model were removed from the data set. The data set that comprises all possible vehicle operating points (PVOP) in random order is split into a PVOP training data set and a PVOP test data set. The randomly chosen PVOP test data set comprises 10 % of the original PVOP data set and the remaining data points form the PVOP training data set.

Additionally, a WR data set was created from the closed-loop reference model output during a simulated drive with the ALC on the WR including slope and road curvature information. The WR data set was split analogously into WR training data set and WR test data set. PVOP data set and WR data set contain roughly the same amount of data points.

The upper diagram of Figure 4.10 depicts the best HQC as well as the corresponding σ^2 for each investigated kernel count. Only for models that are both trained and tested using data from the PVOP data set the HQC course shows a local minimum at 50 kernels along with comparatively large kernel variance values.

The lower diagram of Figure 4.10 depicts the corresponding 95 % quantile $Q_{95\%}$ of absolute values of the relative error e_{rel} between the output of the AEPM and the reference model for PVOP and WR, respectively. With increasing M the 95 % quantiles decrease.

Models that are both trained and tested with PVOP data achieve better HQC values and lower quantiles than models that are trained with PVOP data and tested with WR data. The PVOP data set contains much more dynamic driving states than the WR. Furthermore, only the PVOP test data set evaluates a model that has been trained with the PVOP training data set in its border areas. Therefore, the worse result for the WR test data set seems counterintuitive at first.

However, only the WR data set considers road curvature and road slope. On the WR the road slope is up to 10 %. The AEPM does not use road curvature input and summarizes road grade and change of vehicle velocity in the input a_{x}. Hence, the results indicate that road geometry has more influence on the quality of the model output than the driving style. However, for $M \geq 40$ the quantils are below 2 % for all considered combinations of training and test data sets. Therefore not considering road curvature in the AEPM explicitly is acceptable.

Since the goal is not to optimize the model parametrization with respect to the WR but to find a suitable model for the vehicle in general, the best model parametrization according to HQC for the PVOP test set was chosen. This is a model with $M = 50$ and $\sigma^2 = 1/9$ that achieves $Q_{95\%}(|e_{\text{rel}}|) = 0.94$ %

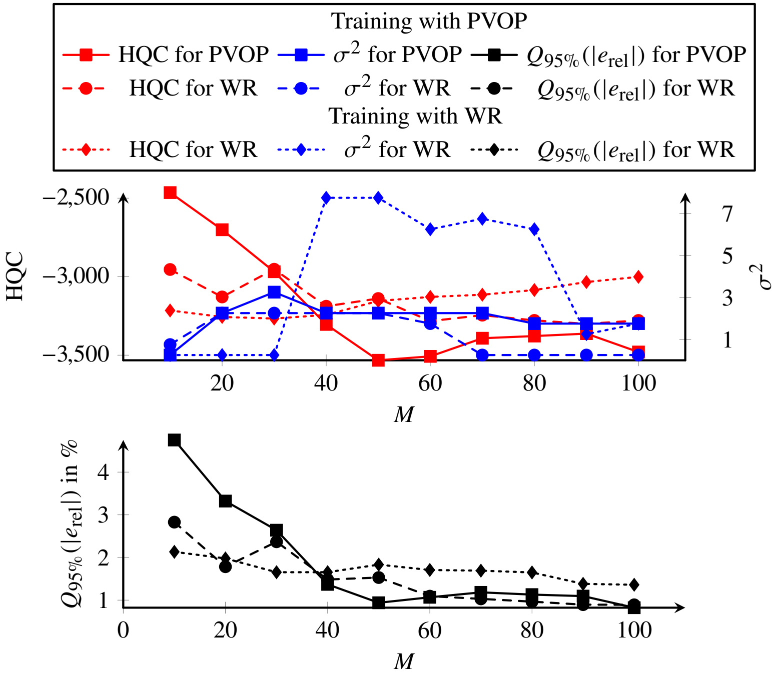


Figure 4.10: Results of full factorial search for best model hyperparameters kernel count M and kernel variance σ^2. Top: Both the best HQC and the corresponding σ^2 are plotted versus M for each combination of training and test using the data sets from possible vehicle operating points (PVOP) and Weissach route (WR) . Bottom: 95 % quantile $Q_{95\%}$ of absolute values of relative error e_{rel} between output of AEPM and reference model versus M.

for PVOP and $Q_{95\%}(|e_{\mathrm{rel}}|) = 1.5$ % for WR. This AEPM model is used in the remainder of this work and depicted in Figure 4.9.

With the hyperparameter optimization technique from [146] the KRLS-T accuracy increases for $M \leq 50$ and converges for more kernels. Therefore $M = 50$ is considered the optimal KRLS-T hyperparameter, analogously to this work. However, with $\sigma^2 = 6.25$ the reported kernel variance is different. Reasons can be contrasting data sets and differences in the KAF algorithms themselves.

5 Planning of velocity trajectories

This chapter presents a trajectory planning approach based on the B-spline approximation methods from Chapter 3. Section 5.1 states how an upper speed limit for the route ahead is created from map data. Section 5.2 includes considerations on how to represent the planned vehicle velocity and Section 5.3 describes the trajectory optimization. Section 5.4 proposes an extension of the trajectory optimization problem for taking into account the required electrical traction power. Section 5.5 states ways to enforce constraints on the trajectory beginning and Section 5.6 summarizes the scientific contribution of this work regarding trajectory planning.

5.1 Generation of upper speed limit

An upper speed limit for each meter of the road section ahead of the vehicle is computed using map data. The upper speed limit takes into account legal restrictions, limitations from driving dynamics as well as safety and comfort requirements and serves as input data for the trajectory optimization process.

The map data consists of three $P \times 1$ vectors, each with component index $p = 1, \ldots, P$. The vectors state the meter-discrete courses of road slope γ, road curvature κ and legal speed limit $v_{\text{Lim,Law}}$ for the road section ahead of the vehicle in driving direction. The first component of each vector refers to the current vehicle position. P is called the *length of the electronic horizon*.

Figure 5.1 depicts various speed limits that occur during generation of the upper speed limit and are described in the following paragraphs.

The course of the legal speed limit $v_{\text{Lim,Law}}$ forms the basis of the upper speed limit. The speedometer in a vehicle includes an offset and therefore indicates a velocity that is slightly higher than the actual vehicle velocity. For better acceptance by the driver, the legal speed limit including speedometer offset $v_{\text{Lim,Speedo}}$ is used instead of $v_{\text{Lim,Law}}$. This causes that the speedometer will indicate roughly the same velocity value as the traffic sign when there are no relevant restrictions other than $v_{\text{Lim,Law}}$ present.

Due to tight curves, crests or comfort requirements the upper speed limit needs to be corrected starting from $v_{\text{Lim,Speedo}}$ further downwards.

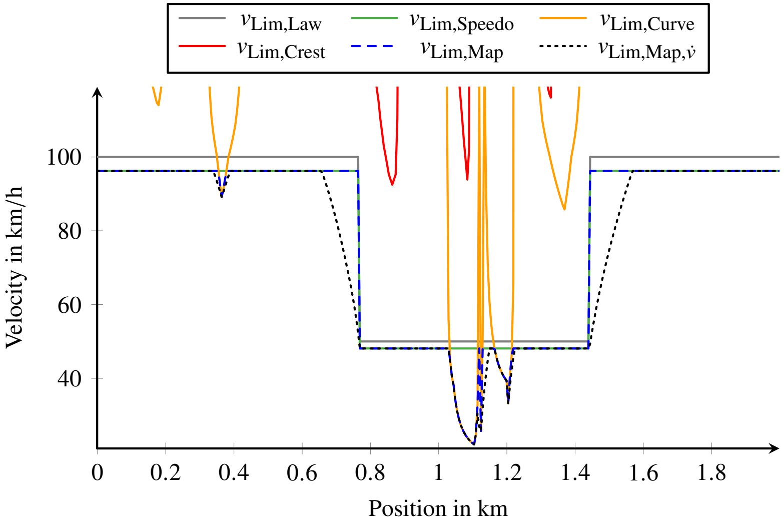


Figure 5.1: Computation of upper speed limit with respect to position measured from vehicle at km 11 of Weissach route in driving direction. Corresponding route data depicted in Figure 4.8 between km 11 and km 13. Legal speed limit $v_{\text{Lim,Law}}$, legal speed limit including speedometer offset $v_{\text{Lim,Speedo}}$, speed limit resulting from curvature $v_{\text{Lim,Curve}}$ and speed limit resulting from crests $v_{\text{Lim,Crest}}$ are intermediate quantities. The speed limit resulting from map data $v_{\text{Lim,Map}}$ is the minimum of $v_{\text{Lim,Speedo}}$, $v_{\text{Lim,Curve}}$ and $v_{\text{Lim,Crest}}$. End result is the speed limit from map data with desired acceleration $v_{\text{Lim,Map},\dot{v}}$.

Driving with vehicle velocity v_{Vhcl} on a road with curvature κ causes a lateral acceleration $\dot{v}_{\text{Vhcl,y}} = v_{\text{Vhcl}}^2 \cdot \kappa$ (c.f. (4.5)). A maximum absolute value of lateral acceleration $\dot{v}_{\text{Vhcl,y,max}}$ is specified, which leads to the speed limit resulting from curvature $v_{\text{Lim,Curve}}$ given by

$$v_{\text{Lim,Curve}} \leq \sqrt{\frac{\dot{v}_{\text{Vhcl,y,max}}}{|\kappa|}}. \tag{5.1}$$

Thereby $|\cdot|$ denotes the absolute value. A comfortable lateral acceleration maximum varies with vehicle speed [153]. As a first draft a lateral acceleration table $\dot{v}_{\text{Vhcl,y,max}}(v)$ was created using a characteristic curve from [153].

In test drives the characteristic curve was further adapted to suit the vehicle characteristics. The lateral acceleration table consists of $\left(\dot{v}_{\text{Vhcl,y,max}}, v\right)$ supporting points because these quantities can be specified more conveniently than κ in test drives. With (4.5) the corresponding κ values are computed. Thereafter $\dot{v}_{\text{Vhcl,y,max}}$ for a given κ is determined from map data by linear interpolation and $v_{\text{Lim,Curve}}$ is derived using (5.1).

Crests on the route cause another restriction on the upper speed limit. From road slope data an elevation profile is obtained and it is calculated how far ahead the road can be seen according to geometrical considerations. It needs to be ensured that the vehicle can always come to a standstill within the sighting distance $\Delta s_{\text{Sighting}}$. Assuming that the absolute value of the maximum deceleration is the gravitational constant g leads to the speed limit resulting from crests $v_{\text{Lim,Crest}}$ given by $v_{\text{Lim,Crest}} \leq \sqrt{2 \cdot \Delta s_{\text{Sighting}} \cdot g}$.

The speed limit resulting from map data $v_{\text{Lim,Map}}$ is the minimum of $v_{\text{Lim,Speedo}}$, $v_{\text{Lim,Curve}}$ and $v_{\text{Lim,Crest}}$. Usually $v_{\text{Lim,Crest}}$ is high compared to the other quantities and therefore rarely determines $v_{\text{Lim,Map}}$.

In the last step the desired longitudinal acceleration a_{des} is taken into account. The desired acceleration and deceleration values are derived by interpolating with respect to velocity v in separate acceleration tables for acceleration $a_{\text{des,pos}}(v)$ and deceleration $a_{\text{des,neg}}(v) < 0$. Enforcing the constraints

$$\begin{aligned} v_{p+1} &\leq \sqrt{v_p + 2 \cdot \Delta s \cdot a_{\text{des,pos}}(v_p)}, \; p = 1, 2, \ldots, P-1 \\ v_{p-1} &\leq \sqrt{v_p - 2 \cdot \Delta s \cdot a_{\text{des,neg}}(v_p)}, \; p = P, P-1, \ldots, 2 \end{aligned} \tag{5.2}$$

with $\Delta s = 1$ on $v_{\text{Lim,Map}}$ gives the speed limit from map data with desired acceleration $v_{\text{Lim,Map},\dot{v}}$, which is the output of the whole procedure.

The content described above in this section originates from F. Bleimund [22, pp. 29, 30], who implemented it during *e-generation*. The adaptions done during *e-volution* are limited to enhancements of the acceleration tables and adaptions of their entries during test drives.

Other works such as [114, 139] additionally specify a lower speed limit as a certain percentage of the upper speed limit. Both limits then form a so-called *driving tube* that defines the solution space of valid trajectories. An alternative term is *driving envelope* [44]. This work specifies no explicit lower limit but enforces a nonnegativity constraint on velocity trajectories as Section 5.5 will state.

5.2 Representation of vehicle velocity in time domain

There are several ways to define a course of velocity that stays below the upper speed limit. Figure 5.1 illustrates that in general only a functional representation with many degrees of freedom can sufficiently adapt to the upper speed limit over several kilometers.

Section 2.3 investigated the properties of several function types and the advantages of spline functions over polynomials. Based on the conclusions drawn there, the velocity trajectory will be represented using a cubic B-spline function. This function is twice continuously differentiable which means the trajectory velocity v_{TJY} is smooth. This is beneficial for driving comfort but not sufficient for jerk-free driving behavior. If the B-spline function has many degrees of freedom in a short time period and unfavorably chosen control points, it can lead to an oscillating and uncomfortable velocity course.

A further helpful feature is that in each interval a B-spline function lies within the convex hull of the control points that are relevant for this interval (c.f. Section 2.3 and Section 3.1). This feature allows to enforce $v_{\text{TJY}} \geq 0$ by simply restricting the control points to nonnegative values.

Since the upper speed limit depends on the position, a straight-forward approach is to define the trajectory with respect to the position as well. This was done in [22, 139].

RBA described in Subsection 3.3.3 allows to adapt the control points of a B-spline function iteratively such that it approximates a set of data points in the WLS sense. With RBA a spatial velocity trajectory $v_{\text{TJY}}(s)$ that depends on the position s can be adapted to the upper speed limit by solving the LS problem

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{p=1}^{P} \left(v_{\text{Lim,TJY},p} - v_{\text{TJY}}(s_p)\right)^2 \tag{5.3}$$

using the data set

$$(s_p, v_{\text{Lim,TJY},p}),\ s_p = p - 1,\ p = 1, 2, \ldots, P, \tag{5.4}$$

in which $v_{\text{Lim,TJY},p}$ is derived from $v_{\text{Lim,Map},\hat{v}}$ as (5.12) will define. A spatially defined trajectory can easily be compared against the upper speed limit and adjusted downwards at any position without risking to violate the upper speed limit at another position. However, the spatial trajectory definition has at least four flaws:

First, at low speeds the position s changes slowly in a given time interval. If the knots of $v_{\text{TJY}}(s)$ are spatially equidistant, the trajectory can only represent a comparatively reluctant driving behavior at low speeds, whereas at highway speeds it has an excessive number of degrees of freedom in each time interval that facilitates undesired velocity oscillations and increases the computational effort unnecessarily.

RBA only chooses the control points and requires knots as an input. Regarding RBA this means that simply specifying spatially equidistant knots is not suitable. This was also confirmed by real test drives with a spatial trajectory planning approach based on RBA. Therefore a knot placement procedure was added that computed spatial knot positions that were equidistant with respect to time. The degrees of freedom were then spatially dense at low speeds and with increasing v_{TJY} their density decreased.

There are also solutions for adapting both knots and control points when fitting a B-spline function to data [36, 93, 130] but this problem is a nonconvex one with many local minima and therefore difficult to solve [15]. For example, [158] solves a reduced nonlinear B-spline LS approximation problem, in which only the knots are optimization variables using a generalized Gauss-Newton method, whereas [141] applies the LM algorithm from Subsection 3.4.1. A review of state of the art methods for this kind of problem is provided by [43].

Second, a spatial velocity trajectory cannot represent driving off again after coming to a standstill because if $v_{\text{TJY}}(s^*) = 0$, at any position s^*, then s^* cannot be left anymore. Hence, starting from a standstill requires a different trajectory planning approach.

Third, the time t^* at which a $v_{\text{TJY}}(s)$ trajectory reaches a certain position s^* requires to compute the integral $t(s^*) = \int_{s=0}^{s^*} \frac{1}{v_{\text{TJY}}(s)} \mathrm{d}s$. For a cubic function $v_{\text{TJY}}(s)$ the solution can be determined using a partial fraction decomposition (PFD). Since the solution of PFD depends on the control point values, a system of equations needs to be solved repeatedly.

Fourth, derivates of $v_{\text{TJY}}(s)$ are no common quantities with known interpretation. Common quantities refer to time and deriving them from $v_{\text{TJY}}(s)$ is costly. For example, the derivative $v_{\text{TJY}}'(s)$ is not the trajectory acceleration $a_{\mathrm{TJY}}(s)$ but the pseudo trajectory acceleration [55]. a_{TJY} needs to be computed from $v_{\mathrm{TJY}}(s)$ as a product:

$$a_{\mathrm{TJY}}(s) = \frac{\mathrm{d}v_{\mathrm{TJY}}(s)}{\mathrm{d}t} = \frac{\mathrm{d}s}{\mathrm{d}t} \cdot \frac{\mathrm{d}v_{\mathrm{TJY}}(s)}{\mathrm{d}s} = v_{\mathrm{TJY}}(s) \cdot {v_{\mathrm{TJY}}}'(s) \tag{5.5}$$

Hence, a_{TJY} cannot be repesented by the linear combination of basis functions and control points from (2.17) and (3.3), respectively, anymore. Equation (5.5) gives oscillations in the acceleration course, which become larger as the knot distance is decreased to allow for more agile vehicle behavior. The reason is that the result of (5.5) is no B-spline product and therefore also does not fulfill the convex hull property. Ways to correctly compute product functions of B-spline functions are given in [121, 133]. The approach in [133] relies on intermediate transformations to the Bézier functions (c.f. Section 2.3). A sliding window B-spline multiplication algorithm is presented in [30].

A time dependent velocity trajectory overcomes these flaws. First, the knots can simply be chosen temporally equidistant. Then the trajectory represents a vehicle behavior whose agility does only depend on the chosen constant knot density but not on vehicle velocity. Second, driving off from a standstill poses no problem because t still changes during a standstill and so can $v_{\mathrm{TJY}}(t)$. Third, the corresponding trajectory position $s_{\mathrm{TJY}}(t)$ is obtained efficiently by integration:

$$s_{\mathrm{TJY}}(t) = \int_{\tau=0}^{t} v_{\mathrm{TJY}}(\tau)\mathrm{d}\tau \tag{5.6}$$

Fourth, meaningful quantities like trajectory acceleration and trajectory jerk arise from a linear combination of basis functions and control points.

For these reasons a time dependent trajectory $v_{\mathrm{TJY}}(t)$ is chosen. The problem of adapting $v_{\mathrm{TJY}}(t)$ to the upper speed limit is formulated analogously to (5.3) as an approximation problem but an additional change of representation space from position to time is included:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{p=1}^{P} \left(v_{\mathrm{Lim,TJY}}\left(s_{\mathrm{TJY}}(t_p)\right) - v_{\mathrm{TJY}}(t_p)\right)^2 \tag{5.7}$$

The trajectory position s_{TJY} is given by (5.6) and computed as in (3.10).

In contrast to (5.3), the error $v_{\text{Lim,TJY}}\,(s_{\text{TJY}}) - v_{\text{TJY}}$ in (5.7) does not linearly depend on the control point vector $\boldsymbol{x}$. Instead, $\boldsymbol{x}$ is linked to the error by the nonlinear upper speed limit, which makes (5.7) a NWLS problem.

The described approach is similar to parametric B-spline curve approximation. As mentioned in Section 2.3, a B-spline curve is a generalization of a B-spline function that uses multi-dimensional control points. These enable an independent curve shape in each dimension such that the curve can represent edges of a geometrical object [3]. The velocity trajectory approximates $(s_p, v_{\text{Lim,TJY},p})$ data points with respect to the parameter t. However, s_{TJY} and v_{TJY} are coupled via (5.6) and by still using a B-spline function with scalar control points this constraint can be enforced.

A disadvantage of the temporal trajectory definition is that road grade, speed limit and road curvature are position dependent, whereas the position of the vehicle depends on the future course of velocity. Therefore a velocity optimization requires a vehicle position estimate [82]. Specifically for the approach in (5.7) this means that if any component of the solution $\hat{\boldsymbol{x}}$ of the optimization problem is modified, the trajectory will likely exceed $v_{\text{Lim,Map},\dot{v}}$ at some later point in time. In contrast, with $v_{\text{TJY}}(s)$ control point values can always be reduced without risking to violate $v_{\text{Lim,Map},\dot{v}}$ afterwards.

For completeness a $s_{\text{TJY}}(t)$ trajectory is considered as well. Against such a trajectory argues the fact that s_{TJY} is unbounded and may take large values for long trajectories. This encourages numerical problems and instable computations in optimization algorithms.

5.3 Trajectory optimization

□ Trajectory optimization involves adapting the $v_{\text{TJY}}(t)$ trajectory function to the data set in (5.4) that is created using the upper speed limit $v_{\text{Lim,Map},\dot{v}}$. $v_{\text{TJY}}(t)$ is defined according to (3.3) with degree $d = 3$, knot vector $\boldsymbol{\kappa}$ and control point vector $\boldsymbol{x}$. $\boldsymbol{\kappa}$ given by

$$\begin{aligned}\boldsymbol{\kappa} &= (\kappa_1, \kappa_2, \ldots, \kappa_K) \\ &= (\Delta t_\kappa \cdot (-d), \Delta t_\kappa \cdot (-d+1), \ldots, \Delta t_\kappa \cdot (-d+K-1))\end{aligned} \tag{5.8}$$

has equidistant and strictly monotonously increasing entries. Δt_κ denotes the constant temporal distance of neighboring knots. Due to the choice of $\boldsymbol{\kappa}$, the trajectory can be evaluated for $t \geq 0$.

t is discretized using a constant temporal distance of neighboring data points Δt_{It}:

$$t_p = (p-1) \cdot \Delta t_{\mathrm{It}}, \; p = 1, \ldots, P \tag{5.9}$$

The approximation problem that is solved reads

$$\begin{aligned} \hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{p=1}^{P} \Big(& R_v^{-1} \cdot \left[v_{\mathrm{Set}} \left(s_{\mathrm{TJY}}(t_p) \right) - v_{\mathrm{TJY}}(t_p) \right]^2 \\ & + R_a^{-1} \cdot \left(a_{\mathrm{Set}} - a_{\mathrm{TJY}}(t_p) \right)^2 + R_j^{-1} \cdot \left(j_{\mathrm{TJY}}(t_p) \right)^2 \Big) . \end{aligned} \tag{5.10}$$

a_{TJY} denotes the trajectory acceleration and j_{TJY} the trajectory jerk. These quantities are the first and second derivative of v_{TJY} and can be calculated according to (3.5). The trajectory position s_{TJY} is measured from $t = 0$, hence $s_{\mathrm{TJY}}(t = 0) = 0$. s_{TJY} can be computed using (3.10) and (5.6).

Unless stated otherwise, the set point of trajectory velocity v_{Set} and the set point of trajectory acceleration a_{Set} are defined as follows:

$$v_{\mathrm{Set}} = v_{\mathrm{Lim,TJY}}, \; a_{\mathrm{Set}} = 0 \tag{5.11}$$

In contrast to (5.7), the optimization problem in (5.10) includes two additional summands that refer to the derivatives a_{TJY} and j_{TJY} of the $v_{\mathrm{TJY}}(t)$ trajectory. By penalizing deviations of a_{TJY} from a_{Set} and deviations of j_{TJY} from zero, the $v_{\mathrm{TJY}}(t)$ function is stabilized and uncomfortable driving is avoided, which can be caused by acceleration peaks and velocity oscillations.

Each optimization goal has a corresponding weight. R_v^{-1} denotes the weight of velocity error square, R_a^{-1} the weight of acceleration error square and R_j^{-1} the weight of jerk error square. The reciprocals of the weights follow the interpretation of the filter algorithms from Subsection 3.3.2 and Subsection 3.4.2 by referring to the variances of the artificial measurements v_{Set}, a_{Set} and 0. R_v is the variance of velocity measurement, R_a the variance of acceleration measurement and R_j the variance of jerk measurement. R_v^{-1} can be interpreted as a weight for the optimization goal of few travel time whereas both R_a^{-1} and R_j^{-1} refer to driving comfort. A suitable weighting combination reads $R_v = 5$, $R_a = 10$, $R_j = 1$. This combination was derived by experiments, validated in real test drives and will be used in the remainder of this work.

WLS and NWLS approximation algorithms usually solve an unconstrained problem and compute a function that is close to the data points. Problem (5.10) follows this approach and penalizes deviations of the function value from the data symmetrically. This interpretation of the data neglects that the data points of the upper speed limit also present a constraint to v_{TJY}. This constraint character is taken into account by using modified data in (5.10). Set point of trajectory velocity v_{Set} in (5.11) is not $v_{\mathrm{Lim,Map},\dot{v}}$ but the speed limit for trajectory optimization $v_{\mathrm{Lim,TJY}}$:

$$\begin{aligned} v_{\mathrm{Lim,TJY}} &= \min\left(v_{\mathrm{Lim,Map},\dot{v}}[p_{\min}], \dots, v_{\mathrm{Lim,Map},\dot{v}}[p_{\max}]\right) \\ p_{\min} &= \mathrm{round}\left(s_{\mathrm{TJY}} - v_{\mathrm{TJY}} \cdot \Delta t_{\mathrm{Lim,TJY}} + 1\right) \\ p_{\max} &= \mathrm{round}\left(s_{\mathrm{TJY}} + v_{\mathrm{TJY}} \cdot \Delta t_{\mathrm{Lim,TJY}} + 1\right) \end{aligned} \tag{5.12}$$

$v_{\mathrm{Lim,TJY}}$ is the minimum of $v_{\mathrm{Lim,Map},\dot{v}}$ within a spatial distance around the calculated trajectory position s_{TJY}. The distance depends on the trajectory velocity v_{TJY} and the temporal safety margin to upper speed limit $\Delta t_{\mathrm{Lim,TJY}}$. $\Delta t_{\mathrm{Lim,TJY}}$ is a tuning parameter. In combination with Δt_{It} sufficiently small it causes that there are enough data points at local minima of $v_{\mathrm{Lim,Map},\dot{v}}$ such that the trajectory does not exceed local minima of $v_{\mathrm{Lim,Map},\dot{v}}$.

The nonlinear approximation problem (5.10) can be solved with LM and NRBA. To an approximation of (5.10) RBA can be applied.

Advantage of solving the approximated optimization problem

Regarding the control, [82] states that the time domain is beneficial because the vehicle dynamics itself is almost linear except from the air resistance. Therefore the problem can easily be simplified to a linear constrained quadratic programming (QP) problem. In contrast, the control in space domain is very nonlinear when the distance to the vehicle ahead is considered and therefore harder to be solved. Due to the spatially defined upper speed limit, (5.10) is a nonlinear problem though.

However, RBA can solve an approximation of the nonlinear problem (5.10) by a quadratic problem. For this, in each iteration p the trajectory position s_{TJY} at the current time t_p is determined by temporal integration of the trajectory velocity v_{TJY} with its currently estimated control point vector $\boldsymbol{x}$ in order to derive the upper speed limit $v_{\mathrm{Lim,TJY}}(s_{\mathrm{TJY}}(t_p))$. Then $v_{\mathrm{Set}} = v_{\mathrm{Lim,TJY}}$ is provided to the linear WLS algorithm RBA. The linear algorithm has no knowledge of the nonlinearity and assumes that the error $v_{\text{Lim,TJY}} - v_{\text{TJY}}$ depends linearly on the control point vector $\boldsymbol{x}$.

The solution of RBA is not guaranteed to be optimal under these conditions. However, the effort is lower than with a method for NWLS and the solution of the approximated problem is satisfactory compared with an in general not globally optimal solution of the actual NWLS problem. For predictive vehicle control, [47, 28, 86] also approximated a nonlinear problem by a quadratic problem in order to be able to apply a quadratic programming method instead of a sequential quadratic programming method like LM.

Influence of temporal safety margin to upper speed limit

The upper diagram in Figure 5.2 depicts the velocity v versus the position s measured from the vehicle in driving direction. The dashed line shows the speed limit from map data with desired acceleration $v_{\text{Lim,Map},\dot{v}}$. The solid red line is a trajectory computed using RBA with $\Delta t_{\text{Lim,TJY}} = 1$ s, $\Delta t_{\kappa} = 1$ s, $\Delta t_{\text{It}} = 0.1$ s and $I = 1$, whereby I denotes the number of spline intervals. Each red dot is a set point of trajectory velocity v_{Set} that occurs during the iterations of RBA. For better view, only every tenth v_{Set} point is shown in Figure 5.2. The blue dots and lines show the same quantities for $\Delta t_{\text{Lim,TJY}} = 4$ s. The temporal parametrization causes that in the $v - s$ diagram the v_{Set} data points are less close at higher speeds (e.g. for $s \leq 200$ m) compared to at low speeds (e.g. around $s = 600$ m). Due to the temporal safety margin to upper speed limit in (5.12), the spatial distance between the courses of v_{Set} and $v_{\text{Lim,Map},\dot{v}}$ is larger at higher speeds. Provided that Δt_{It} is sufficiently small and that $\Delta t_{\text{Lim,TJY}}$ is sufficiently large, there are enough data points to cause that the resulting trajectory does not exceed $v_{\text{Lim,Map},\dot{v}}$ at local minima, e.g. at $s = 400$ m and $s = 600$ m.

Increasing $\Delta t_{\text{Lim,TJY}}$ also reduces the possibility of short-lasting velocity peaks. For example, at $s = 500$ m $v_{\text{Lim,Map},\dot{v}}$ has a local maximum that both v_{Set} with $\Delta t_{\text{Lim,TJY}} = 1$ s and the corresponding trajectory $v_{\text{TJY}}(t)$ reflect. For $\Delta t_{\text{Lim,TJY}} = 4$ s, both v_{Set} and $v_{\text{TJY}}(t)$ cannot maintain the local maximum of $v_{\text{Lim,Map},\dot{v}}$ at $s = 500$ m, however.

Since the course of v_{Set} is not jerk-free and must be smoothed by the trajectory, close proximity of $v_{\text{TJY}}(t)$ to v_{Set} is not intended in all situations. The extent to which the trajectory can follow v_{Set} is determined by the weighting factors R_v^{-1}, R_a^{-1}, R_j^{-1} and the temporal distance of neighboring knots Δt_{κ}. Furthermore, the temporal distance of neighboring data

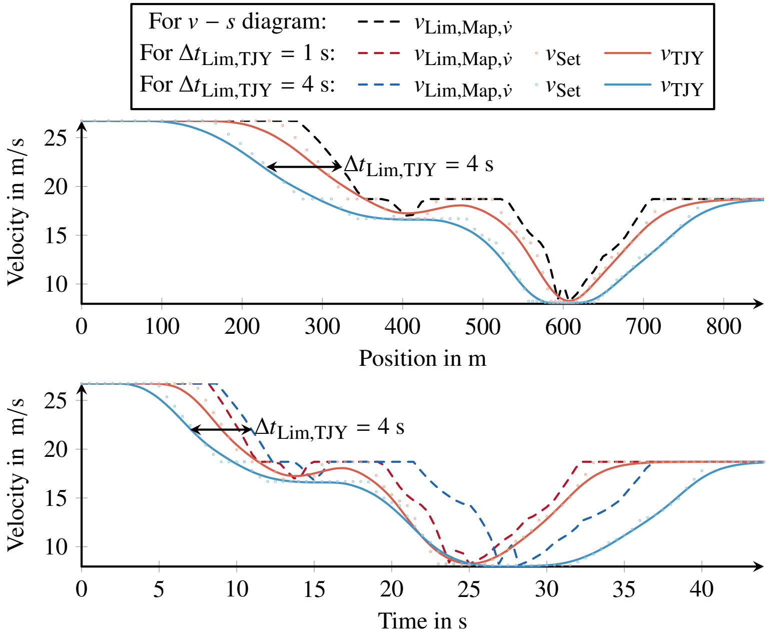


Figure 5.2: Influence of temporal safety margin to upper speed limit $\Delta t_{\mathrm{Lim,TJY}}$ on set point of trajectory velocity v_{Set} and trajectory velocity v_{TJY}. Only a subset v_{Set} is shown. Upper diagram: The speed limit from map data with desired acceleration $v_{\mathrm{Lim,Map},\dot{v}}$ is identical for both trajectories. Lower diagram: $v_{\mathrm{Lim,Map},\dot{v}}$ differs with the trajectories. *Adopted from [79].*

points Δt_{It} needs to be sufficiently small. Reducing Δt_{It} increases the computational effort of RBA linearly.

In the upper $v - s$ diagram of Figure 5.2 $v_{\mathrm{Lim,Map},\dot{v}}$ is identical for all trajectories. The lower diagram depicts the same quantities versus time. In a $v - t$ diagram $v_{\mathrm{Lim,Map},\dot{v}}$ needs to be shown individually for each trajectory because when two trajectories differ, in general they have a different trajectory position at the same time t, hence $v_{\mathrm{Lim,Map},\dot{v}}$ also differs between the two trajectories. *Adopted from [79].* A $v - s$ diagram is often more convenient for comparisons because of the identical $v_{\mathrm{Lim,Map},\dot{v}}$. However, the lower $v - t$ diagram in Figure 5.2 illustrates well that the temporal safety margin to upper speed limit leads to a constant time gap between the courses of v_{Set} and $v_{\text{Lim,Map},\dot{v}}$.

Moreover, because of the constant temporal density of degrees of freedom resulting from the temporal distance of neighboring knots Δt_κ, the curvature of the trajectory is independent of velocity in the $v - t$ diagram. In contrast, in the $v - s$ diagram the trajectory curvature usually decreases as velocity increases. Due to the equidistant knots with respect to time and (5.12), the trajectory can approximate the upper speed limit very close at low speeds because its degrees of freedom are dense with respect to position whereas at high speeds the approximation is less close.

Advantage of iterative approach of RBA

LM, RBA and NRBA compute the trajectory iteratively but in different ways. LM adapts all control points simultaneously, hence the whole trajectory is improved during an iteration. In contrast, RBA and NRBA process only a subset of the control points simultaneously and the temporal length of the trajectory increases with the number of iterations.

Figure 5.3 shows a $v_{\text{TJY}}(t)$ trajectory generated using RBA after 100, 200, 300 and 400 iterations with $\Delta t_\kappa = 2$ s, $\Delta t_{\text{It}} = 0.1$ s and $\Delta t_{\text{Lim,TJY}} = 2$ s. The iterative approach of RBA allows to pause the calculation and use the intermediate result before continuing the calculations. This possibility is beneficial in presence of restrictions on computation time. It is also known that the optimization is finished for the subtrajectory that can be computed from knots and control points that are not used by RBA anymore but have been shifted out of the KF.

Furthermore, with RBA the number of data points is not required to be bounded because of the shifting function. This is beneficial because a priori it is not known how much time data points t_p the trajectory needs in order to reach the end of the electronic horizon. Also the length of the electronic horizon usually increases as the navigation system is recalculating the route (c.f. Section 5.1). RBA can start trajectory optimization immediately parallel to the route recalculation and can add knots during the optimization process as needed.

In contrast, LM cannot change the knots. Hence, the number of needed knots has to be overestimated, which causes additional computational effort, because more control point values are computed.

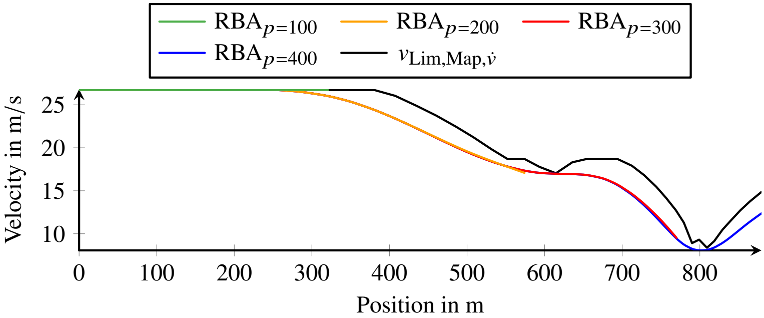


Figure 5.3: Iterative trajectory optimization using RBA: Speed limit from map data with desired acceleration $v_{\text{Lim,Map},\dot{v}}$ (black) and trajectory after 100 (green), 200 (yellow), 300 (red) and 400 (blue) iterations p.

Influence of temporal distance of neighboring knots and number of spline intervals

Figure 5.4 shows trajectories with $\Delta t_{\text{Lim,TJY}} = 1.5$ s and $\Delta t_{\text{It}} = 0.2$ s. Trajectories with temporal distance of neighboring knots $\Delta t_{\kappa} = 1$ s are depicted in red and trajectories with $\Delta t_{\kappa} = 5$ s in blue. For both cases trajectories computed by RBA with number of spline intervals $I = 1$ and $I = 5$ are compared to the LM solution. The solutions of both algorithms are similar and differences can mainly be identified from the acceleration diagram.

Increasing the temporal distance of neighboring knots Δt_{κ} reduces the possible oscillation of velocity in a given time interval. This effect can be seen in the acceleration diagram for $200 \text{ m} \leq s \leq 600 \text{ m}$. However, a large change of v_{TJY} as for $600 \text{ m} \leq s \leq 800 \text{ m}$ can still be represented by $\Delta t_{\kappa} = 5$ s through correspondingly larger differences of temporally less dense neighboring control point values. For $\Delta t_{\kappa} = 1$ s, the RBA solutions are closer to the LM solution than for $\Delta t_{\kappa} = 5$ s.

Increasing I also enables to smooth the trajectory because the filter can rework larger parts of the trajectory with hindsight when additional information is available. However, this can result in undesired velocity oscillations, e.g. for $800 \text{ m} \leq s \leq 1200$ s. This applies to both RBA and LM.

Furthermore, RBA trajectories with larger I are more likely to exceed $v_{\text{Lim,Map},\dot{v}}$. This slightly occurs at $s = 1200$ m where the RBA trajectory with $\Delta t_{\kappa} = 5$ s accelerates too early. The reason is a general flaw of the filter-

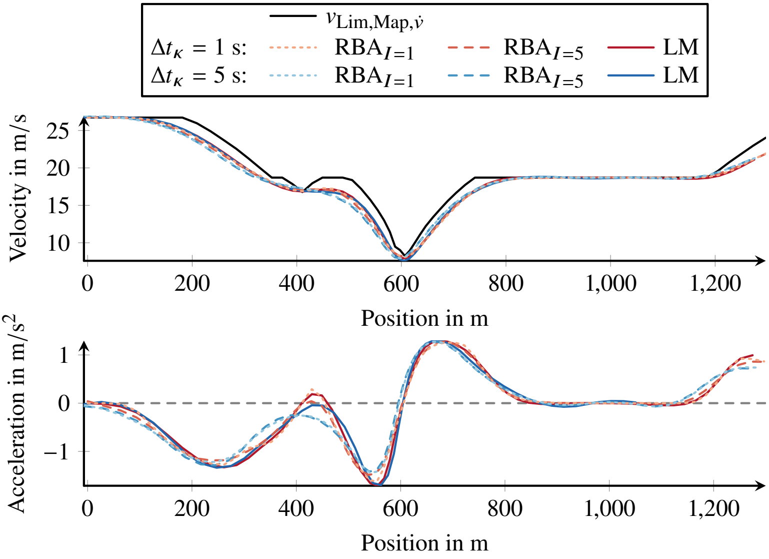


Figure 5.4: Influence of temporal distance of neighboring knots Δt_κ on trajectory optimization using RBA and LM and influence of number of spline intervals I on trajectory optimization using RBA.

based approach. When the filter adapts control points of previous spline intervals according to the knowledge of its state estimation covariance matrix, it does not compute any deviations or errors for these control points that could be penalized. Exceeding $v_{\text{Lim,Map},\dot{v}}$ can be avoided by increasing $\Delta t_{\text{Lim,TJY}}$ with I.

As LM evaluates the whole sum of squared errors in the approximated NWLS problem in each iteration, this problem does not occur with LM. At around s = 1200 m both LM trajectories are identical.

Although the above stated approach does not optimize the energy consumption by taking into account the electrical power explicitly, a reduction of energy consumption can be achieved implicitly by increasing the weight of acceleration error square R_a^{-1}. The method of reducing acceleration and braking was also chosen in [1, 107, 112]. According to the investigations in [81] it is less effective than using an energy consumption model. Instead its advantage is that a simpler optimization algorithm can be used.

5.4 Trajectory optimization considering electrical power

□ According to Section 4.4 low absolute values of the electrical traction power $P_{\text{trac,elec}}$ lead to low power losses. $P_{\text{trac,elec}}$ is described by the adaptive electrical power model (AEPM) proposed in Subsection 4.7.2. The model output P_{AEPM} is given by

$$P_{\text{AEPM}}(t) = \text{AEPM}(v_{\text{TJY}}(t), a_{\text{x}}). \tag{5.13}$$

The sensor longitudinal acceleration a_{x} from (4.22) is computed as

$$a_{\text{x}} = a_{\text{TJY}}(t) + g \cdot \sin(\alpha(s_{\text{TJY}}(t))) \tag{5.14}$$

and the road slope angle α can be derived from the road slope stated in percent according to map data using (4.2).

When considering the required electrical power in trajectory planning, (5.10) is augmented with the summand $R_P^{-1} \cdot (P_{\text{AEPM}})^2$ which penalizes absolute values of P_{AEPM}. R_P^{-1} denotes the weight of power error square and its reciprocal R_P is the variance of power measurement. The resulting optimization problem reads

$$\begin{aligned} \hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{p=1}^{P} \Big(& R_v^{-1} \cdot \left[v_{\text{Lim,TJY}} \left(s_{\text{TJY}}(t_p) \right) - v_{\text{TJY}}(t_p) \right]^2 \\ & + R_a^{-1} \cdot \left(a_{\text{des}} - a_{\text{TJY}}(t_p) \right)^2 + R_j^{-1} \cdot \left(j_{\text{TJY}}(t_p) \right)^2 \\ & + R_P^{-1} \cdot \left(P_{\text{AEPM}}(t_p) \right)^2 \Big) . \end{aligned} \tag{5.15}$$

As P_{AEPM} correlates with the product of v_{TJY} and its derivative a_{TJY}, an approximation of this fourth optimization goal such that RBA can be applied, seems not promising for good results. *Adopted from [79].*

Most nonlinear filter algorithms such as UKF or PF do not accept a measurement matrix that relates a control point vector to a measurement as input. Instead, they apply a given nonlinear measurement function to a set of control point vectors and use the resulting function values for estimating the relation between measurement and control points. However, for the first three optimization goals the exact values of the measurement matrix are known. These result from the values of the B-spline basis functions.

Estimating the B-spline basis functions and additionally performing the transition from a spatial to a temporal representation poses a challenge for the convergence of a nonlinear filter. Experiments were conducted with a square-root cubature Kalman filter (SCKF) stated in [6, 7]. The SCKF can be seen as a special case of the UKF. In experiments no weighting factor combination was found that yielded consistent convergence of the trajectory to the upper speed limit. This was also the case when the SCKF was applied to optimization problem (5.10), which does not consider electrical power.

The MPF is beneficial for problems that can be subdivided in a linear subproblem and a nonlinear subproblem. A KF solves the linear subproblem optimally using a given measurement matrix while a PF is applied to the nonlinear problem [155]. In the trajectory optimization application the nonlinear problem is the fourth optimization goal. The MPF converges for problem (5.15) reliably provided that the weighting factors are within a certain range. For these reasons, NRBA includes a MPF. However, even though the MPF comprises a KF, the convergence radius of the MPF is still noticably smaller than that of a pure KF and the MPF is much more sensitive to filter parameter variations.

□ Braking for an upcoming curve usually requires negative P_{AEPM} for recuperation. Strong penalization of P^2_{AEPM} can prevent braking for upcoming curves. In order to enable sufficient braking power, the computation of the error $e = v_{\mathrm{Lim,TJY}} - v_{\mathrm{TJY}}$ within the MPF is designed asymmetrically. If $e \leq 0$, the artificial measurement $v_{\mathrm{Lim,TJY}}$ is replaced with $v_{\mathrm{Lim,TJY}} = R_{\mathrm{v>vLim}} \cdot e + v_{\mathrm{TJY}}$. Hence, e is multiplied with $R_{\mathrm{v>vLim}}$ if v_{TJY} is above $v_{\mathrm{Lim,TJY}}$. In case of $e > 0$ the original e is multiplied with $R_{\mathrm{v<vLim}}$, the error weighting if v_{TJY} is below $v_{\mathrm{Lim,TJY}}$. In each case the resulting error square is afterwards weighted with R_v^{-1} in the MPF. *Adopted from [79].*

$R_{\mathrm{v>vLim}}$ cannot be chosen arbitrarily large. First, $R_{\mathrm{v>vLim}} > 2$ can prevent the MPF from converging to the upper speed limit. Second, during operation of the ALC situations can occur in which the vehicle exceeds $v_{\mathrm{Lim,Map},\dot{v}}$. An example is that the driver intervens and accelerates in order to overtake a slower vehicle. At its beginning the trajectory needs to indicate the current vehicle velocity. $R_{\mathrm{v>vLim}}$ should be only so large that a clearly noticable braking occurs until the vehicle is below $v_{\mathrm{Lim,Map},\dot{v}}$. For $R_{\mathrm{v>vLim}} > 2$ braking according to the computed trajectory is uncomfortably abrupt.

□ The asymmetrical weighting of the velocity error can also be used in combination with RBA but since there is no incentive to exceed $v_{\mathrm{Lim,Map},\dot{v}}$

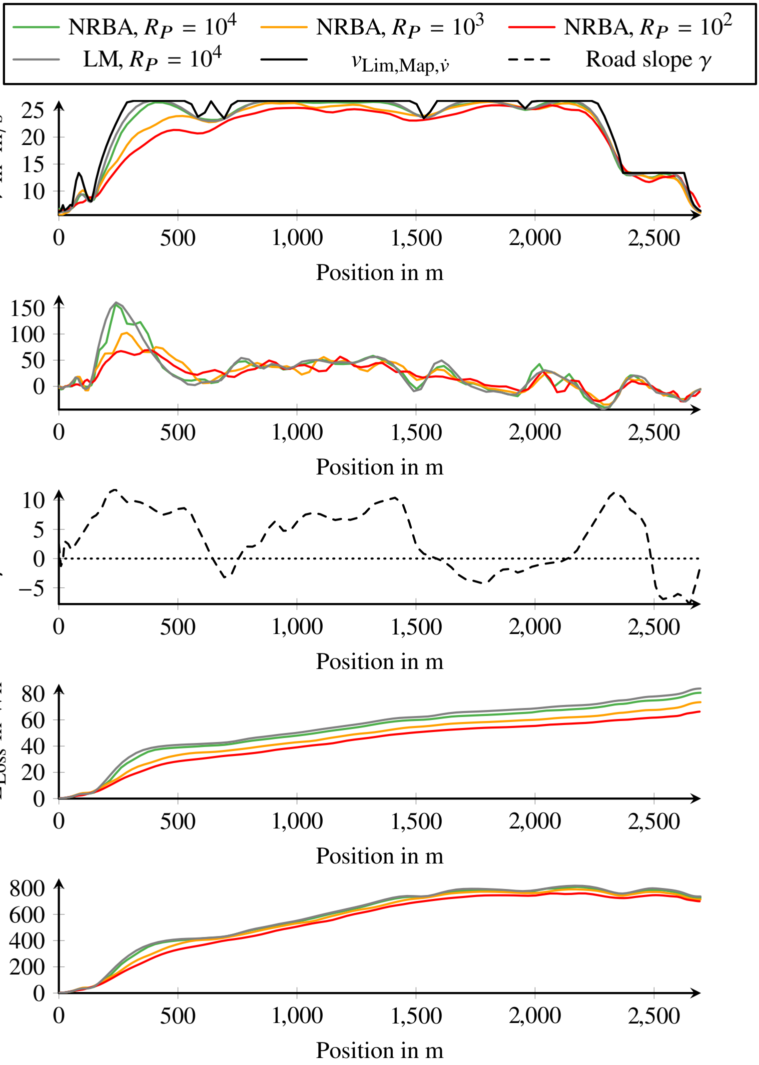


Figure 5.5: Relevant quantities of trajectories determined by NRBA and LM for different variance of power measurement. Velocity v, electrical traction power $P_{\mathrm{trac,elec}}$, road slope γ, energy loss between HV battery and wheel E_{Loss} and HV battery energy E_{Batt} for trajectories determined by NRBA and LM that differ in the variance of power measurement R_P. *Adopted from [79].*

because of the missing optimization goal regarding P_{AEPM}, the asymmetric weighting has very limited influence. *Adopted from [79].*

Influence of weight of power error square

□ By increasing the weight of power error square R_P^{-1} or by lowering the variance of power measurement R_P, respectively, the trajectory can be optimized towards low energy consumption. Figure 5.5 depicts various quantities for trajectories determined by NRBA with $R_P = 10^2$, $R_P = 10^3$ and $R_P = 10^4$. For comparison, the LM solution for $R_P = 10^4$ is also shown. The remaining trajectory parameters are $\Delta t_\kappa = 2$ s, $\Delta t_{\mathrm{It}} = 0.25$ s, $\Delta t_{\mathrm{Lim,TJY}} = 1$ s and $I = 1$.

In the depicted situation, all tractories indicate roughly the same velocity at the start position and also at the end position. Hence, the kinetic and potential energy of a vehicle at the beginning and end is roughly the same regardless of which trajectory it tracks. Therefore the energy consumptions resulting from following each of the trajectories can be compared.

For $R_P = 10^4$ the depicted quantities for the NRBA and LM trajectories are very similar and both trajectories follow $v_{\mathrm{Lim,Map},\dot{v}}$ closely. With LM the energy loss between HV battery and wheel (E_{Loss}) is $E_{\mathrm{Loss}} = 77$ Wh and the required HV battery energy (E_{Batt}) is $E_{\mathrm{Batt}} = 721$ Wh. For $R_P = 10^4$ NRBA achieves $E_{\mathrm{Loss}} = 75$ Wh and $E_{\mathrm{Batt}} = 714$ Wh.

When R_P is lowered, peaks in the electrical traction power $P_{\mathrm{trac,elec}}$ are reduced which translates to less energy loss between HV battery and wheel as well as less HV battery energy. $R_P = 10^3$ leads to $E_{\mathrm{Loss}} = 71$ Wh and $E_{\mathrm{Batt}} = 703$ Wh. For $R_P = 10^2$ $E_{\mathrm{Loss}} = 65$ Wh and $E_{\mathrm{Batt}} = 684$ Wh result.

Regarding the driving style it can be observed that with lower R_P, the trajectory exhibits lower acceleration, stays more below $v_{\mathrm{Lim,Map},\dot{v}}$, avoids short-lasting velocity peaks and tends to oscillate more. *Adopted from [79].* The oscillations can mainly be explained by the larger slope variations which affect $P_{\mathrm{trac,elec}}$ but they partly also result from the fact that the MPF estimations are more strongly based on the state-space sampling of the PF than on the KF. With LM no energy savings were achieved by lowering R_P, even after varying LM parameters. Instead, the trajectory diverged which indicates a narrow convergence radius of the unapproximated nonlinear optimization problem.

□ During the acceleration phase up to $s = 1000$ m the NRBA does clearly avoid peaks in $P_{\mathrm{trac,elec}}$ by reducing the acceleration at the slope. However, this does apply to the deceleration phase not to such an extent, even under consideration that from $s = 2200$ m on there is a high road grade which avoids the need for large negative $P_{\text{trac,elec}}$. The filter could still smooth power demand more by decelerating earlier. It was also oberserved during road sections with negligible slope that NRBA decelerates too late and therefore the asymmetrical weighting of the velocity error was added. *Adopted from [79].*

As shown in Subsection 3.3.4 and Subsection 3.4.4, both novel filter based methods RBA and NRBA can reproduce a symmetrical approximation of WLS and LM, respectively. Provided that I is large enough, the filter lag is negligible with RBA. When the nonlinear optimization goal is weighted heavily with NRBA, a filter lag occurs, solutions become more oscillating and increasing I is not as unambigiously beneficial as with RBA. In the situation depicted by Figure 5.5, increasing I mainly leads to more oscillating trajectories whereas no significant change in deceleration can be noticed. Hence, the energy-saving potential of NRBA during recuperation is lower than expected.

Increasing I should facilitate early deceleration because NRBA can adapt larger parts of the trajectory with hindsight when data that involves the lower speed limit is processed. MPF improvements that achieve the same approximation quality with less particles as stated in [203] would be beneficial for good results with larger I. They could help to overcome the flaw concerning deceleration because they enable to keep the state-space sampling density in each dimension constant when I is increased.

At an early stage, at which the trajectory representation was not yet temporal, the AEPM hyperparameters were not final and NRBA still included the SCKF instead of the MPF, automated tests for finding suitable target criteria weightings were done by A. Thorgeirsson [173] as a student assistant.

5.5 Considering trajectory constraints

After each iteration of the trajectory optimization, a lower limit on the trajectory velocity v_{TJY} is enforced and the trajectory acceleration a_{TJY} is restricted. This ensures that after any iteration a feasible trajectory is available. Due to the high execution frequency the formulae are rather simple, avoid evaluating the B-spline function and take advantage of the convex hull property.

v_{TJY} is restricted to nonnegative values, which translates to restricting all components of the state estimate $\hat{\boldsymbol{x}}$ to nonnegative values:

$$\hat{\boldsymbol{x}} = \min\left(\hat{\boldsymbol{x}}, \mathbf{0}\right) \tag{5.16}$$

Additionally the differences between neighboring components $\hat{x}$ of $\hat{\boldsymbol{x}}$ are limited according to

$$\begin{aligned} \hat{x}_i &= \max\left(\hat{x}_{i-1} + a_{\text{des,neg}} \cdot \Delta t_\kappa, \min\left(\hat{x}_i, \hat{x}_{i-1} + a_{\text{des,pos}} \cdot \Delta t_\kappa\right)\right), \\ i &= 2, 3, \ldots \end{aligned} \tag{5.17}$$

$a_{\text{des,pos}}$ denotes the desired positive longitudinal acceleration and $a_{\text{des,neg}}$ is the desired negative longitudinal acceleration. The approach in (5.17) overestimates the absolute value of trajectory acceleration using the convex hull property of the B-spline function. Hence, (5.17) might in some situations adjust control points although the specified acceleration limits are not yet exceeded.

The following subsections state further constraints which are taken into account partly as *hard constraints* and partly as *soft constraints*. Hard constraints are constraints that reduce the solution space of an optimization problem and can lead to an empty set of feasible solutions. In contrast, the term soft constraints means that the optimization function is chosen such that violating these soft constraints leads to very bad function values with the aim that such solutions are avoided by the optimization method.

5.5.1 Adaption to vehicle motion state

□ Small deviations between trajectory velocity v_{TJY} and vehicle velocity v_{Vhcl} always occur because of an imperfect control. However, there are several cases in which both quantities differ significantly from each other. Examples are driving off from a parking position and when the driver stops overriding the ALC by pressing accelerator or braking pedal.

If the deviation exceeds a tolerated threshold, a new trajectory should be planned. The velocity and acceleration at the trajectory beginning should be adapted to the vehicle motion state given by vehicle velocity v_{Vhcl} and vehicle longitudinal acceleration $\dot{v}_{\text{Vhcl,x}}$. Hence, the constraints

$$\begin{aligned} v_{\text{TJY}}(t_1 = 0) &= v_{\text{Vhcl}} \\ a_{\text{TJY}}(t_1 = 0) &= \dot{v}_{\text{Vhcl,x}} \end{aligned} \tag{5.18}$$

shall be enforced. Several methods for taking into account state constraints during Kalman filtering are presented in [163]. The state projection method can modify the control points such that the trajectory fulfills certain hard equality constraints at single points in time. *Adopted from [79].* The state projection method projects an unconstrained estimate $\hat{\boldsymbol{x}}_p^+$ onto the hyperplane defined by the constraints $\boldsymbol{D}\boldsymbol{x} = \boldsymbol{d}$. Usually $\boldsymbol{D}\boldsymbol{x} = \boldsymbol{d}$ is an underdetermined equation system. The constrained estimate $\tilde{\boldsymbol{x}}_p^+$ is given by

$$\tilde{\boldsymbol{x}}_p^+ = \hat{\boldsymbol{x}}_p^+ - \boldsymbol{D}^\top \left(\boldsymbol{D}\boldsymbol{D}^\top\right)^{-1} \left(\boldsymbol{D}\hat{\boldsymbol{x}}_p^+ - \boldsymbol{d}\right). \tag{5.19}$$

The difference of the projected control points need to be checked against the desired longitudinal acceleration. If they exceed the acceleration limits $a_{\text{des,neg}}$ or $a_{\text{des,pos}}$, the projection needs to be repeated with additional constraints that refer to differences between neighboring control point values.

□ The state projection method changes only the control points that influence the trajectory function at $t_1 = 0$. If the function is projected onto a vehicle at standstill and the temporal distance of neighboring knots Δt_κ between knots is small, the trajectory will demand an abrupt acceleration towards the upper speed limit unless control points that refer to following spline intervals are adapted too.

A jerk-free and comfortable velocity transition can be achieved by modifying in the optimization problems (5.10) and (5.15) the desired acceleration value $a_{\text{Set}}(t_p)$ from (5.11) and the acceleration weight R_a^{-1} in the first filter iterations as follows:

$$\begin{aligned} R_a^{-1} &= R_{a,t_0}^{-1} + (R_a^{-1} - R_{a,t_0}^{-1})/\Delta t_{\text{VhclAdapt}} \cdot \min(\Delta t_{\text{VhclAdapt}}, t), \\ a_{\text{Set}}(t_p) &= \dot{v}_{\text{Vhcl,x}} + (0 - \dot{v}_{\text{Vhcl,x}})/\Delta t_{\text{VhclAdapt}} \cdot \min(\Delta t_{\text{VhclAdapt}}, t). \end{aligned} \tag{5.20}$$

$\Delta t_{\text{VhclAdapt}}$ and R_{a,t_0}^{-1} are tuning parameters. From an initially strong weighting R_{a,t_0}^{-1} of $a_{\text{Set}} = \dot{v}_{\text{Vhcl,x}}$ at $t = 0$, the quantities R_a^{-1} and a_{Set} return to their standard values in a linear fashion until $t_0 + \Delta t_{\text{VhclAdapt}}$. *Adopted from [79].*

Figure 5.6 illustrates the individual effects of projection onto vehicle motion state with (5.19), enforcement of acceleration constraints using (5.17) and adaption of filter settings according to (5.20) on the trajectory. The trajectory parameters are $\Delta t_\kappa = 2$ s, $\Delta t_{\text{It}} = 0.1$ s, $\Delta t_{\text{Lim,TJY}} = 2$ s and $I = 1$.

In the depicted situation, the speed limit from map data with desired acceleration $v_{\text{Lim,Map},\dot{v}}$ is constant at 26.7 m/s (black solid line in upper di-

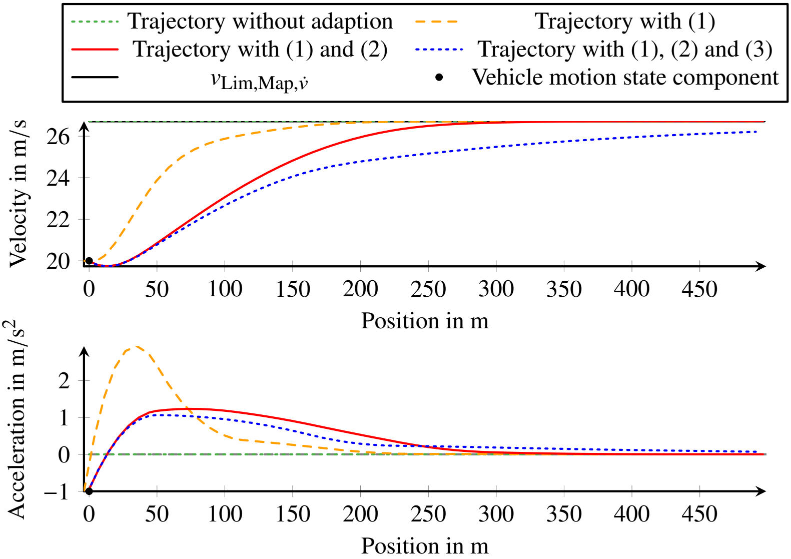


Figure 5.6: Individual effects of projection onto vehicle motion state using (5.19) (1), enforcement of acceleration constraints with (5.17) (2) and adaption of filter settings using (5.20) (3) on the trajectory during trajectory adaption to vehicle motion state.

agram). The vehicle motion state consists of the vehicle velocity $v_{\mathrm{Vhcl}} = 20\ \mathrm{m/s}$ and the vehicle longitudinal acceleration $\dot{v}_{\mathrm{Vhcl,x}} = -1\ \mathrm{m/s^2}$. Each component is indicated by a black dot in the corresponding diagram. A trajectory without adaption (depicted by green dotted line) starts at $v_{\mathrm{Lim,Map},\dot{v}}$ instead of v_{Vhcl}. A trajectory that is only projected onto the vehicle state (orange dashed line) can demand strong accelerations since the projection only changes the first four control points. The red solid line results from additionally enforcing acceleration constraints with (5.17). The adaption of the filter setting using (5.20) with $R_{a,t_0}^{-1} = 0.01$ and $\Delta t_{\mathrm{VhclAdapt}} = 150$ s allows to get a more comfortable transition.

The desired effect could also be achieved using a sigmoid function. However, the parameters of the sigmoid function need to be chosen carefully because filter sensitivity differs strongly with the magnitude of R_a^{-1}. For simplicity a linear transition with rather high $\Delta t_{\mathrm{VhclAdapt}}$ was used.

5.5.2 Adaption to previous trajectory

$v_{\text{TJY}}(t_1)$, $a_{\text{TJY}}(t_1)$ and $j_{\text{TJY}}(t_1)$ with $t_1 = 0$ of a new trajectory can also be projected onto v_{TJY}, a_{TJY} and j_{TJY} of the current trajectory at its current evaluation point t^* in order to achieve a C^2 continuous connection between both trajectories.

When connecting two trajectories using state projection, the knot positions of the new trajectory are additionally chosen such that the distance of t_1 to the neighboring knots equals the distances of t^* to the neighboring knots of the previous trajectory. Then the shape of the new trajectory around t_1 is identical to that of the previous trajectory around t^* and uncomfortable velocity oscillations at the joint are avoided.

Figure 5.7 depicts various trajectories with $\Delta t_\kappa = 1$ s, $\Delta t_{\text{Lim,TJY}} = 1$ s, $\Delta t_{\text{It}} = 0.1$ s and $I = 3$. At $t^* = 14.333$ s or $s_{\text{TJY}}(t^*) = 201.75$ m, respectively, replanning occurs and the new trajectory needs to be connected to the current trajectory depicted in green. The new trajectory depicted in orange is not projected and therefore the connection is not continuous. The new trajectory depicted in red is projected and a C^2 continuous connection to the current trajectory is achieved. However, shortly behind the connection point the courses of its derivatives deviate from the corresponding courses of the current trajectory. The blue trajectory results when additionally the control point vector of the new trajectory is initialized with the corresponding control point values of the current trajectory. The deviations in the courses of the derivatives are significantly reduced.

However, in case of $I = 1$ the second derivative still deviates with this approach. This is because the estimate cannot be improved with hindsight using the knowledge stored in the KF covariance matrix that results from additional data points.

5.5.3 Adaption to vehicle ahead

The Intelligent Driver Model (IDM) described in [174] provides an ACC functionality by computing a desired vehicle acceleration a_{IDM} given by

$$a_{\text{IDM}} = a_{\text{free}} + a_{\text{int}}. \tag{5.21}$$

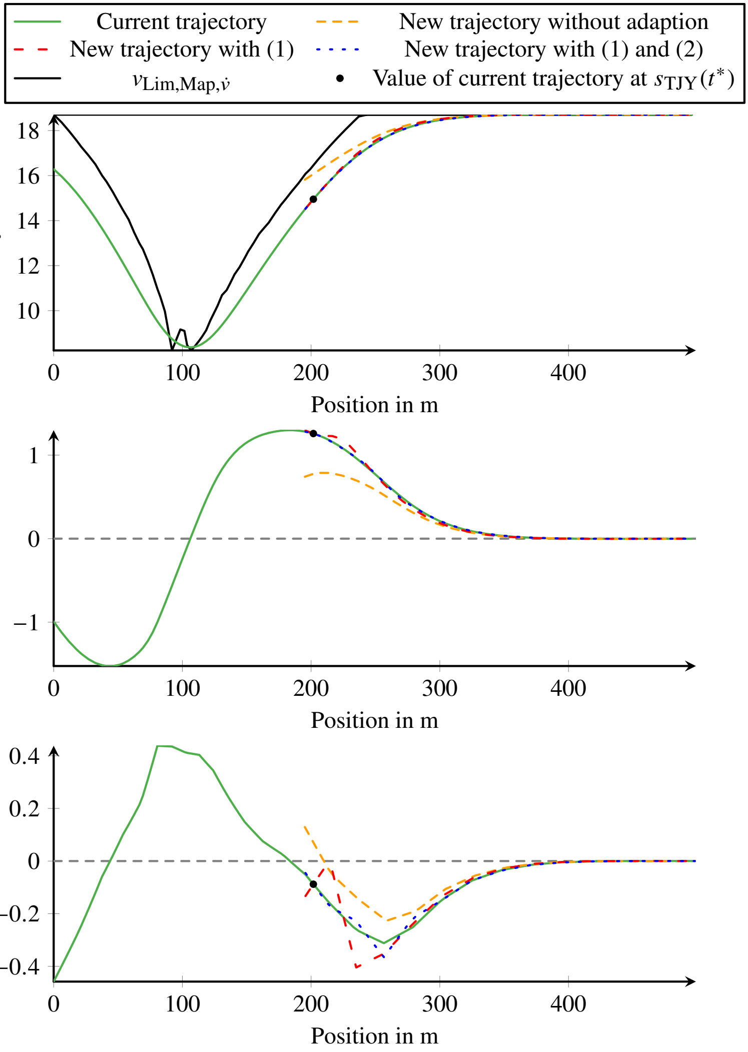


Figure 5.7: Individual effects of projection (1) and improved control point initialization (2) on the new trajectory during adaption to current trajectory at evaluation point $s_{\text{TJY}}(t^*)$ of current trajectory.

The free-road acceleration term a_{free} with

$$a_{\text{free}} = a_{\text{des,pos}} \left[1 - \left(\frac{v_{\text{Vhcl}}}{v_{\text{Set}}} \right)^{\delta} \right] \tag{5.22}$$

determines how the vehicle velocity v_{Vhcl} converges to the set velocity v_{Set} without a vehicle ahead. $a_{\text{des,pos}}$ is the desired positive longitudinal acceleration. The acceleration exponent δ is usually 4. The interaction acceleration term a_{int} with

$$a_{\text{int}} = a_{\text{des,neg}} \left(\frac{\Delta s_{\text{des}}}{\Delta s_{\text{VhclAhead}}} \right)^{2} \tag{5.23}$$

causes the vehicle to decelerate in order to adjust the distance to the vehicle ahead $\Delta s_{\text{VhclAhead}}$ to the desired minimum distance Δs_{des} given by

$$\Delta s_{\text{des}} = \Delta s_{\text{Jam}} + \Delta t_{\text{VhclAhead}} \cdot v_{\text{Vhcl}} + \frac{v_{\text{Vhcl}} \cdot (v_{\text{Vhcl}} - v_{\text{VhclAhead}})}{2\sqrt{a_{\text{des,pos}} \cdot |a_{\text{des,neg}}|}}. \tag{5.24}$$

$\Delta s_{\text{VhclAhead}}$ and the velocity of the vehicle ahead $v_{\text{VhclAhead}}$ are measurements of the radar sensor. Δs_{Jam} denotes the desired traffic jam distance with a typical value of 2 m. $\Delta t_{\text{VhclAhead}}$ is the desired time gap to the vehicle ahead and $a_{\text{des,neg}} < 0$ the desired negative longitudinal acceleration.

v_{Set} in (5.11) can be modified with a_{IDM} during the iterative solution of (5.10) and (5.15) such that the trajectory optimization process takes into account the vehicle ahead as a soft constraint. The radar sensor data $\Delta s_{\text{VhclAhead}}$ and $v_{\text{VhclAhead}}$ refers to the new trajectory at start time $t_1 = 0$. Under the assumption that $v_{\text{VhclAhead}}$ is constant and that the ego vehicle will follow the planned trajectory perfectly ($v_{\text{Vhcl}}(t) = v_{\text{TJY}}(t), \forall t \geq 0$), $\Delta s_{\text{VhclAhead}}$ can be calculated in following iterations as the sum of the initial distance $\Delta s_{\text{VhclAhead}}(t_1)$ and $v_{\text{VhclAhead}}(t_1) \cdot t$, from which the trajectory position s_{TJY} is subtracted:

$$\Delta s_{\text{VhclAhead}}(t_p) = \Delta s_{\text{VhclAhead}}(t_1) + v_{\text{VhclAhead}}(t_1) \cdot t_p - s_{\text{TJY}}(t_p) \tag{5.25}$$

In order to avoid dividing by small values, the IDM interaction free road term from (5.22) with velocity dependent $a_{\text{des,pos}}$ according to the acceleration tables and with $v_{\text{Set}} = v_{\text{TJY}}$ is used only if v_{TJY} is larger than 2 m/s. For $v_{\text{TJY}} \leq 2$ m/s, a_{free} is set to the trajectory acceleration a_{TJY}. The interaction term remains as in (5.23).

v_{Set} is determined depending on a hysteresis with respect to a_{int}: If $a_{\mathrm{int}} > -0.05\ \mathrm{m/s}^2$, v_{Set} for trajectory optimization is computed according to (5.11). If $a_{\mathrm{int}} < -0.15\ \mathrm{m/s}^2$, braking because of a vehicle ahead is required and a_{IDM} is used to correct v_{Set} downwards:

$$v_{\mathrm{Set}} = \max\left(0, \min\left(v_{\mathrm{TJY}} + \Delta t_{\mathrm{It}} \cdot a_{\mathrm{IDM}}, v_{\mathrm{Lim,TJY}}\right)\right) \tag{5.26}$$

In experiments it turned out beneficial to let a_{Set} remain at zero. With these modifications the free-road term in (5.22) causes a comfortable acceleration or deceleration towards the current v_{TJY} coming from the upper speed limit.

The minimum operation in (5.26) ensures that v_{TJY} can decrease fast enough. Without this minimum operation, the IDM can weaken the braking for a tight upcoming curve when there is no vehicle ahead. This is because (5.22) causes a deceleration towards a lower velocity set point v_{Set} that is sufficient and comfortable on highways but too weak for country roads, on which v_{Set} can vary strongly and deviations between v_{Vhcl} and v_{Set} should be minimal.

Figure 5.8 shows a situation in which the IDM adapts the trajectory using (5.26). If the vehicle trajectory tracks the planned trajectory and the velocity of the vehicle ahead remains constant, the specified desired time gap to the vehicle ahead will result.

5.6 Scientific contribution

DM are popular for automotive applications and have great potential in combination with DP as their orthogonal features can complement each other. However, the exponential growth of computational effort with increasing time horizon limits the application of DM to short time horizons.

In general the resulting static optimization problem (2.7) in the DM approach is nonlinear and solved by sequential quadratic programming (SQP) techniques or interior point methods [187]. Frequently trajectory optimization is applied to vehicles with combustion engine or a hybrid power train, both of which have lots of degrees of freedom and constraints. For example, if the optimization problem incorporates the gear selection it is a mixed integer problem.

In comparison, BEVs often have a power train with 1-speed gear box which simplifies the optimization problem. Quadratic problems are a subset of nonlinear problems that can be optimized more efficiently with QP

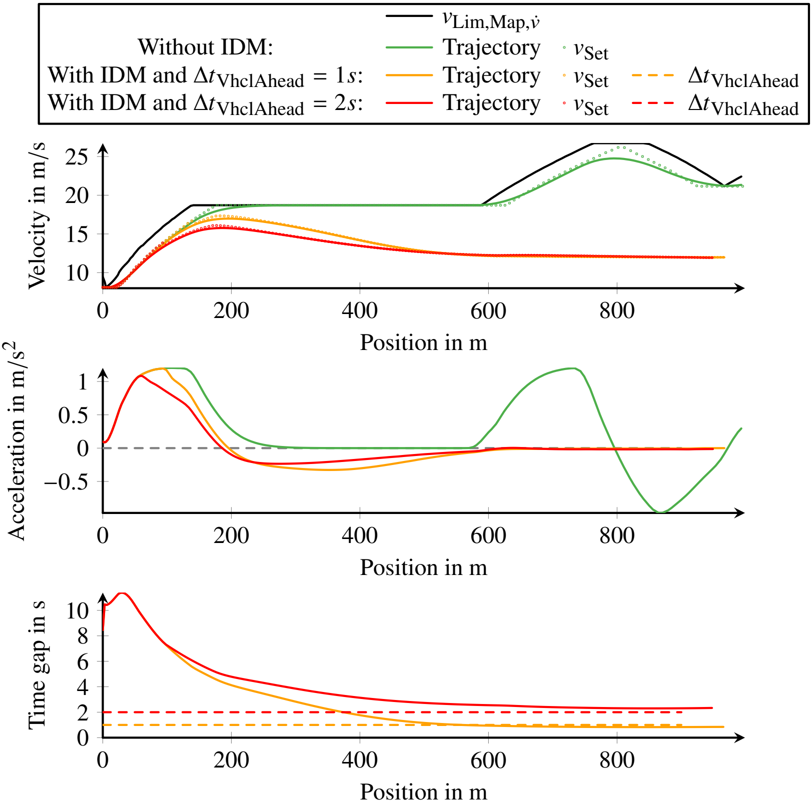


Figure 5.8: Effect of Intelligent Driver Model (IDM) on set point of trajectory velocity v_{Set} and resulting trajectory depending on the chosen desired time gap to the vehicle ahead $\Delta t_{\mathrm{VhclAhead}}$ during trajectory adaption to a vehicle ahead with a velocity of 12 m/s.

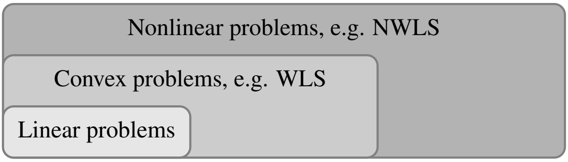


Figure 5.9: Classes and examples of optimization problems

methods because setting the derivative of the convex optimization function to zero results in sufficient conditions for the global optimum.

The common linear WLS approximation problem is an unconstrained quadratic optimization problem and the NWLS approximation an unconstrained nonlinear optimization problem. Figure 5.9 depicts different classes of optimization problems.

This work presents a DM based trajectory optimization approach for an ALC of a BEV with fixed gear ratio such as the considered research vehicle. With the presented approach, the computational effort only grows linearly with the number of function parameters or the time horizon. This substantial saving is achieved by formulating the trajectory optimization problem as either a WLS approximation problem or a NWLS approximation problem. The approximation problem is solved iteratively by RBA or NRBA, respectively. Each iterative method includes an iterative state estimator, also known as *filter*.

Usually filters solve unconstrained linear and nonlinear WLS problems. Therefore limitations of the vehicle and restrictions of the environment enter the trajectory optimization problem in the presented approach mainly as soft constraints. This means that the objective function is designed such that undesired solutions coincide with comparatively very bad values of the objective function. However, some hard constraints are also enforced using the state projection method from [163]. Hard constraints narrow the solution space of the optimization problem.

The trajectory is defined by a B-spline function and describes the desired vehicle velocity with respect to time. The trajectory results from an iterative solution of the approximation problem. Spatially or temporally defined soft constraints or optimization goals referring to derivatives of the function or its integral can be taken into account during the optimization process. For example, a spatial upper limit on the velocity defined by the trajectory results from the legal speed limit or tight curves.

Treating such spatial constraints as soft constraints leads to a NWLS problem. However, the iterative trajectory optimization approach allows to approximate this problem in each time step by a WLS problem similar to [28, 47], who approximated a nonlinear problem by a quadratic problem in order to be able to apply a QP method instead of a SQP method. The linear state estimator for WLS problems enables large savings in computational effort compared to the nonlinear state estimator for NWLS problems.

The energetic optimization of the trajectory is a case in which the approximation with a WLS problem cannot be expected to produce acceptable results. Then a nonlinear state estimator can be applied as demonstrated.

The presented iterative trajectory optimization approach offers the following additional advantages:

First, the number of function parameters does not need to be bounded. Instead, the iterative estimator can determine additional function parameters if the temporal length of the trajectory needs to be increased. As this does not influence the computational effort in each iteration of the estimator, arbitrarily long trajectories can be planned.

Second, the temporal length of the trajectory increases with the iterations and after each iteration an intermediate result is available for use. Therefore, the optimization can be paused and continued in accordance with the time constraints.

Differentation from previous works

The previous project *e-generation* [22, pp. 29-35] uses a spatially defined velocity trajectory represented by a cubic polynomial. The polynomial is iteratively adapted to map data using a KF in such a way that an optimization problem similar to (5.3) but with additional target criteria referring to the first two derivatives is solved.

Due to few degrees of freedom of the cubic polynomial, only a trajectory with simple shape can be represented. The spatially defined velocity trajectory exhibits the flaws mentioned in Section 5.2. Especially the reluctant behavior at low velocities is noticable in test drives. A target criterion referring to the electrical traction power or the consumed energy is not included in the optimization problem.

The approach presented in this chapter ensures nonnegative trajectory velocities at the spline function level by restricting its control point values via

(5.16). In contrast, the nonnegativity of the polynomial is not ensured, only the result of the polynomial evaluation is restricted to nonnegative values.

Furthermore, the previous approach adapts the trajectory only roughly to the vehicle velocity and acceleration by modification of the target values in the approximation, whereas in (5.18) an exact adaption via parameter projection is done.

An adaption to a previous trajectory as in Subsection 5.5.2 is not needed with the polynomial function and constraints resulting from a vehicle ahead as in Subsection 5.5.3 were not considered.

6 Automated energy-efficient longitudinal control

Section 6.1 of this chapter states the stages of the development process of the ALC followed by a description of the architecture of the ALC and its components in Section 6.2. Section 6.3 investigates the influence of selected parameters on the ALC energy-saving potential in simulations. Section 6.4 summarizes the technical contribution of this work concerning ALC.

6.1 Development process

The development of the ALC in the project *e-volution* followed the V model, which describes the software development process in the shape of the letter V and is depicted in Figure 6.1. The V model is widely used in the automotive industry [152, p. 25]. ISO 26262 states requirements for the development process of safety critical components and systems in vehicles. The procedure described in ISO 26262 is oriented towards the V model [187, pp. 110-111].

In the descending branch of the V model the customer requirements are analyzed and translated into a logical architecture from which a technical architecture is derived. In subsequent steps this technical architecture is decomposed into systems and components. During this process on each level specifications and test cases are defined for later review of the development steps. The last step of the descending branch coincides with the first step of the ascending branch and includes the implementation of the specified components.

The ascending branch comprises subsequent testing and integration steps from individual components up to testing the customer acceptance on the whole system.

Each step of the descending branch defines specifications that have to be met during verification on the same level of the ascending branch as indicated by dashed arrows in Figure 6.1 [187, pp. 163-168].

The V model distinguishes four different test steps. The first three are component test, integration test and system test. These are verificiation

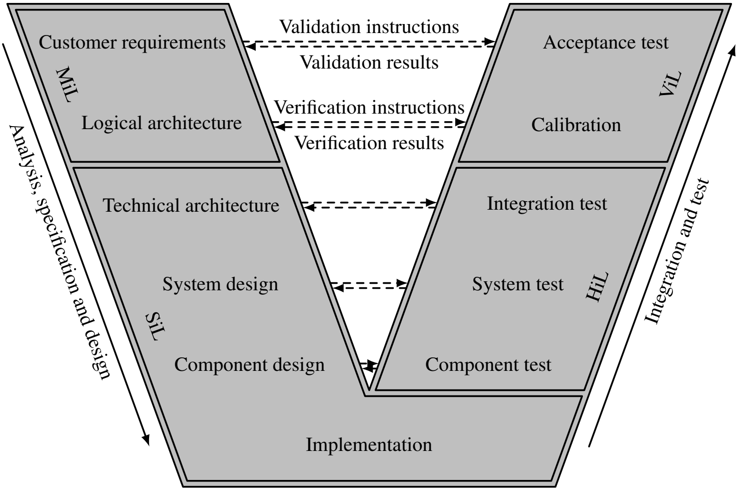


Figure 6.1: Software development process according to V model with allocation of in-the-Loop methods according to [187, p. 165].

tests. Verification denotes the process of evaluating whether an implementation meets the specified requirements for the corresponding development phase. Only the customer acceptance test is a validation test that determines whether all customer requirements are fulfilled. Methodological additions to the V model such as rapid prototyping enable validation at an early stage and help to avoid time-consuming and costly reworking loops across various stages of the V model [152, pp. 33, 152-154].

Rapid prototyping denotes process steps that enable early verification or validation of specifications. Typical process steps are modelling, simulation, integration and test of the prototype in the vehicle [152, pp. 160-162]. Model-based methods can be applied for specification of software functions as well as validation of specifications. Model-based software development allows to model software functions in a graphic environment using block diagrams [152, pp. 31-32]. A model compiler can compile the software functions for various target hardware such that the functions can be simulated as well as tested in the vehicle [152, p. 208]. If rapid prototyping methods support the development process, a virtual integration of the system is possible at the end of the descending branch. Before the integration steps this virtual prototype can be evalutated in simulated test drives [187, pp. 163-168].

In-the-Loop methods allow to combine models or real components at each development step with a reproduction of their real environment in order to obtain an assessable system. As there are no real components until the implementation phase of the V model, a simulation environment is used for virtual integration. Figure 6.1 allocates different in-the-Loop methods to the steps of the development process.

Model-in-the-Loop (MiL) enables a confirmation of the specification of customer requirements up to the logical architecture. The created model-based algorithms do not yet refer to the hardware of the target system. By transferring the model-based algorithms into a simulation environment that is hardware independent but exhibits already technical characteristics similar to that of the target system one can perform Software-in-the-Loop (SiL) which allows for an assurance up to individual components. Hardware-in-the-Loop (HiL) denotes methods that transfer models from the SiL environment to real components. For example a function can be executed on a vehicle ECU while the system architecture of the vehicle is simulated. This allows to verify the interaction of the ECU with other simulated vehicle components [187, pp. 163-168]. HiL simulation usually requires real-time capable components [152, pp. 297-301]. When HiL methods are applied consequently, the entire system exists in real components and can be verified up to the logical architecture. The Vehicle-in-the-Loop (ViL) method is especially useful for the development of driver assistance systems as it allows the operation of a real test vehicle in a virtual environment. Vehicle and virtual environment can be linked by either replacing the real sensors with interfaces to the virtual environment or by stimulating real sensors artificially.

In the research project *e-volution*, the customer requirement was stated as an ALC that is as comfortable and reliable as its predecessor from the previous project and offers at least the same dynamic driving style as the ALC that was developed during the previous project *e-generation* for the same research vehicle. As main enhancements the *e-volution* ALC was required to be capable of planning of longer, more far-sighted trajectories and of realizing an energy-efficient driving style.

The specifications of logical and technical architecture were taken over from *e-generation*. The logical architecture can be summarized as a system that controls motor torque according to map data. The technical architecture defines the interfaces of the ALC to the vehicle, for example which CAN messages are sent and received. Furthermore, it specifies the interfaces to the driver, e.g. how the driver can control the ALC using buttons next to the steering wheel and which information about the ALC state is displayed by the instrument cluster.

System design specifies the software architecture that implements the ALC functionality. The functionality is divided into different components whose interfaces are defined. System design was conducted during *e-generation* and reused for *e-volution*. □ The system design is specified by a framework in the model-based development environment MATLAB Simulink by MathWorks. The framework was created by an employee of the Porsche Engineering Services GmbH. It also takes care of communication with other vehicle components via several CAN busses and converts CAN messages into physical quantities. Furthermore, it checks the torque demand of the ALC system output for plausibility. This allows a safe vehicle operation in conjunction with an agile development of the ALC. *Adopted from [79].*

Design of components converts the black-box description of the system design into a white-box description for each module including the internal data flow such that each module can be implemented [187, pp. 168-173]. Within *e-volution*, component design led to the approximation algorithms RBA and NRBA and the AEPM. These components were implemented and tested in MATLAB. Then the trajectory planning based on RBA and NRBA was developed and tested in MATLAB using the open-loop vehicle reference model mentioned in Section 4.5 to investigate its energy-saving capabilities.

For further integration steps these functions were included into the Simulink framework. Translating MATLAB functions into sequences of Simulink blocks seemed not be advantageous. Therefore these functionalities were integrated into the Simulink framework as embedded MATLAB functions. As a result, each of the ALC components is based on few Simulink blocks that contain embedded MATLAB functions. Around the framework several simulink blocks including the closed-loop vehicle model from Section 4.5 were added in order to enable SiL tests of individual components and the whole ALC system. Extensions for considering constraints stated in Section 5.5, such as the projection of the trajectory as well as taking into account the vehicle ahead, were almost exclusively developed within the Simulink framework because of the closed control loop.

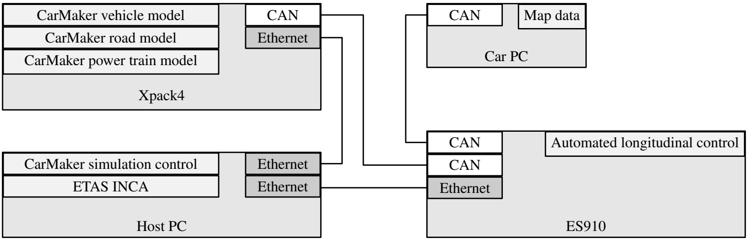


Figure 6.2: Simplified illustration of HiL architecture according to [22, p. 66]

Additionally, a HiL test bench was used for component, system and integration tests as well as calibration. The HiL test bench was created by B. Fath during *e-generation* [22, pp. 60-73]. During HiL tests the developed functions run on the target hardware of the research vehicle. The target hardware is an ETAS ES910 rapid prototyping ECU and can execute compiled Simulink models. The HiL test bench allows to test the interaction of the software on ECU with the simulated vehicle architecture close to reality before the actual test in the vehicle.

The HiL test bench consists of four hardware components. A real-time computer, a host computer, the prototyping ECU and a car PC. Figure 6.2 shows the connections between the components of the HiL test bench in a simplified manner. The real-time computer is an IPG Xpack4 that runs a simulation environment including all CAN communication between the ES910 and the architecture of the virtual vehicle. The host PC is an ordinary desktop computer that controls the real-time computer via an ethernet connection, starts simulations, records simulation data and provides a graphic simulation output. The simulation software is IPG CarMaker HiL, a vehicle dynamics simulation software that is extended by the capability to control the Xpack4.

In CarMaker a vehicle model was created that is more detailed than the vehicle reference models described in Section 4.5 and also takes into account tire characteristics and vehicle kinematics. Furthermore, a model of the power train as described in Section 4.3 was integrated into the CarMaker vehicle model as a Simulink model and a road model of the WR was created from Open Street Map data and included in CarMaker. These models were also added by B. Fath.

The host PC also runs the experiment environment ETAS INCA. Via a second ethernet connection of the host PC, INCA can flash compiled Simulink models onto the ES910, start them on the ES910 and view and change model parameters during run-time, e.g. for calibration purposes.

The car PC is a small computer designed for automotive applications and connected to the ES910 via a CAN bus. It receives a GPS position that is generated by the simulation on the Xpack4 and passed trough the ES910. Using this information the map software EB Assist ADTF by Electrobit, which runs on the car PC, provides map data for the ES910.

A similar HiL setup also for a trajectory optimization and vehicle control use case is described in [54, pp. 181-192].

The HiL environment enables safe, reproducible and automatically performed tests. It also helps to reduce expensive testing time in the real vehicle because most specifications can be verified in advance [187, pp. 163-168].

Despite the great benefits of the in-the-Loop methods, they cannot completely replace real test drives. At further development stages new test scenarios that cause problems more likely occur in reality than in the already well-known and simplified simulation environment. Furthermore, some features of driver assistance systems require a subjective assessment [187, pp. 163-168].

□ Real test drives were conducted on the WR itself as well as on the highway and at roundabouts nearby. The test drives could focus on determining various calibration parameters such as acceleration look-up tables, weighting factors of the trajectory optimization and parameters of the controller that computes the torque demand. Parameter settings for three different modes that correspond to different driving styles were derived. In a final acceptance test about ten employees of the project partner who work in automotive research and engineering tested the ALC on self chosen routes. *Adopted from [79].*

6.2 System architecture and design

□ The software components of the ALC are denoted modules. The ALC system consists of route data module, parameter adaption module, trajectory module and controller module. Figure 6.3 depicts the architecture of ALC. The following sections describe the component design, i.e. the function each module performs. *Adopted from [79].*

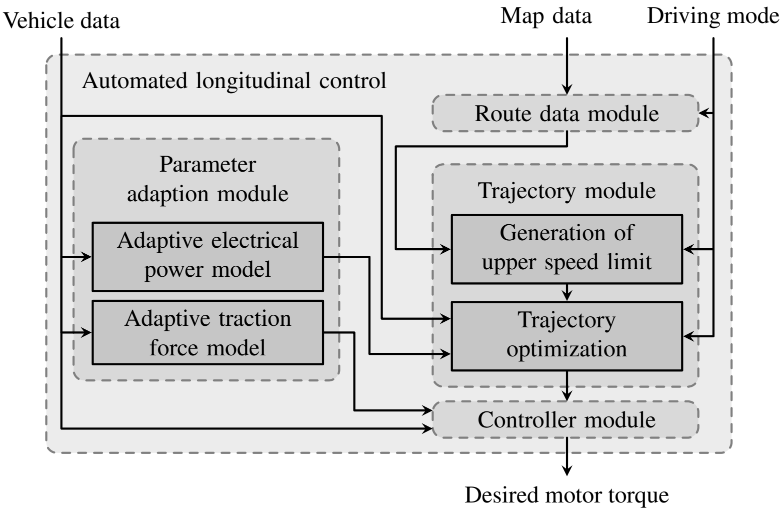

Vehicle data
Map data
Driving mode
Automated longitudinal control
Route data module
Parameter adaption module
Adaptive electrical power model
Adaptive traction force model
Trajectory module
Generation of upper speed limit
Trajectory optimization
Controller module
Desired motor torque

Figure 6.3: System architecture of the automated longitudinal control. *Adopted from [79].*

6.2.1 Route data module

□ The input to the route data module comprises six vectors that contain the values and positions of legal speed limits, the road curvature and the slope for the road section ahead of the vehicle in driving direction. Similar to a run-length encoding, the vectors contain only information for positions at which the corresponding quantity changes significantly. The map data is provided via the CAN bus by the navigation system. *Adopted from [79].* Since no destination is specified, the navigation system only knows the route up to the next junction or roundabout. If all roads connected to the junction have the same average amount of traffic, the electronic horizon ends at this junction. If one road has a higher traffic density, the navigation system assumes that the driver takes this road because it is the most probable path (MPP).

Following the MPP repeatedly leads onto the next larger road and finally onto a highway. Unless the driver follows an assumed MPP, there will be a short time period with no available map data. The navigation system then recalculates a new path whereby the length of the electronic horizon quickly increases.

Sometimes the driver likes to turn at a junction but the navigation system assumes that the junction will be crossed straight. In such cases the ALC does not reduce the vehicle velocity sufficiently and the driver needs to brake. Since the ALC neither uses information about right of way nor can detect traffic signs or traffic lights via a camera, the driver usually has to override the ALC at junctions and roundabouts.

The length of the electronic horizon ranges from several hundred meters in urban areas over roughly 1 km on country roads to several kilometers on highways.

The route data module is executed every 500 ms and mainly converts the map data to a meter discrete representation for up to 3 km ahead of the vehicle position and provides three vectors that describe the courses of road slope, road curvature and legal speed limit. However, the input vectors of the legal speed limit can be modified before the conversion by three mechanisms:

First, the driver can overwrite the current legal speed limit of the map data by adding a positive or negative offset using buttons next to the steering wheel. This adaption functionality proves beneficial in situations in which the legal speed limit in the map data does not match the actual legal speed limit, for example at temporary construction sites.

Second, the positions of upcoming speed limits can be changed. If the legal speed limit is increased, the ALC will not start to accelerate the vehicle before the new traffic sign is passed because the planned trajectory does not exceed the generated upper speed limit. When leaving a city, this behavior is usually undesired by the driver and can provoke overtaking maneuvers of following vehicles. Therefore two parameters were added. Both describe a time offset, one for the case of an increase of the legal speed limit and one for a decrease of the legal speed limit. In both cases the corresponding parameter is multiplied with the lower of the two speed limits to determine the spatial distance by which the position of the higher speed limit is shifted in the map data. This allows to implement a more realistic driving behavior or adaption to driver preferences. For example, it can be achieved that the ALC already starts to accelerate when the driver sees the traffic sign that indicates a higher legal speed limit.

Third, an upper limit is enforced on the legal speed limit depending on the driving mode of the ALC selected by the driver. This is needed on highways because there is not always a legal speed limit.

The route data module was implemented by D. Dörr [22, pp. 23, 24]. The author of this work added the time offset parameters and logic for processing roundabout data.

6.2.2 Parameter adaption module

□ The parameter adaption module updates the ATFM and AEPM described in Subsection 4.7.1 and Subsection 4.7.2, respectively, every 50 ms using vehicle data from the CAN bus. During the ATFM update one KF iteration is performed and during the AEPM update the FB-KRLS takes into account a new measurement.

An update of the ATFM requires the traction force F_{trac}, the vehicle velocity v_{Vhcl} and the sensor longitudinal acceleration a_{x} and an update of the AEPM the electrical traction power $P_{\text{trac,elec}}$, v_{Vhcl} and a_{x}. While v_{Vhcl} and a_{x} are measurement signals on the CAN bus, F_{trac} needs to be approximately computed from the motor torque signals using (4.25). $P_{\text{trac,elec}}$ can be determined from the voltages and currents of the electric motors that are available on the CAN bus.

Before CAN signals are fed to an estimator or used to calculate the input quantity for an estimator, signal noise is removed using a polynomial function approximation method [145], also denoted polynomial Kalman smoother [144].

In two cases the models are not adapted: First, during a vehicle standstill because the missing excitation in the data can cause estimations to diverge from their true value. Second, if CAN signals indicate that there is significant hydraulic brake pressure because the models only consider braking via recuperation and are not valid when hydraulic brakes are active.

Output of the parameter adaption module are updated parameters of ATFM and AEPM. The parameter adaption module including ATFM is from F. Bleimund [22, pp. 24-29]. The author of this work added the AEPM. *Adopted from [79].*

6.2.3 Trajectory module

□ The trajectory module performs the tasks that Chapter 5 described in detail. These tasks include generating an upper speed limit and planning a trajectory. Inputs are the map data from the route data module, parameters of the AEPM provided by the parameter adaption module, vehicle data and the selected driving mode. The selected driving mode determines which longitudinal acceleration look-up table is used to compute the speed limit from map data with desired acceleration. Furthermore, in driving mode "Normal" and driving mode "Sport" RBA performs the trajectory optimization whereas in driving mode "Range" NRBA optimizes the trajectory using the AEPM, whereby the absolute value of the electrical traction power is penalized strongly.

The trajectory module is called every 500 ms and then can either continue planning the current trajectory or start planning a new trajectory.

Continuing planning the current trajectory means that the module performs additional iterations of RBA or NRBA until it reaches the iteration limit for the current module call or the optimized trajectory reaches the end of the electronic horizon (c.f. Figure 5.3).

Planning of a new trajectory is only triggered if distance limit or time limit since the last beginning of a trajectory planning have been exceeded, if the driver has stopped overriding the ALC by pressing the braking pedal or accelerator pedal or if data of the curvature vector has changed more than a threshold since the last module call indicating that the driver has chosen a route that differs from the MPP. Additionally, the length of the electronic horizon must exceed a specified lower limit to enable planning of a new trajectory. *Adopted from [79].*

In two cases, v_{TJY} and a_{TJY} of the new trajectory are projected onto the vehicle velocity v_{Vhcl} and vehicle longitudinal acceleration $\dot{v}_{\mathrm{Vhcl,x}}$ according to Subsection 5.5.1: First, if there is no valid previous trajectory, e.g. because map data was temporarily not available, and second, if the current vehicle velocity deviates more than a certain threshold from the planned velocity v_{TJY}, e.g. because the vehicle is starting from a standstill or because of an intervention by the driver.

In all other cases $v_{\mathrm{TJY}}(t_1)$, $a_{\mathrm{TJY}}(t_1)$ and $j_{\mathrm{TJY}}(t_1)$ with $t_1 = 0$ of the new trajectory are projected onto $v_{\mathrm{TJY}}(t^*)$, $a_{\mathrm{TJY}}(t^*)$ and $j_{\mathrm{TJY}}(t^*)$ of the current trajectory at its last evaluation point t^* according to Subsection 5.5.2 in order to achieve a smooth connection between both trajectories at t^*.

The knot vector $\boldsymbol{\kappa}$ and estimated control point vector $\hat{\boldsymbol{x}}$ are the outputs of the trajectory module.

6.2.4 Controller module

□ Every 20 ms the controller modules computes the desired velocity v_{des} and desired acceleration a_{des} and translates these quantities into a motor torque demand T_{des} that causes the vehicle to track the velocity trajectory. In most cases v_{des} and a_{des} equal the trajectory velocity v_{TJY} and the trajectory acceleration a_{TJY}, respectively. v_{TJY} and a_{TJY} are determined by evaluating the trajectory defined by the knot vector and estimated control point vector from the trajectory module at the current point in time measured starting from the last trajectory planning start. *Adopted from [79].*

However, the controller also comprises three functionalities of the trajectory module:

1. Trajectory adaption to vehicle motion state: While the driver is overriding the ALC by using the pedals, the controller module projects the trajectory onto the vehicle motion state given by vehicle velocity and vehicle acceleration as stated in Subsection 5.5.1. Therefore, as soon as the ALC takes over control again, the controller can immediately evaluate the trajectory that is connected jerk-free to the vehicle motion state. During the next call of the trajectory module planning of a new trajectory is initiated.

2. Trajectory adaption to the vehicle ahead (c.f. Subsection 5.5.3): The IDM in the trajectory module serves mainly for planning the approach to the vehicle ahead. For the actual vehicle following task an IDM is implemented in the controller as well. In order to be able to modify v_{TJY} and a_{TJY} according to IDM, the controller module can perform the state projection method described in Subsection 5.5.1.

3. Enforcement of velocity and acceleration constraints: If the controller changes the control points of the trajectory, it must also enforce the restrictions $v_{\mathrm{TJY}} \geq 0$ by adapting control point values and enforce restrictions with respect to a_{TJY} by adapting the differences of neighboring control point values (c.f. Section 5.5).

Since the controller module can alter the trajectory, the trajectory module accepts a control point vector coming from the controller module as input and interprets it as the previous trajectory. If there is no valid trajectory available for evaluation, the controller module causes the desired vehicle

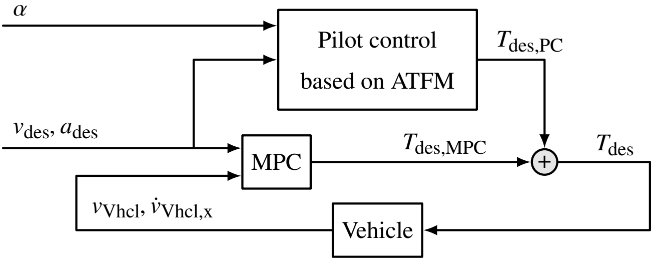


Figure 6.4: Architecture of controller module and control loop. The pilot control generates an open-loop motor torque demand $T_{\text{des,PC}}$ based on desired velocity v_{des}, desired acceleration a_{des} and road slope angle α using the adaptive traction force model (ATFM). The model predictive control (MPC) computes a closed-loop torque demand $T_{\text{des,MPC}}$ to minimize the remaining deviation of vehicle velocity v_{Vhcl} and vehicle longitudinal acceleration $\dot{v}_{\text{Vhcl,x}}$ from v_{des} and a_{des}. *Adopted from [79].*

acceleration to quickly diminish and keeps the velocity constant. If no map data is available for several seconds, the ALC system deactivates itself.

The in this subsection above stated functionality was added by the author of this work. However, the architecture and control loop including the pilot control were adopted from F. Bleimund [22, pp. 35, 36].

□ Figure 6.4 depicts the controller module architecture and the control loop. v_{des} and a_{des} are inputs to a pilot control and a model predictive control (MPC). The pilot control contains the ATFM and computes an open-loop torque demand $T_{\text{des,PC}}$ based on road slope angle α, v_{des} and a_{des}. Due to imperfect map data, sensor data and ATFM, the vehicle velocity v_{Vhcl} deviates from v_{des} and the vehicle longitudinal acceleration $\dot{v}_{\text{Vhcl,x}}$ deviates from a_{des}. The two-dimensional MPC computes a closed-loop torque demand $T_{\text{des,MPC}}$ in order to minimize these deviations.

The torque demand of the controller module T_{des} is the sum of the pilot control torque demand $T_{\text{des,PC}}$ and the MPC torque demand $T_{\text{des,MPC}}$ and reaches the motor ECU via the CAN bus. The motor ECU computes the distribution of the torque demand between the front and rear motor using the strategy described in Section 4.3 and controls these actuators accordingly. If the torque demands indicates a deceleration request, it can additionally be allocated to the brake ECU that controls the hydraulic brakes.

The control loop is closed through the vehicle where CAN messages transfer the feedback information. A comprehensive review of trajectory tracking methods is provided in [40].

Pilot control

The pilot control determines an open-loop torque demand $T_{\mathrm{des,PC}}$ with

$$T_{\mathrm{des,PC}} = \frac{F_{\mathrm{trac}}}{i_{\mathrm{G}}} \cdot \frac{r_{\mathrm{dyn,FA}} + r_{\mathrm{dyn,RA}}}{2} \tag{6.1}$$

so that the vehicle roughly tracks v_{des} and a_{des}. i_{G} is the gear ratio, $r_{\mathrm{dyn,FA}}$ the dynamic front wheel radius and $r_{\mathrm{dyn,RA}}$ the dynamic rear wheel radius. The traction force F_{trac} results from the ATFM in Subsection 4.7.1:

$$F_{\mathrm{trac}} = \underbrace{(1, a_{\mathrm{x}}, v_{\mathrm{des}}^2)}_{=:\boldsymbol{C}_{\mathrm{Vhcl}}} \cdot \boldsymbol{x}_{\mathrm{Vhcl}} \tag{6.2}$$

The parameter adaption module provides the adapted vehicle parameter vector $\boldsymbol{x}_{\mathrm{Vhcl}}$. The vehicle motion vector $\boldsymbol{C}_{\mathrm{Vhcl}}$ depends on ${v_{\mathrm{des}}}^2$ and a_{x}, whereby

$$a_{\mathrm{x}} = a_{\mathrm{des}} + g \cdot \sin(\alpha). \tag{6.3}$$

Adopted from [79]. According to Subsection 4.2.1, the road slope angle α can be computed from the road slope γ, which can be derived from map data or the signal of a slope estimator that is available on the CAN bus.

The simple calculations in the pilot control are very transparent, the estimated vehicle parameters can easily be limited to a valid range and the other values are measurements. The pilot control contributes to the torque demand to a large extent and serves for overcoming the majority of the sum of driving resistances. Using the pilot control allows to restrict the output of the MPC that closes the control loop to low absolute values without risiking a vehicle standstill at a slope. Restricting the MPC output seems beneficial from safety considerations because the MPC performs operations that can be numerically problematic such as dividing by potentially small values. Additionally it allows to apply a simpler vehicle model in the MPC for less computational effort.

Model predictive control

Model predictive control (MPC) is a direct trajectory optimization method that approximates the constrained infinite dimensional OCP from (2.1) by a finite dimensional problem that can be solved efficiently in real-time. The approximation occurs by discretizing the problem and solving it for a finite time horizon T.

MPC uses a system model to predict the future system behavior. A possible linear time invariant system model without uncertainties reads

$$\begin{aligned} \boldsymbol{x}_{p+1} &= \boldsymbol{\mathcal{A}}\boldsymbol{x}_p + \boldsymbol{\mathcal{B}}\boldsymbol{u}_p \\ \boldsymbol{y}_p &= \boldsymbol{C}\boldsymbol{x}_p \end{aligned} \tag{6.4}$$

with system state $\boldsymbol{x}_p$, control input $\boldsymbol{u}_p$, system output $\boldsymbol{y}_p$ and discrete time index p. In each time step t, MPC solves the following optimization problem for the horizon $t = 0, \delta t, \ldots, T$ with cycle time δt:

$$\begin{aligned} \hat{\boldsymbol{u}}_t &= \arg\min_{\boldsymbol{u}_t} \mathcal{J}(\boldsymbol{x}_t, \boldsymbol{u}_t) \\ \mathcal{J}(\boldsymbol{x}_t, \boldsymbol{u}_t) &= \sum_{i=0}^{T-1} \left[(\boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{x}_{i|t})^\top \boldsymbol{Q} \, (\boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{x}_{i|t}) + \boldsymbol{u}_{i|t}^\top \boldsymbol{\mathcal{R}} \boldsymbol{u}_{i|t}^\top \right] \\ \text{subject to } & \boldsymbol{\mathcal{F}}\boldsymbol{x}_t + \boldsymbol{\mathcal{G}}\boldsymbol{u}_t \leq \mathbf{1} \end{aligned} \tag{6.5}$$

Herein the control input sequence $\boldsymbol{u}_t = \{\boldsymbol{u}_{0|t}, \boldsymbol{u}_{1|t}, \ldots, \boldsymbol{u}_{T-1|t}\}$ over the horizon is the optimization variable. $\boldsymbol{x}_{i|t}$ and $\boldsymbol{u}_{i|t}$ denote the predicted values of $\boldsymbol{x}$ and $\boldsymbol{u}$, respectively, for time $t+i$ based on the information available at t, whereby $\boldsymbol{x}_{0|t} = \boldsymbol{x}_t$ is assumed. Present and future set points are specified via the system output $\boldsymbol{y}_t = \{\boldsymbol{y}_{0|t}, \boldsymbol{y}_{1|t}, \ldots, \boldsymbol{y}_{T-1|t}\}$.

The cost function $\mathcal{J}$ includes symmetric error weighting matrices $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{Q}$, which are assumed positive definite and positive semidefinite, respectively.

The matrices $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{G}}$ allow for linear constraints with respect to $\boldsymbol{x}$ and $\boldsymbol{u}$, respectively. The constraints are formulated as inequalities, which apply elementwise. $\mathbf{1}$ is a matrix of ones. For increased robustness and in order to avoid an empty set of feasible solutions, constraint softening can be applied. Thereby hard constraints become penality terms in the cost function analogously to the approach stated in Section 5.5 [61].

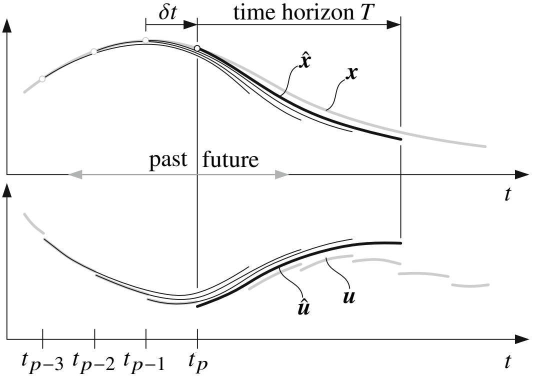


Figure 6.5: Model predictive control optimization horizon with time horizon T, cycle time δt, current time step t_p, planned and actual state $\hat{\boldsymbol{x}}$, $\boldsymbol{x}$, planned and actual control input $\hat{\boldsymbol{u}}$, $\boldsymbol{u}$ [187, p. 1431, adapted].

Of the optimal control sequence $\hat{\boldsymbol{u}}_t$ only the first element $\hat{\boldsymbol{u}}_{0|t}$ is applied to the system. At the following time step the process is repeated with the additional information $\boldsymbol{x}_{t+1}$ on the system state.

Figure 6.5 illustrates this MPC approach, that creates a feedback which partially compensates for model inaccuracies and leads to a closed-loop control that can take anticipatory control actions for future set points and events. The abilities of looking ahead and considering constraints are also distinguishing features of MPC compared to a PID controller [187, pp. 1431-1432], [95, pp. 2, 13-16].

An analysis of different controller designs is provided by [169] and different vehicle models for controller design are stated in [136, p. 12].

Vehicle related MPC applications in literature include ACC systems, e.g. [13, 61, 112, 154]. These use MPC as a high level controller that calculates a sequence of acceleration commands for the ACC equipped vehicle so that it keeps the required distance to a vehicle ahead, whereas a low level controller tracks the acceleration sequence using commands to power train and brakes. In the MPC the vehicle is represented with a first-order model because the low level controller cannot realize commands instantaneously due to the limited bandwidth of the vehicle.

In this work the MPC serves only as a low level controller and is additionally supported by the pilot control that considers the driving resistances.

Furthermore it seems reasonable to assume that the research vehicle is capable of following power train and brake commands relatively quickly because of the high power to weight ratio and the missing delay from changing gears. For these reasons a vehicle model similar to the one in [116] is used. It is not a first-order model but a double integrator involving vehicle velocity v_{Vhcl}, vehicle longitudinal acceleration $\dot{v}_{\mathrm{Vhcl,x}}$ and vehicle longitudinal jerk $\ddot{v}_{\mathrm{Vhcl,x}}$. The forward difference approximations

$$\begin{aligned}\dot{v}_{\mathrm{Vhcl,x}}(t) &= \frac{v_{\mathrm{Vhcl}}(t+\delta t) - v_{\mathrm{Vhcl}}(t)}{\delta t}\\ \ddot{v}_{\mathrm{Vhcl,x}}(t) &= \frac{\dot{v}_{\mathrm{Vhcl,x}}(t+\delta t) - \dot{v}_{\mathrm{Vhcl,x}}(t)}{\delta t}\end{aligned} \tag{6.6}$$

give the time-discrete state-space representation of the vehicle model, in which $\ddot{v}_{\mathrm{Vhcl,x}}$ is the control input [13, 116]:

$$\begin{pmatrix} v_{\mathrm{Vhcl},t+1} \\ \dot{v}_{\mathrm{Vhcl,x},t+1} \end{pmatrix} = \begin{pmatrix} 1 & \delta t \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} v_{\mathrm{Vhcl},t} \\ \dot{v}_{\mathrm{Vhcl,x},t} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \ddot{v}_{\mathrm{Vhcl,x},t} \tag{6.7}$$

The system state comprises vehicle velocity and acceleration and should be equal to the set points for the desired velocity v_{des} and desired acceleration a_{des}. Hence, the quantities in the state-space model of (6.4) read:

$$\begin{aligned}&\mathcal{A} = \begin{pmatrix} 1 & \delta t \\ 0 & 1 \end{pmatrix}, \quad \mathcal{B} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \boldsymbol{C} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\\ &\boldsymbol{u}_t = \ddot{v}_{\mathrm{Vhcl,x}}, \quad \boldsymbol{x}_t = \begin{pmatrix} v_{\mathrm{Vhcl}} \\ \dot{v}_{\mathrm{Vhcl,x}} \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} v_{\mathrm{des}} \\ a_{\mathrm{des}} \end{pmatrix}\end{aligned} \tag{6.8}$$

Simplifying F_{trac} in (6.1) to $F_{\mathrm{trac}} = m_{\mathrm{vhcl}} \cdot \dot{v}_{\mathrm{Vhcl,x}}$ and calculating the temporal derivative shows that the control input $\boldsymbol{u}$ defined as the vehicle longitudinal jerk $\ddot{v}_{\mathrm{Vhcl,x}}$ translates to changes of MPC torque demand, $T_{\mathrm{des,MPC}}$:

$$\frac{\mathrm{d}T_{\mathrm{des,MPC}}}{\mathrm{d}t} = \frac{m_{\mathrm{vhcl}} \cdot \ddot{v}_{\mathrm{Vhcl,x}}}{i_{\mathrm{G}}} \cdot \frac{r_{\mathrm{dyn,FA}} + r_{\mathrm{dyn,RA}}}{2} \tag{6.9}$$

Discretization of the above equation using the backward difference approximation gives $T_{\text{des,MPC}}$ for the current time step:

$$T_{\text{des,MPC},t} = T_{\text{des,MPC},t-1} + \dddot{v}_{\text{Vhcl,x},t} \cdot \frac{m_{\text{vhcl}} \cdot \delta t}{i_{\text{G}}} \cdot \frac{r_{\text{dyn,FA}} + r_{\text{dyn,RA}}}{2} \tag{6.10}$$

Thereby $T_{\text{des,MPC,0}} = 0$ and $\delta t = 0.2$ s are used. The change of $T_{\text{des,MPC}}$ in each δt is limited to $[\Delta T_{\text{min}}, \Delta T_{\text{max}}] = [-50 \text{ Nm}, +50 \text{ Nm}]$ with following matrices in (6.5):

$$\mathcal{F} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \mathcal{G} = \begin{pmatrix} -\frac{m_{\text{vhcl}} \cdot \delta t \cdot (r_{\text{dyn,FA}} + r_{\text{dyn,RA}})}{2 \cdot i_{\text{G}} \cdot \Delta T_{\text{min}}} \\ \frac{m_{\text{vhcl}} \cdot \delta t \cdot (r_{\text{dyn,FA}} + r_{\text{dyn,RA}})}{2 \cdot i_{\text{G}} \cdot \Delta T_{\text{max}}} \end{pmatrix} \tag{6.11}$$

Furthermore, the $\mathcal{R}$ describing the cost of the jerk and $\boldsymbol{Q}$ describing the cost of deviating from the trajectory velocity and the trajectory acceleration on its main diagonal elements are set to

$$\mathcal{R} = 10^{-4}, \quad \boldsymbol{Q} = \begin{pmatrix} 20 & 1 \end{pmatrix} \cdot \boldsymbol{I}. \tag{6.12}$$

The horizon comprises the current time step t and nine future time steps, which results in 1.8 s look ahead of the MPC. With a previously in *e-generation* applied PID controller uncomfortable oscillations between acceleration and deceleration occured on test drives while driving downhill with roughly constant velocity. After the PID controller had been replaced with the MPC, these oscillations were not noticed anymore.

However, the computational effort of MPC is usually much larger than that of the PID controller and depends on the system model and the prediction horizon [147]. Problem (6.5) has a convex quadratic objective function and linear constraints, which allows using various QP solvers based on active set methods or interior point methods for performing the online MPC optimization.

This work uses the general purpose QP solver provided by [94] with minor modifications that consist of rewriting instructions that are incompatible with compilation for the target hardware and of removing functionality and dependencies that are not needed to solve the above stated problem. Modifications of the solver, first implementations and tests of promising state-space models within a provided simplified control loop were done by C. Lee [98] during his time as student assistant that started after completion of his Master's thesis. Parameter tuning and closed-loop testing within the ALC in simulated and real test drives were done by the author of this work.

The disadvantage of general purpose QP solvers is that they do not exploit the special MPC problem structure and therefore using them might prevent the application of MPC to problems with a high sample rate, high-dimensional model, or long time horizon T. For example, the computational effort of both active set and interior point methods increases roughly cubically with T, whereas the effort of algorithms that exploit sparse matrices increases only linearly [95, p. 42], [147]. A survey of technologies for linear and nonlinear MPC is given by [137] and [171] provides a comparison between PID controller, linear MPC and nonlinear MPC.

6.3 Energy-saving potential and effects of parameters

□ This section investigates the energy-saving potential using the automated longitudinal control (ALC) with different parameter settings compared to drives with manual longitudinal control (MLC) on the Weissach route (WR) defined in Section 4.6 in simulations.

Automated longitudinal control setup and data generation

The varied parameters are the temporal distance of neighboring knots Δt_κ, the number of spline intervals I, the temporal safety margin to upper speed limit $\Delta t_{\mathrm{Lim,TJY}}$ and the weight of power error square R_P^{-1}. Simulations of the ALC are conducted for all possible parameter combinations that result from $\Delta t_\kappa = \{1.5\text{ s}, 3\text{ s}\}$, $I = \{1, 3\}$, $\Delta t_{\mathrm{Lim,TJY}} = \{1\text{ s}, 2\text{ s}\}$ and $R_P^{-1} = \{\frac{1}{10000}, \frac{1}{5000}, \frac{1}{1000}, \frac{1}{500}, \frac{1}{100}, \frac{1}{50}\}$. *Adopted from [79].*

During a simulation of the ALC on the WR, the position s on the WR is calculated by integrating the vehicle velocity over time. With s the corresponding road curvature and road grade are determined from the map data of the WR.

Section 5.4 applies the open-loop reference model from Section 4.5 in a backward simulation in order to calculate the energy consumption that results when the vehicle tracks a planned velocity trajectory perfectly. The current section considers the whole ALC including its controller module and calculates the energy consumption that results from the actually realized velocity trajectory using the closed-loop reference model from Section 4.5 in a forward simulation.

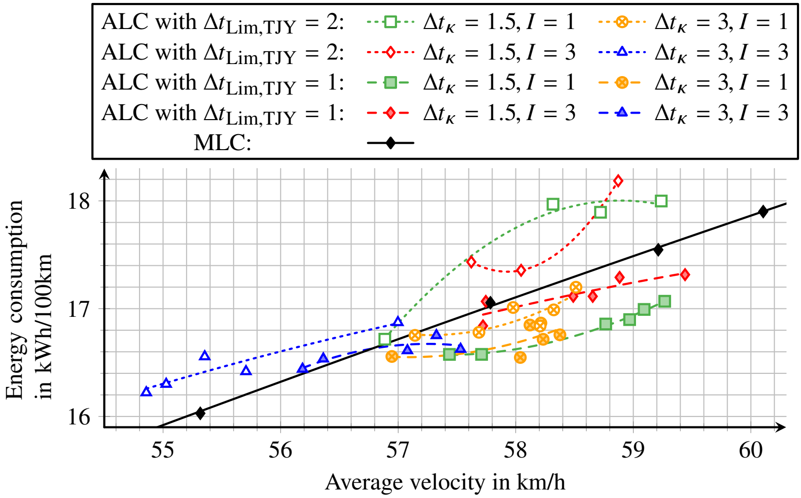


Figure 6.6: Average energy consumption vs. average velocity on the Weissach route with automated longitudinal control (ALC) under different parameter settings in comparison to adapted and resimulated real drive with manual longitudinal control (MLC). *Adopted from [79].*

□ In order to compare different ALC settings, the required trip time t_{Trip} for completing the WR is converted into the average velocity by dividing t_{Trip} by the length of the WR and the energy consumption is scaled to the energy consumption per 100 km. This approach allows to summarize each simulation in a single data point in Figure 6.6, which depicts energy consumption versus average velocity. The lines are approximations of data points differing only in R_P^{-1} by quadratic polynomials. *Adopted from [79].*

For some parameter combinations the values of R_P^{-1} are high or low enough to make the MPF instable. In such cases the resulting energy consumption is very high while the average velocity is either very low or very high. Such data points are excluded from Figure 6.6. Reduced convergence radii are noticed mainly for the combinations that include $\Delta t_{\mathrm{Lim,TJY}} = 2$ s and $\Delta t_\kappa = 1.5$ s. $\Delta t_{\mathrm{Lim,TJY}}$ influences the data provided to the MPF and for nonlinear filters the covariance also depends on the data. If Δt_κ is reduced while Δt_{It} is kept constant, the covariance matrix entries tend to remain larger. This suggests that for these settings, the MPF parameter $\boldsymbol{Q}^L, \boldsymbol{Q}^N$ need to be reduced.

Manual longitudinal control setup and data generation

□ For comparison with a drive with MLC, a real test drive on the WR with MLC was performed and CAN data was recorded. From the recorded data, only the vehicle velocity v_{Vhcl} with a sample rate of 100 Hz was further used. By integrating v_{Vhcl} over time, the corresponding position on the WR for each data point is computed. Time information is then discarded because the following modifications of v_{Vhcl} do not depend on time:

First v_{Vhcl} is multiplied with a factor around one in order to achieve MLC drives with various average velocities in the same range as the average velocities of the ALC simulations.

During the real test drive the driver had to stop the vehicle at junctions in order to check for other traffic participants with right of way. When the vehicle has to come to a standstill and drive off again this increases both the energy consumption and time required for completing the WR.

The simulation environment neglects right of way and other traffic participants do not occur. Therefore the vehicle passes junctions with a low speed without coming to a standstill. In order to not disadvantage MLC drives in the comparison, position dependent lower limits on v_{Vhcl} are applied to MLC drives. These lower limits coincide with the velocities the ALC chooses at the corresponding junction. Varying ALC parameter settings influences the velocity with ALC at the occuring junctions only slightly.

During the real test drive there was few traffic and only once during a short period the driver was restricted in the choice of vehicle velocity because of a vehicle ahead. At the corresponding position the recorded v_{Vhcl} data was adapted to a typical ALC behavior.

The ALC computes an upper speed limit for the route ahead, which among others depends on the specified maximum absolute value of lateral acceleration $\dot{v}_{\text{Vhcl,y,max}}$. The originally recorded v_{Vhcl} results in a lateral acceleration larger than the maximum absolute value of lateral acceleration for the ALC. Increasing $\dot{v}_{\text{Vhcl,y,max}}$ leads to higher average velocity and less energy consumption. In order to compare ALC and real drive under same conditions, an upper limit is enforced on the v_{Vhcl} of the MLC drives. The upper limit is the speed limit from map data with desired acceleration.

After multiplying the recorded v_{Vhcl} with a factor around one and imposing the various limits the corresponding time t is re-calculated using $t = \frac{s}{v_{\mathrm{Vhcl}}}$ and the MLC energy consumption is determined with the open-loop reference model. Figure 6.6 depicts the resulting data points and quadratic approximation for the MLC in black and by a solid line.

The quadratic approximation line serves as a benchmark. The ALC data points should be below this line, meaning that the ALC achieves the same average velocity with less energy consumption or a higher average velocity for the same energy consumption.

Initial evaluations of the energy consumption of MLC and ALC in a non-final setup were performed by A. Thorgeirsson [173] as a student assistant.

Parameter effects

Temporal distance of neighboring knots

Approximations for data points that differ only in the temporal distance of neighboring knots Δt_κ and weight of power error square R_P^{-1} indicate higher average velocities and mostly also higher energy consumptions for $\Delta t_\kappa = 1.5$ s than for $\Delta t_\kappa = 3$ s. The reason for this is that trajectories with larger Δt_κ cannot follow the upper speed limit as closely because their degrees of freedom are temporally less dense. Hence, the vehicle is generally slower and omits some inefficient velocity peaks. *Adopted from [79].*

Number of spline intervals

When the number of spline intervals I is increased, the MPF can adjust the trajectory in more previous trajectory intervals. When R_P^{-1} is chosen high for strong penalization of the electrical traction power, a large I should be beneficial because the MPF can for instance retrospectively reduce the acceleration if the road grade suddenly increases and maintaining the acceleration would lead to large power demand and therefore large power loss.

However, similar to Chapter 3 and Chapter 5 the simulation results do not support this statement. All approximations for $\Delta t_\kappa = 1.5$ s and $I = 1$ are below those for $\Delta t_\kappa = 1.5$ s and $I = 3$. For $\Delta t_\kappa = 3$ s the ranges of the average velocities of the approximations for $I = 1$ and $I = 3$ do not always overlap but they can still be compared using their positions with respect to the MLC approximation. Using this criterion the approximations for $I = 1$ are always better.

The reason for the worse results with larger I can be explained by the fact that the PF in the MPF uses a set of particles, each of which is a possible trajectory control point vector. The number of particles equals 100 for all simulations. For larger I, each particle comprises more dimensions and therefore each particle dimension is sampled less densely.

In order to keep the sampling density in each dimension constant, the number of particles needs to be increased exponentially with I. Since the computational effort increases linearly with the number of particles, this quickly becomes infeasible especially with computation time restrictions. From these considerations and the simulation results, it seems beneficial to only use $I = 1$ in combination with MPF and power penalization. Section 5.3 investigated the effect of increasing I in combination with the KF.

Furthermore, with $I = 3$ there is less potential for an effect from varying weight R_P^{-1}. Moreover, the approximation for $\Delta t_{\mathrm{Lim,TJY}} = 2$ s, $\Delta t_\kappa = 1.5$ s and $I = 1$ comprises a larger range of average velocities and energy consumptions than the corresponding approximations for $\Delta t_\kappa = 3$ s.

Temporal safety margin to upper speed limit

When the temporal safety margin to upper speed limit $\Delta t_{\mathrm{Lim,TJY}}$ is decreased, the trajectory follows the speed limit from map data with desired acceleration $v_{\mathrm{Lim,Map},\dot{v}}$ more closely which increases the average verlocity. Furthermore, the trajectory can follow short lasting inefficient velocity peaks which increases energy consumption. Both these effects can be observed for the setting $\Delta t_\kappa = 3$ s and $I = 3$, which achieve comparatively low average velocities. For the other settings, however, the average velocity remains almost unchanged but the energy consumption decreases. Lowering $\Delta t_{\mathrm{Lim,TJY}}$ also increases the trajectory velocity at local minima of $v_{\mathrm{Lim,Map},\dot{v}}$ and can cause the trajectory to exceed $v_{\mathrm{Lim,Map},\dot{v}}$ at its local minima. Such a behavior decreases energy consumption and increases average velocity at the same time at the expense of driving safety. The increase in average velocity is not observed though.

$v_{\mathrm{Lim,Map},\dot{v}}$ is an upper speed limit calculated using parameters that are determined in real test drives. Slightly exceeding $v_{\mathrm{Lim,Map},\dot{v}}$ is in most cases not noticed but should be avoided because the extent to which the trajectory exceeds $v_{\mathrm{Lim,Map},\dot{v}}$ is not clearly defined. Figures in Chapter 5 demonstrate that trajectories with $\Delta t_{\mathrm{Lim,TJY}} = 1$ s stay below $v_{\mathrm{Lim,Map},\dot{v}}$. Figure 5.4 indi-

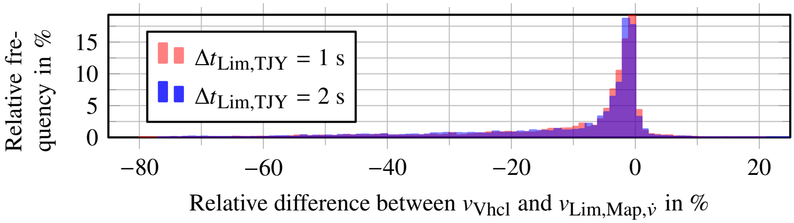


Figure 6.7: Histograms of the relative frequencies of relative differences between vehicle velocity v_{Vhcl} and speed limit from map data with desired acceleration $v_{\mathrm{Lim,Map},\dot{v}}$ for different temporal safety margin to upper speed limit $\Delta t_{\mathrm{Lim,TJY}}$ with $R_P^{-1} = \frac{1}{5000}$, $\Delta t_{\kappa} = 1.5$ s and $I = 1$.

cates that even with the combination $\Delta t_{\mathrm{Lim,TJY}} = 1$ s, $I = 5$ and $\Delta t_{\kappa} = 5$ s the trajectory remains below $v_{\mathrm{Lim,Map},\dot{v}}$.

For a more complete assessment of the extent and relative frequency of $v_{\mathrm{Lim,Map},\dot{v}}$ exceedances, Figure 6.7 shows histograms of the relative difference between the actual vehicle velocity v_{Vhcl} and $v_{\mathrm{Lim,Map},\dot{v}}$ for the $\Delta t_{\mathrm{Lim,TJY}} = 1$ s and $\Delta t_{\mathrm{Lim,TJY}} = 2$ s with the other parameters $\Delta t_{\kappa} = 1.5$ s, $I = 1$ and $R_P^{-1} = \frac{1}{5000}$. The histograms reveal that there are exceedances which was to be expected because of the unconstrained trajectory optimization problem formulation. The majority of these remain below 2 %, which also includes the controller inaccuracies. Higher exceedances can be observed but their relative frequencies are too low to lead to a safety critical situation during the roughly 1400 s long drive. It also needs to be mentioned that map data itself is not perfect and can be deprecated or be unavailable, especially during the first seconds after the driver has taken a different route than expected by the system in real test drives, which poses a higher risk in practice. Nevertheless, a slight shift to higher velocities is noticeable when $\Delta t_{\mathrm{Lim,TJY}}$ is reduced from 2 s to 1 s, which leads to a reduction in the energy consumption if the exceedance occurs in curves.

Figure 6.8 shows histograms of the relative exceedance for two values of the weight of power error square R_P^{-1}, whereby $\Delta t_{\mathrm{Lim,TJY}} = 1$ s, $\Delta t_{\kappa} = 1.5$ s and $I = 1$. The results indicate that stronger penalization of the electrical traction power mainly leads to an in general slower drive and is unlikely to cause an exceedance of $v_{\mathrm{Lim,Map},\dot{v}}$ in curves for the chosen settings.

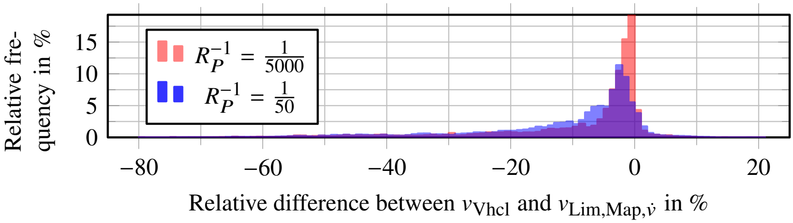


Figure 6.8: Histograms of the relative frequencies of relative differences between vehicle velocity v_{Vhcl} and speed limit from map data with desired acceleration $v_{\text{Lim,Map},\dot{v}}$ for different weight of power error square R_P^{-1} with $\Delta t_{\text{Lim,TJY}} = 1$ s, $\Delta t_\kappa = 1.5$ s and $I = 1$.

Saving potential and assessment

□ For the parameter combination $\Delta t_{\text{Lim,TJY}} = 1$ s, $\Delta t_\kappa = 1.5$ s and $I = 1$, the ALC achieves on average the largest absolute energy savings with respect to MLC. With additionally $R_P^{-1} = \frac{1}{1000}$, the ALC requires 3.4 % less energy than the MLC at the same average velocity of 59.0 km/h, and for the same energy of 16.9 kWh/100km the ALC achieves a 2.6 % higher average velocity. Increasing R_P^{-1} from $\frac{1}{1000}$ to $\frac{1}{50}$, reduces the energy consumption by 2.9 % while the average velocity decreases by 3.1 %.

Table 2.2 on page 11 supports assessing the time and energy savings by summarizing the results for different types of vehicles, roads and ALCs. The results in the first three lines were obtained for the WR, however for a conventional vehicle. The saving potential of a BEV is much lower than for a conventional or hybrid vehicle because of three reasons:

- *Recuperation capability*: The table indicates that when investigations include a highway section, the saving potential is in general less than when only urban and country road sections are considered. This seems reasonable as drives on the latter types of roads usually incorporate much more acceleration and deceleration than highways and kinetic energy cannot be recovered for later use without losses. While conventional vehicles need to convert all kinetic energy into heat, hybrid vehicles can recuperate at least some of it. BEVs have the highest recuperation power limit and therefore can reuse much more energy.

- *Efficient power train*: The tank to wheel efficiency (c.f. Section 4.4) of a BEV is much higher, especially because the electric motor does not perform the inefficient conversion from the chemical energy of the fuel into kinetic energy as a combustion engine in a conventional vehicle or hybrid electric vehicle (HEV) does. An efficiency of up to 95 % for electric motors and of up to 35% for petrol engines are stated in [10]. As part of the electricity comes from non-renewable energy sources, an inefficient energy conversion process usually also takes place for the operation of a BEV but it occurs earlier during the electricity production from non-renewable energy sources and is therefore less frequently considered. Since the mix of available energy sources varies with the region so does the efficiency of fuel and electricity generation. If these processes are efficient, the total efficiency also referred to as *well to wheel efficiency* can be twice as high for a BEV than for a conventional vehicle. The gear box in a BEV is also more efficient since it usually only has a fixed gear ratio and therefore much less moving parts [10].

- *Less degrees of freedom for optimization of driving stategy*: Since most BEVs have a fixed gear ratio the only degree of freedom is the choice of torque, which is however strongly coupled to the trip time and constrained quantities such as velocity and acceleration. Opposed to that, the power train of a conventional vehicle offers selected gear and clutch state as additional degrees of freedom for optimization. In a HEV even the way internal combustion engine, electric motor and battery work together can be chosen. For that reason the latter vehicle types benefit strongly from a look-ahead driving strategy.

Due to these aspects it is no surprise that the largest time saving of 21.13 % at the same energy consumption is stated for a HEV and the largest energy saving of 23.8 %, that however is accompanied by a 1.5 % longer trip time, is reported for a conventional vehicle. These large savings were achieved in investigations including country road and urban sections though.

For a BEV the results of Table 2.2 state average energy savings of 1.21 % on a 1400 km long highway section and of 5.6 % on an only 400 m artificial road profile which comes closest to an urban section due to its velocity of roughly 30 km/h and maximum absolute slopes of 10 %. Compared to these results, the savings achieved by the ALC of this work seem reasonable. *Adopted from [79].*

6.4 Technical contribution

On the level of driver assistance systems, the contribution of this work is the development of an energy-efficient automated longitudinal control for a BEV and its parametrization and test in simulations and real test drives. The developed ALC system shares some similiarities with the solution presented in [114, 139, 183]. All systems use predictive route data and measurements of a radar sensor. Each system performs an ACC functionality that is extended by the use of map data. This enables the system to choose a suitable velocity depending on legal speed limits and road geometry.

The systems can provide the most benefit on winding country roads, on which the standard ACC requires frequent manual adaption of the desired velocity. The systems can also be used on highways and within cities. However, on highways there is a comfort advantage over ACC only when the legal speed limit changes and within cities the driver frequently has to interven because the systems do no consider right of way or traffic signs. Furthermore, other traffic participants are only detected by a radar sensor.

Power train models enable the systems to determine a more energy efficient driving strategy compared to manual driving.

The differences and unique features of the system described in this work refer to the target vehicle and especially the approach for planning the course of velocity. The two target vehicles in [139] have an internal combustion engine and the two target vehicles in [183] a hybrid power train. Therefore rather complex planning approaches that consider gear selection are applied to derive an energy-efficient course of velocity. □ In contrast a BEV like the target vehicle for this work usually has a 1-speed gear box, i.e. a constant gear ratio. The proposed ALC includes a novel trajectory planning method that takes advantage of the simple BEV power train.

Compared to the ALC that was developed for the same research vehicle during the preceding project, the ALC in this work uses a trajectory with respect to time. Especially at low speeds the time-based approach enables a much more dynamic vehicle behavior, which is beneficial for driver acceptance. Furthermore, the ALC of this work considers the required electrical power explicitly and allows to optimize the trajectory not only with respect to travel time and comfort but also with respect to driving efficiency.

In simulative investigations of the energy-saving potential for a BEV the ALC requires up to 3.4 % less energy than the MLC for the same average velocity and achieves a 2.6 % higher average velocity for the same energy consumption on the reference route.

In a final acceptance test about ten employees of the project partners who all work in automotive research and engineering tested the ALC on self chosen routes. The ALC yielded throughout good comments, which especially addressed the comfortable and smooth driving style.

The developed system can be used on its own or in combination with other assistance systems such as an automated lateral control. *Adopted from [79].*

7 Conclusions

7.1 Summary

This work describes novel methods for the general problem of approximating an unbounded number of data points using a B-spline function in the linear and nonlinear weighted least squares (WLS) sense. The developed methods are based on iterative algorithms for state estimation and their computational effort increases linearly with the number of data points. The methods can adjust the bounded definition range of a B-spline function during runtime if this is required to approximate the latest data point.

The approximation problem is reformulated as a trajectory optimization problem such that the approximation methods compute a velocity trajectory with respect to time using data points created from map data. The developed trajectory optimization methods fall into the category of direct methods and its effort increases linearly with the temporal length of the planned trajectory. The combination with an adaptive model that describes the power train properties of the battery electric vehicle (BEV), allows to plan velocity trajectories whose resulting energy consumption varies depending on the chosen relative weighting of different target criteria.

The trajectory optimization is extented to an assistance system for automated longitudinal control (ALC) that is tested in simulation as well as in real test drives. In simulations on a reference route the developed ALC is compared to a recorded and re-simulated real drive with manual longitudinal control (MLC) and can achieve better ratios between average velocity and energy consumption than MLC.

7.2 Contribution

The contribution of this work is threefold and comprises approximation algorithms, a trajectory planning method and a system for energy-efficient ALC of a BEV.

Iterative algorithms are proposed for the approximation of an unbounded set of data as it occurs in online applications. The first algorithm denoted recursive B-spline approximation (RBA) and prepublished in [80] solves a linear WLS approximation problem iteratively using a Kalman filter (KF). The second algorithm is termed nonlinear recursive B-spline approximation (NRBA) and prepublished in [78]. NRBA is generalization of RBA for nonlinear weighted least squares (NWLS) problems and uses a marginalized particle filter (MPF). In the MPF a particle filter (PF) deals with the nonlinear subproblem, whereas a linear KF solves the linear subproblem optimally.

RBA and NRBA include a novel shift operation which allows to shift the bounded definition range of a B-spline function during run-time such that it is always possible to take into account the latest data point. With previous recursive algorithms the approximation interval is fixed during run-time and if data outside this interval occurs, it cannot be considered. Moreover, the shift operation allows to decrease the size of vectors in the filters in order to reduce computational effort in both offline and online applications. In offline applications all data points are available at once, so their number is bounded and batch processing in one step is possible. In contrast, in online applications, additional data points are oberved in each time step, therefore a previously calculated solution must be updated with each new observation.

The exponential growth of computational effort with increasing time horizon limits the application of direct trajectory optimization approaches to short time horizons. □ This work presents a direct trajectory optimization method whose computational effort only grows linearly with the number of function parameters or the time horizon.

This substantial saving in computational effort is achieved by taking adantage of the comparatively simple power train of a BEV with fixed gear ratio and by formulating constraints coming from the vehicle dynamics or the environment mainly as soft constraints. The resulting nonlinear optimization problem is formulated as an NWLS problem and solved using NRBA. The trajectory can be optimized with respect to travel time, comfort and energy consumption. Without consideration of energy consumption the nonlinear optimization problem can be approximated by a quadratic optimization problem that is solved with less computational effort using RBA. *Adopted from [79].*

The trajectory is defined by a B-spline function that describes the desired vehicle velocity with respect to time. The temporal length of the trajectory increases with the iterations and the number of function parameters does not need to be bounded. If the temporal length of the trajectory must be increased, function parameters can be added during the optimization. After each iteration an intermediate result is available. The optimization can be paused and continued in accordance with computation time constraints.

□ Based on the novel trajectory planning approach a system for energy-efficient automated longitudinal control of a BEV is developed and tested in simulation and real test drives. The system can provide the most benefit on winding country roads, but can also be used on highways and within cities.

In simulations on a chosen reference route the ALC needs up to 3.4 % less energy than the MLC for the same average velocity and achieves a 2.6 % higher average velocity for the same energy consumption as with MLC.

For additional subjective impressions, employees in automotive research and engineering tested the ALC on other, individually chosen routes. The throughout good reviews especially mentioned the comfortable and smooth driving style of the ALC. *Adopted from [79].*

7.3 Suggestions for further research

The approximation algorithms RBA and NRBA have only been investigated for B-spline functions. According to Section 2.3, they can be extended to multidimensional control points for approximation of parametric curves and surfaces. Furthermore, RBA and NRBA can be transferred to other spline representations which offer local control and arise from a linear combination of basis functions and control points.

□ Additionally, the modular concept of these algorithms allows to replace the included filters. The KF used for the linear approximation is known to be optimal, but for the PF and the MPF used for the nonlinear approximation various improvements are available in literature. For example, [203] proposes a MPF that determines the particles of the PF using particle swarm optimization and achieves better results at lower effort compared to both a standard MPF and PF. *Adopted from [78].* Even more promising for the nonlinear approximation seems using a kernel adaptive filter as it does not suffer from the curse of dimensionality and requires only solving a convex problem, which is very beneficial for real-time applications (c.f. Section 2.4 and Subsection 4.7.2).

Regarding trajectory planning the computation of the temporal safety margin to upper speed limit can be enhanced such that the trajectory follows the upper speed limit closer at high speeds while still remaining below local minima of the upper speed limit.

A more comprehensive improvement consists of an adaptive selection of the values of temporal distance of neighboring knots, the temporal safety margin to upper speed limit and the temporal distance of neighboring data points so that the trajectory degrees of freedom per trajectory time unit are not constant but depend on the driving situation. For example, on country roads and highways with simple shape of the upper speed limit, temporal distance of neighboring knots and temporal distance of neighboring data points can be increased. Then, for the same computational effort, longer, more farsighted trajectories can be planned because the tractory degrees of freedom are temporally less dense. The resulting larger time horizon, that can be optimized simultaneously by the proposed local trajectory optimization approach, facilitates achieving higher reductions of energy consumption. In contrast, in situations with complicated upper speed limit or at low speeds, the values can be reduced to enable a more dynamic trajectory. Analogously, the relative weighting of the trajectory optimization can be designed adaptive.

Moreover, additional information can be used for calculating the upper speed limit. This can be either data from vehicle-based sensors such as cameras for detecting traffic signs and other traffic participants like pedestrians or it can be messages of other vehicles or the infrastructure. In particular the driving style should be adapted according to the present weather and friction coefficient between tire and road [19].

The mentioned improvements of the trajectory planning directly translate to improvements of the ALC system which can also be extended by other systems such as a lateral control system.

Alternatively, the trajectory optimization approach can be combined with Dynamic Programming (DP), as [187, pp. 1430-1431] suggests and Table 2.5 illustrates, in order to solve more complex problems, for which the presented filter-based approach on its own seems not suitable.

In such a case, DP can still plan in the spatial domain, which can be even more coarsely discretized for less computational effort because a rough long-term trajectory is sufficient. From this long-term trajectory the filter-based trajectory optimization approach creates a smooth trajectory in time domain with the benefits mentioned in Section 5.2 that can represent the vehicle velocity starting from zero so that the case of driving off can also be covered by the trajectory optimization.

List of Figures

List of Tables

List of Algorithms

References

[1] S. Akhegaonkar, S. Glaser, L. Nouveliere, and F. Holzmann. “Smart and Green ACC series: A city and highway specific approach towards a safe and efficient eDAS”. In: *World Electric Vehicle Journal* 6.1 (2013), pp. 214–222. DOI: 10.3390/wevj6010214.

[2] A. Alessandri, M. Cuneo, S. Pagnan, and M. Sanguineti. “A recursive algorithm for nonlinear least-squares problems”. In: *Computational Optimization and Applications* 38.2 (2007), pp. 195–216. DOI: 10.1007/s10589-007-9047-7.

[3] M. Amirfakhrian and H. Mafikandi. “Approximation of parametric curves by Moving Least Squares method”. In: *Applied Mathematics and Computation* 283 (2016), pp. 290–298. DOI: 10.1016/j.amc.2016.02.039.

[4] M. Amirfakhrian and S. Didab. “Approximation of Parametric Functions by Bicubic B-spline Functions”. In: *Journal of American Science* 9.1 (2013).

[5] E. Angel and D. Shreiner. *Interactive Computer Graphics - A Top-Down Approach with Shrader-based opengl*. 6th ed. Pearson Higher Education, 2011.

[6] I. Arasaratnam and S. Haykin. “Cubature Kalman Filters”. In: *Automatic Control, IEEE Transactions on* 54.6 (2009), pp. 1254–1269. DOI: 10.1109/TAC.2009.2019800.

[7] I. Arasaratnam and S. Haykin. “Cubature Kalman Smoothers”. In: *Automatica* 47.10 (2011), pp. 2245–2250.

[8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking”. In: *IEEE Transactions on Signal Processing* 50.2 (2002), pp. 174–188. DOI: 10.1109/78.978374.

[9] K. Arun and L. K. Govil. “Interpolation of natural cubic spline”. In: *International Journal of Mathematics and Mathematical Sciences* 15 (1992).

[10] J. Asamer, A. Graser, B. Heilmann, and M. Ruthmair. "Sensitivity Analysis for Energy Demand Estimation of Electric Vehicles". In: *Transportation Research Part D Transport and Environment* 46 (2016). DOI: 10.1016/j.trd.2016.03.017.

[11] H. B. Jond, V. V. Nabiyev, and R. Benveniste. "Trajectory Planning Using High Order Polynomials under Acceleration Constraint". In: *Journal of Optimization in Industrial Engineering* 10.21 (2016), pp. 1–6. DOI: 10.22094/joie.2016.255.

[12] D. Baek and N. Chang. "Runtime Power Management of Battery Electric Vehicles for Extended Range With Consideration of Driving Time". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.03 (2019), pp. 549–559. DOI: 10.1109/TVLSI.2018.2880441.

[13] V. L. Bageshwar, W. L. Garrard, and R. Rajamani. "Model predictive control of transitional maneuvers for adaptive cruise control vehicles". In: *IEEE Transactions on Vehicular Technology* 53.5 (2004), pp. 1573–1585. DOI: 10.1109/TVT.2004.833625.

[14] M. Bargende, H. Reuss, and J. Wiedemann. *14. Internationales Stuttgarter Symposium: Automobil- und Motorentechnik*. Springer Wiesbaden, 2014, pp. 19–29. ISBN: 9783658051303.

[15] G. Beliakov. "Least squares splines with free knots: Global Optimization approach". In: *School of Computing and Mathematics* (2004).

[16] S. Bender et al. "e-generation - Ein Forschungsprojekt mit positiver Bilanz". In: *Porsche Engineering Magazin* 2 (2015). Visited on 28/04/2018, pp. 22–27. URL: https://www.porscheengineering.com/filestore/download/peg/de/pemagazin-02-2015-artikel-05/default/ca827e19-bdd6-11e5-8bd4-0019999cd470/Forschungsprojekt-e-generation-Eine-positive-Bilanz-Porsche-Engineering-Magazin-02-2015.pdf.

[17] T. Berglund et al. "Planning Smooth and Obstacle-Avoiding B-Spline Paths for Autonomous Mining Vehicles". In: *IEEE Transactions on Automation Science and Engineering* 7.1 (2010), pp. 167–172.

[18] L. Bertoni et al. “An adaptive cruise control for connected energy-saving electric vehicles”. In: *IFAC-PapersOnLine* 50.1 (2017), 2359–2364. DOI: 10.1016/j.ifacol.2017.08.425.

[19] C. Bian, T. Zhu, G. Yin, and L. Xu. “Integrated Speed Planning and Friction Coefficient Estimation Algorithm for Intelligent Electric Vehicles”. In: *Algorithms* 12.2 (2019). DOI: 10.3390/a12020044.

[20] Å. Björck. *Numerical Methods in Matrix Computations*. Vol. 59. Texts in Applied Mathematics. Springer International Publishing, 2015. DOI: 10.1007/978-3-319-05089-8.

[21] F. Bleimund and S. Rhode. “Method, computer program product, device, and vehicle for calculating an actuation variable for the operation of a vehicle”. European patent application. EP 2 886 409 A1. 2015.

[22] F. Bleimund, D. Dörr, B. Fath, and M. Frey. *Verbundprojekt "Schlüsseltechnologien für die nächste Generation der Elektrofahrzeuge (e-generation)": Teilvorhaben KIT: "Assistenzsysteme für effizienten Energieeinsatz bei Elektrofahrzeugen": Schlussbericht: Laufzeit des Vorhabens: 01.02.2012-31.12.2014*. Tech. rep. Karlsruher Institut für Technologie, Institut für Fahrzeugsystemtechnik, Lehrstuhl für Fahrzeugtechnik, 2015. DOI: 10.2314/GBV:86364080X.

[23] M. J. Bossé, W. Bauldry, and S. Otey. “Locating Complex Roots of Quintic Polynomials”. In: *The Mathematics Enthusiast* 15.3 (2018). Visited on 01/04/2020, pp. 529–540.

[24] H. Braess and U. Seiffert. *Vieweg Handbuch Kraftfahrzeugtechnik*. 7th ed. Springer Wiesbaden, 2013. ISBN: 9783658016913. DOI: 10.1007/978-3-658-01691-3.

[25] J. R. Bunch and J. E. Hopcroft. “Triangular Factorization and Inversion by Fast Matrix Multiplication”. In: *Mathematics of Computation* 28.125 (1974), pp. 231–236. DOI: 10.2307/2005828.

[26] B. Cao, G. I. Doods, and G. W. Irwin. “Time-optimal and smooth constrained path planning for robot manipulators”. In: *IEEE International Conference on Robotics and Automation*. Vol. 3. 1994, pp. 1853–1858. DOI: 10.1109/ROBOT.1994.351191.

[27] O. Cappe, S. J. Godsill, and E. Moulines. “An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo”. In: *Proceedings of the IEEE* 95.5 (2007), pp. 899–924. DOI: 10.1109/JPROC.2007.893250.

[28] A. Carvalho et al. “Predictive control of an autonomous ground vehicle using an iterative linearization approach”. In: (Oct. 2013), pp. 2335–2340.

[30] X. Chen, R. F. Riesenfeld, and E. Cohen. “Sliding Windows Algorithm for B-spline Multiplication”. In: *Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling*. ACM, 2007, pp. 265–276. DOI: 10.1145/1236246.1236283.

[31] X.-D. Chen, W. Ma, and J.-C. Paul. “Cubic B-spline curve approximation by curve unclamping”. In: *Computer-Aided Design* 42.6 (2010), pp. 523–534. DOI: 10.1016/j.cad.2010.01.008.

[33] B. Y. Choi and Z. Bien. “Sliding-windowed weighted recursive least-squares method for parameter estimation”. In: *Electronics Letters* 25.20 (1989), pp. 1381–1382. DOI: 10.1049/el:19890924.

[34] I. L. u. J. Z. Christoph Stiller Sören Kammel. “Probalistische Methoden in der Umfeldwahrnehmung Kognitiver Automobile”. In: *at - Automatisierungstechnik* 56.11 (2008). DOI: 10.1524/auto.2008.0738.

[35] W. Dahmen and A. Reusken, eds. *Numerik für Ingenieure und Naturwissenschaftler*. 2nd ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. DOI: 10.1007/978-3-540-76493-9.

[36] C. Deng and H. Lin. “Progressive and iterative approximation for least squares B-spline curve and surface fitting”. In: *Computer-Aided Design* 47 (2014), pp. 32–44. DOI: 10.1016/j.cad.2013.08.012.

[37] F. Deng, J. Chen, and C. Chen. “Adaptive unscented Kalman filter for parameter and state estimation of nonlinear high-speed objects”. In: *Journal of Systems Engineering and Electronics* 24.4 (2013), pp. 655–665. DOI: 10.1109/JSEE.2013.00076.

[38] F. M. Dias, A. Antunes, J. Vieira, and A. Mota. “A sliding window solution for the on-line implementation of the Levenberg-Marquardt algorithm”. In: *Engineering Applications of Artificial Intelligence* 19.1 (2006), pp. 1–7. DOI: 10.1016/j.engappai.2005.03.005.

[39] W. Dib et al. “Optimal energy management for an electric vehicle in eco-driving applications”. In: *Control Engineering Practice* 29 (2014), pp. 299–307. DOI: 10.1016/j.conengprac.2014.01.005.

[40] S. Dixit et al. “Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects”. In: *Annual Reviews in Control* 45 (2018), pp. 76–86. DOI: 10.1016/j.arcontrol.2018.02.001.

[41] D. Dörr, B. Fath, J. Jauch, and M. Frey. *Verbundprojekt: "Gesamtfahrzeugintegration innovativer Konzepte für ein effizientes und performantes E-Fahrzeug (e-volution)", Teil vorhaben KIT: "Assistenzsysteme für effiziente und performante Elektrofahrzeuge" : Schlussbericht*. Tech. rep. Karlsruher Institut für Technologie (KIT), 2017.

[42] M. Du et al. “Drivers’ Visual Behavior-Guided RRT Motion Planner for Autonomous On-Road Driving”. In: *Sensors* 16.1 (2016). DOI: 10.3390/s16010102.

[43] V. T. Dung and T. Tjahjowidodo. “A direct method to solve optimal knots of B-spline curves: An application for non-uniform B-spline curves fitting”. In: *PLoS ONE* 12.3 (2017). DOI: 10.1371/journal.pone.0173857.

[44] D. Efremov, M. Klaučo, T. Haniš, and M. Hromčík. “Driving Envelope Definition and Envelope Protection Using Model Predictive Control”. In: *2020 American Control Conference (ACC)*. July 2020, pp. 4875–4880. DOI: 10.23919/ACC45564.2020.9147211.

[45] M. Elbanhawi, M. Simic, and R. N. Jazar. “Continuous Path Smoothing for Car-Like Robots Using B-Spline Curves”. In: *Journal of Intelligent & Robotic Systems* 80.1 (2015), pp. 23–56. DOI: 10.1007/s10846-014-0172-0.

[46] EVOlution Graphics B.V. *the-blueprints.com - Porsche Boxster 981*. Visited on 27/05/2018. URL: https://www.the-blueprints.com/vectordrawings/show/8051/porsche_boxster_981/.

[47] P. Falcone et al. “A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems”. In: *46th IEEE Conference on Decision and Control*. 2007, pp. 2980–2985. DOI: 10.1109/CDC.2007.4434137.

[48] C. Fiori, K. Ahn, and H. A. Rakha. “Power-based electric vehicle energy consumption model: Model development and validation”. In: *Applied Energy* 168 (2016), pp. 257–268. DOI: 10.1016/j.apenergy.2016.01.097.

[50] H. Fujioka and H. Kano. “Recursive construction of smoothing spline surfaces using normalized uniform B-splines”. In: *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. 2009, pp. 5550–5555. DOI: 10.1109/CDC.2009.5399523.

[51] R. Galvin. “Energy consumption effects of speed and acceleration in electric vehicles: Laboratory case studies and implications for drivers and policymakers”. In: *Transportation Research Part D: Transport and Environment* 53 (2017), pp. 234–248. DOI: 10.1016/j.trd.2017.04.020.

[52] Z. Gao and H. Song. “Study on the Electric Vehicle Adaptive Cruise Control Based on the Model Predictive Control Algorithm”. In: *Green Intelligent Transportation Systems*. Ed. by W. Wang, K. Bengler, and X. Jiang. Springer Singapore, 2019, pp. 39–51. ISBN: 978-981-13-0302-9.

[53] Z. Gao, W. Yan, H. Hu, and H. Li. “Human-centered headway control for adaptive cruise-controlled vehicles”. In: *Advances in Mechanical Engineering* 7.11 (2015), pp. 1–13. DOI: 10.1177/1687814015615042.

[54] P. Gáspár and B. Németh. *Predictive Cruise Control for Road Vehicles Using Road and Traffic Information*. Cham: Springer International Publishing, 2019. DOI: 10.1007/978-3-030-04116-8.

[55] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni. “Trajectory Planning in Robotics”. In: *Mathematics in Computer Science* 6.3 (2012), pp. 269–279. DOI: 10.1007/s11786-012-0123-8.

[56] M. Gerdts, S. Karrenberg, B. Müller-Beßler, and G. Stock. “Generating locally optimal trajectories for an automatically driven car”. In: *Optimization and Engineering* 10.4 (2008), p. 439. DOI: 10.1007/s11081-008-9047-1.

[57] J. M. Giron-Sierra. “Kalman Filter, Particle Filter and Other Bayesian Filters”. In: *Digital Signal Processing with Matlab Examples, Volume 3*. Springer Singapore, 2017, pp. 3–148. DOI: 10.1007/978-981-10-2540-2_1.

[58] G. Golub and C. Van Loan. *Matrix Computations*. 3rd ed. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996.

[59] D. González, J. Pérez, V. Milanés, and F. Nashashibi. “A Review of Motion Planning Techniques for Automated Vehicles”. In: *IEEE Transactions on Intelligent Transportation Systems* 17.4 (2016), pp. 1135–1145. DOI: 10.1109/TITS.2015.2498841.

[60] M. S. Grewal and A. P. Andrews. *Kalman Filtering*. John Wiley & Sons, Inc., 2008. DOI: 10.1002/9780470377819.

[61] L. Guo, P. Ge, D. Sun, and Y. Qiao. “Adaptive Cruise Control Based on Model Predictive Control with Constraints Softening”. In: *Applied Sciences* 10.5 (2020). DOI: 10.3390/app10051635.

[62] L. Guzzella and A. Sciarretta. *Vehicle propulsion systems: introduction to modeling and optimization*. 3rd ed. Berlin: Springer, 2013. ISBN: 978-3-642-35912-5; 978-3-642-43847-9; 3-642-35912-4.

[63] X. V. Ha, C. Ha, and J. Lee. “Trajectory Estimation of a Tracked Mobile Robot Using the Sigma-Point Kalman Filter with an IMU and Optical Encoder”. In: *Intelligent Computing Technology*. Ed. by D.-S. Huang, C. Jiang, V. Bevilacqua, and J. C. Figueroa. Springer Berlin Heidelberg, 2012, pp. 415–422.

[64] K.-L. Haken. *Grundlagen der Kraftfahrzeugtechnik*. 5th ed. Hanser, 2018. DOI: 10.3139/9783446455702.

[65] A. Hamednia, N. Murgovski, and J. Fredriksson. “Predictive velocity control in a hilly terrain over a long look-ahead horizon”. In: *IFAC-PapersOnLine* 51.31 (2018). 5th IFAC Conference on Engine and Powertrain Control, Simulation and Modeling E-COSM 2018, pp. 485–492. DOI: 10.1016/j.ifacol.2018.10.107.

[66] S. Hammarling and C. Lucas. “Updating the QR Factorization and the Least Squares Problem”. In: *Manchester Institute for Mathematical Sciences, School of Mathematics, MIMS EPrint 2008.111* (2008). URL: http://www.eprints.ma.man.ac.uk/1192/01/covered/MIMS_ep2008_111.pdf.

[67] M. Harashima, L. Ferrari, and P. Sankar. "Spline Approximation Using Kalman Filter State Estimation". In: *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on* 44.5 (1997), pp. 421–424. DOI: 10.1109/82.580860.

[68] G. T. Haupt, N. J. Kasdin, G. M. Keiser, and B. W. Parkinson. "Optimal recursive iterative algorithm for discrete nonlinear least-squares estimation". In: *Journal of Guidance, Control, and Dynamics 1996* 19.3 (1996), pp. 643–649.

[69] S. He, D. Ou, C. Yan, and C.-H. Lee. "A chord error conforming tool path B-spline fitting method for NC machining based on energy minimization and LSPIA". In: *Journal of Computational Design and Engineering* 2.4 (2015). CNCM 2015 Special Issue, pp. 218–232. DOI: 10.1016/j.jcde.2015.06.002.

[70] G. Hendeby, R. Karlsson, and F. Gustafsson. "A New Formulation of the Rao-Blackwellized Particle Filter". In: *2007 IEEE/SP 14th Workshop on Statistical Signal Processing*. 2007, pp. 84–88. DOI: 10.1109/SSP.2007.4301223.

[71] X. Hu, S. Li, and H. Peng. "A comparative study of equivalent circuit models for Li-ion batteries". In: *Journal of Power Sources* 198 (2012), pp. 359–367. DOI: 10.1016/j.jpowsour.2011.10.013.

[72] M. Huber. "Nonlinear Gaussian Filtering : Theory, Algorithms, and Applications". Habilitation. 2015. DOI: 10.5445/IR/1000046060.

[73] J. Izadian and N. Farahbakhsh. "Solving Nonlinear Least Squares Problems with B-Spline Functions". In: *Applied Mathematical Sciences* (2012).

[74] J. Jauch*, J. Masino*, T. Staiger, and F. Gauterin. "Road Grade Estimation With Vehicle-Based Inertial Measurement Unit and Orientation Filter". In: *IEEE Sensors Journal* 18.2 (2018). * Equal contribution, pp. 781–789. DOI: 10.1109/JSEN.2017.2772305.

[75] J. Jauch. *ALC Matlab files*. 2019. DOI: 10.5281/zenodo.3334806. URL: http://github.com/JensJauch/ALC/releases.

[76] J. Jauch. *NRBA Matlab files*. 2019. DOI: 10.5281/zenodo.2604318. URL: http://github.com/JensJauch/nrba.

[77] J. Jauch. *RBA source code*. 2016. DOI: 10.5281/zenodo.157732. URL: http://github.com/JensJauch/RBA.

[78] J. Jauch, F. Bleimund, M. Frey, and F. Gauterin. “An Iterative Method Based on the Marginalized Particle Filter for Nonlinear B-Spline Data Approximation and Trajectory Optimization”. In: *Mathematics* 7.4 (2019). DOI: 10.3390/math7040355.

[79] J. Jauch, F. Bleimund, M. Frey, and F. Gauterin. “Automated Longitudinal Control Based on Nonlinear Recursive B-Spline Approximation for Battery Electric Vehicles”. In: *World Electric Vehicle Journal* 10.3 (2019). DOI: 10.3390/wevj10030052.

[80] J. Jauch, F. Bleimund, S. Rhode, and F. Gauterin. “Recursive B-spline approximation using the Kalman filter”. In: *Engineering Science and Technology, an International Journal* 20.1 (2017), pp. 28–34. DOI: 10.1016/j.jestch.2016.09.015.

[81] Y. Jia, R. Jibrin, Y. Itoh, and D. Görges. “Energy-Optimal Adaptive Cruise Control for Electric Vehicles in Both Time and Space Domain based on Model Predictive Control”. In: *IFAC-PapersOnLine* 52.5 (2019). 9th IFAC Symposium on Advances in Automotive Control AAC 2019, pp. 13–20. DOI: 10.1016/j.ifacol.2019.09.003.

[82] Y. Jia et al. “Energy-Optimal Adaptive Cruise Control in Time Domain based on Model Predictive Control”. In: *IFAC-PapersOnLine* 51.31 (2018). 5th IFAC Conference on Engine and Powertrain Control, Simulation and Modeling E-COSM 2018, pp. 846–853.

[83] J. Jiang and Y. Zhang. “A revisit to block and recursive least squares for parameter estimation”. In: *Computers & Electrical Engineering* 30.5 (2004), pp. 403–416. DOI: 10.1016/j.compeleceng.2004.05.002.

[84] Z. Jiang. “A New Approximation Method with High Order Accuracy”. In: *Mathematical and Computational Applications* 22.1 (2017). DOI: 10.3390/mca22010011.

[86] L. Johannesson et al. “Predictive energy management of hybrid long-haul trucks”. In: *Control Engineering Practice* 41 (2015), 83–97. DOI: 10.1016/j.conengprac.2015.04.014.

[87] C. S. Julius Ziegler. “Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios”. In: *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2009), pp. 1879–1884.

[88] M. Kalabis and S. Müller. “A Model Predictive Approach for a Fuel Efficient Cruise Control System”. In: *Zukünftige Entwicklungen in der Mobilität: Betriebswirtschaftliche und technische Aspekte*. Ed. by H. Proff, J. Schönharting, D. Schramm, and J. Ziegler. Wiesbaden: Gabler, 2012, pp. 201–211. DOI: 10.1007/978-3-8349-7117-3_16.

[89] P. S. Karlsson, L. Behrenz, and G. Shukur. “Performances of Model Selection Criteria When Variables are Ill Conditioned”. In: *Computational Economics* (25, 2017). DOI: 10.1007/s10614-017-9682-8.

[90] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka. “Real-time planning methods for autonomous on-road driving: State-of-the-art and future research directions”. In: *Elsevier, Transportation Research Part C* (2015).

[91] L. J. Kere, S. Kelouwani, K. Agbossou, and Y. Dubé. “Improving Efficiency Through Adaptive Internal Model Control of Hydrogen-Based Genset Used as a Range Extender for Electric Vehicles”. In: *IEEE Transactions on Vehicular Technology* 66.6 (2017), pp. 4716–4726.

[92] Z. Khalik, G. P. Padilla, T. C. J. Romijn, and M. C. F. Donkers. “Vehicle Energy Management with Ecodriving: A Sequential Quadratic Programming Approach with Dual Decomposition”. In: *2018 Annual American Control Conference (ACC)*. 2018, pp. 4002–4007. DOI: 10.23919/ACC.2018.8431544.

[93] Y. Kineri, M. Wang, H. Lin, and T. Maekawa. “B-spline surface fitting by iterative geometric interpolation/approximation algorithms”. In: *Computer-Aided Design* 44.7 (2012), pp. 697–708. DOI: 10.1016/j.cad.2012.02.011.

[94] M. Kleder. *quadprog2 - convex QP solver*. Visited on 24/01/2018. 2005. URL: http://mathworks.com/matlabcentral/fileexchange/7860-quadprog2-convex-qp-solver.

[95] B. Kouvaritakis and M. Cannon. *Model Predictive Control - Classical, Robust and Stochastic*. Springer, Cham, 2016. DOI: 10.1007/978-3-319-24853-0.

[96] J. Kühlwein. “Driving resistance of light-duty vehicles in Europe: Present situation, trends, and scenarios for 2025”. In: *International Council on Clean Transportation Europe* (2016). Visited on 01/04/2020. URL: https://theicct.org/sites/default/files/publications/ICCT_LDV-Driving-Resistances-EU_121516.pdf.

[97] F. Le Gall. “Faster Algorithms for Rectangular Matrix Multiplication”. In: *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*. 2012, pp. 514–523. DOI: 10.1109/FOCS.2012.80.

[98] C. Lee. “Adaptive Cruise Control mit einer modellprädiktiven Regelung”. Master thesis. Karlsruher Institut für Technologie, Institut für Fahrzeugsystemtechnik, 2016.

[99] Y. Lee. “Optimization of Moving Objects Trajectory Using Particle Filter”. In: *Intelligent Computing Theory*. Ed. by D.-S. Huang, V. Bevilacqua, and P. Premaratne. Cham: Springer International Publishing, 2014, pp. 55–60.

[100] T. Levermore, M. N. Sahinkaya, Y. Zweiri, and B. Neaves. “Real-Time Velocity Optimization to Minimize Energy Use in Passenger Vehicles”. In: *Energies* 10.1 (2017). DOI: 10.3390/en10010030.

[101] B. Li, J. Zhang, H. Du, and W. Li. “Two-layer structure based adaptive estimation for vehicle mass and road slope under longitudinal motion”. In: *Measurement* 95 (2017), pp. 439–455. DOI: 10.1016/j.measurement.2016.10.045.

[102] S. E. Li, F. Gao, D. Cao, and K. Li. “Multiple-Model Switching Control of Vehicle Longitudinal Dynamics for Platoon-Level Automation”. In: *IEEE Transactions on Vehicular Technology* 65.6 (2016), pp. 4480–4492.

[103] M. Likhachev and D. Ferguson. “Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles”. In: *The International Journal of Robotics Research* 28.8 (2009), pp. 933–945. DOI: 10.1177/0278364909340445.

[104] K. H. Lim, K. P. Seng, and L.-M. Ang. “River Flow Lane Detection and Kalman Filtering-Based B-Spline Lane Tracking”. In: *International Journal of Vehicular Technology* (2012). DOI: 10.1155/2012/465819.

[105] C. Lin, Q. Yu, R. Xiong, and L. Y. Wang. "A study on the impact of open circuit voltage tests on state of charge estimation for lithium-ion batteries". In: *Applied Energy* 205 (2017), pp. 892–902. DOI: 10.1016/j.apenergy.2017.08.124.

[106] H. Lin, G. Wang, and C. Dong. "Constructing iterative non-uniform B-spline curve and surface to fit data points". In: *Science in China Series : Information Sciences* 47.3 (2004), pp. 315–331. DOI: 10.1360/02yf0529.

[107] X. Lin, D. Görges, and A. Weißmann. "Simplified Energy-Efficient Adaptive Cruise Control based on Model Predictive Control". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 4794–4799. DOI: 10.1016/j.ifacol.2017.08.963.

[108] H. Liu, X. Lai, and W. Wu. "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints". In: *Robotics and Computer-Integrated Manufacturing* 29.2 (2013), pp. 309–317. DOI: 10.1016/j.rcim.2012.08.002.

[109] J. Liu, Z. Wang, and M. Xu. "A Kalman Estimation Based Rao-Blackwellized Particle Filtering for Radar Tracking". In: *IEEE Access* 5 (2017), pp. 8162–8174. DOI: 10.1109/ACCESS.2017.2693288.

[110] M. Liu et al. "Progressive iterative approximation for regularized least square bivariate B-spline surface fitting". In: *Journal of Computational and Applied Mathematics* 327 (2018), pp. 175–187. DOI: 10.1016/j.cam.2017.06.013.

[111] W. Liu, J. C. Principe, and S. Haykin. *Kernel Adaptive Filtering: A Comprehensive Introduction*. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control. Wiley, 2011. ISBN: 9781118211212.

[112] L.-H. Luo, H. Liu, P. Li, and H. Wang. "Model predictive control for adaptive cruise control with multi-objectives: comfort, fuel-economy, safety and car-following". In: *Journal of Zhejiang University SCIENCE A* 11.3 (2010), pp. 191–201. DOI: 10.1631/jzus.A0900374.

[113] T. Lyche and K. Mørken. *Spline Methods Draft*. Department of Informatics, Centre of Mathematics for Applications, University of Oslo, 2008.

[114] P. Markschläger, H.-G. Wahl, F. Weberbauer, and M. Lederer. "Assistenzsystem für mehr Kraftstoffeffizienz". In: *Vernetztes Automobil: Sicherheit - Car-IT - Konzepte*. Ed. by W. Siebenpfeiffer. Springer Wiesbaden, 2014, pp. 146–153. DOI: 10.1007/978-3-658-04019-2_21.

[115] D. W. Marquardt. "An Algorithm for Least-Squares Estimation of Nonlinear Parameters". In: *Journal of the Society for Industrial and Applied Mathematics* 11.2 (1963), pp. 431–441. DOI: 10.1137/0111030.

[116] J. A. Matute, M. Marcano, A. Zubizarreta, and J. Perez. "Longitudinal Model Predictive Control with comfortable speed planner". In: *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. 2018, pp. 60–64.

[117] G. Meyer and S. Beiker. *Road Vehicle Automation*. 2018. DOI: 10.1007/978-3-319-05990-7.

[118] W. W. Min Zhou Hui Jin. "A review of vehicle fuel consumption models to evaluate eco-driving and eco-routing". In: *Transportation Research Part D* (2016).

[119] M. Mitschke and H. Wallentowitz. *Dynamik der Kraftfahrzeuge*. Springer Wiesbaden, 2014. ISBN: 9783658050689.

[120] A. Monir and H. Mraoui. "Spline approximations of the Lambert W function and application to simulate generalized Gaussian noise with exponent $\alpha = 1/2$". In: *Digital Signal Processing* 33 (2014), pp. 34–41. DOI: 10.1016/j.dsp.2014.06.013.

[121] K. Mørken. "Some identities for products and degree raising of splines". In: *Constructive Approximation* 7.1 (1991), pp. 195–208. DOI: 10.1007/BF01888153.

[122] M. Nieto et al. "Real-time lane tracking using Rao-Blackwellized particle filter". In: *Journal of Real-Time Image Processing* 11.1 (2016), pp. 179–191. DOI: 10.1007/s11554-012-0315-0.

[123] M. Nyolt et al. "Marginal filtering in large state spaces". In: *International Journal of Approximate Reasoning* 61 (2015), pp. 16–32. DOI: 10.1016/j.ijar.2015.04.003.

[124] O. Olsson and T. Ivarsson. *Using the QR Factorization to swiftly update least squares problems*. student paper. 2014.

[125] S. J. Olszanskyj, J. M. Lebak, and A. W. Bojanczyk. “Rank-k modification methods for recursive least squares problems”. In: *Numerical Algorithms* 7.2 (1994), pp. 325–354. DOI: 10.1007/BF02140689.

[126] E. Ozatay, U. Ozguner, and D. Filev. “Velocity profile optimization of on road vehicles: Pontryagin’s Maximum Principle based approach”. In: *Control Engineering Practice* 61 (2017), pp. 244–254. DOI: 10.1016/j.conengprac.2016.09.006.

[127] G. P. Padilla, S. Weiland, and M. C. F. Donkers. “A Global Optimal Solution to the Eco-Driving Problem”. In: *IEEE Control Systems Letters* 2.4 (2018), pp. 599–604.

[128] R. Panda and B. Chatterji. “Least squares generalized B-spline signal and image processing”. In: *Signal Processing* 81.10 (2001), pp. 2005–2017. DOI: 10.1016/S0165-1684(01)00022-6.

[129] R. Panda, G. Rath, and B. Chatterji. “Generalized B-spline signal processing”. In: *Signal Processing* 55.1 (1996), pp. 1–14. DOI: 10.1016/S0165-1684(96)00116-8.

[130] H. Park. “Choosing nodes and knots in closed B-spline curve interpolation to point data”. In: *Computer-Aided Design* 33.13 (2001), pp. 967–974. DOI: 10.1016/S0010-4485(00)00133-0.

[131] B. Passenberg. “Theory and Algorithms for Indirect Methods in Optimal Control of Hybrid Systems”. PhD thesis. Technische Universität München, 2012.

[132] N. Petit and A. Sciarretta. “Optimal drive of electric vehicles using an inversion-based trajectory generation approach”. In: *IFAC Proceedings Volumes* 44.1 (2011). 18th IFAC World Congress, 14519–14526. DOI: 10.3182/20110828-6-IT-1002.01986.

[133] L. Piegl and W. Tiller. “Symbolic Operators for NURBS”. In: *Computer-Aided Design* 29.5 (1997), pp. 361–368.

[134] G. L. Plett. “Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification”. In: *Journal of Power Sources* 134.2 (2004), pp. 262–276. DOI: 10.1016/j.jpowsour.2004.02.032.

[135] K. Qian, X. Ma, X. Dai, and F. Fang. “Improved Rao-Blackwellized particle filter for simultaneous robot localization and person-tracking with single mobile sensor”. In: *Journal of Control Theory and Applications* 9.4 (2011), pp. 472–478. DOI: 10.1007/s11768-011-9105-7.

[136] X. Qian. “Model predictive control for autonomous and cooperative driving”. PhD thesis. Institut des Sciences et Technologies, Paris Institute of Technology, 2016.

[137] S. Qin and T. A. Badgwell. “A survey of industrial model predictive control technology”. In: *Control Engineering Practice* 11.7 (2003), pp. 733–764. DOI: 10.1016/S0967-0661(02)00186-7.

[138] C. Radhakrishna Rao, Shalabh, H. Toutenburg, and C. Heumann. *Linear Models and Generalizations: Least Squares and Alternatives*. 3rd ed. Springer Series in Statistics. Springer Berlin Heidelberg, 2008. DOI: 10.1007/978-3-540-74227-2.

[139] T. Radke. “Energieoptimale Längsführung von Kraftfahrzeugen durch Einsatz vorausschauender Fahrstrategien”. PhD thesis. KIT Scientific Publishing, Karlsruhe, 2013. DOI: 10.5445/KSP/1000035819.

[140] E. Raffone. “Road slope and vehicle mass estimation for light commercial vehicle using linear Kalman filter and RLS with forgetting factor integrated approach”. In: *Proceedings of the 16th International Conference on Information Fusion*. 2013, pp. 1167–1172.

[141] M. Randrianarivony and G. Brunnett. “Approximation by NURBS Curves with Free Knots”. In: *Conference Vision Modeling and Visualization, Erlangen Germany, Akademische Verlagsgesellschaft AKA* (2002), pp. 195–202.

[142] L. Rebollo-Neira and Z. Xu. “Sparse signal representation by adaptive non-uniform B-spline dictionaries on a compact interval”. In: *Signal Processing* 90.7 (2010), pp. 2308–2313. DOI: 10.1016/j.sigpro.2010.02.004.

[143] M. J. Reis, P. J. Ferreira, and S. F. Soares. “Linear combinations of B-splines as generating functions for signal approximation”. In: *Digital Signal Processing* 15.3 (2005), pp. 226–236. DOI: 10.1016/j.dsp.2004.12.003.

[144] S. Rhode, F. Bleimund, and F. Gauterin. "Recursive Generalized Total Least Squares with Noise Covariance Estimation". In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 4637–4643. DOI: 10.3182/20140824-6-ZA-1003.01568.

[145] S. Rhode, S. Hong, J. K. Hedrick, and F. Gauterin. "Vehicle tractive force prediction with robust and windup-stable Kalman filters". In: *Control Engineering Practice 46* (2016).

[146] S. Rhode, S. Van Vaerenbergh, and M. Pfriem. "Power prediction for electric vehicles using online machine learning". In: *Engineering Applications of Artificial Intelligence* 87 (2020), p. 103278. DOI: 10.1016/j.engappai.2019.103278.

[147] S. Richter, C. N. Jones, and M. Morari. "Computational Complexity Certification for Real-Time MPC With Input Constraints Based on the Fast Gradient Method". In: *IEEE Transactions on Automatic Control* 57.6 (2012), pp. 1391–1403. DOI: 10.1109/TAC.2011.2176389.

[148] R. M. Roark and M. A. Escabi. "B-spline design of maximally flat and prolate spheroidal-type FIR filters". In: *IEEE Transactions on Signal Processing* 47.3 (1999), pp. 701–716. DOI: 10.1109/78.747777.

[149] A. Ruhe and P. A. Wedin. "Algorithms for Separable Nonlinear Least Squares Problems". In: *SIAM Review* 22.3 (1980), pp. 318–337. DOI: 10.1137/1022057.

[150] C. Runge. "Über empirische Funktionen und die Interpolation zwischen äquidistanten Orten". In: *Zeitschrift für Mathematik und Physik* 46 (1901), pp. 224–243.

[151] S. Sager. "Numerical methods for mixed-integer optimal control poblems". PhD thesis. Interdisciplinary Center for Scientific Computing, Universität Heidelberg, 2005.

[152] J. Schäuffele and T. Zurawka. *Automotive Software Engineering - Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen*. 2016. DOI: 10.1007/978-3-658-11815-0.

[153] N. Schimmelpfennig Karl-Heinz; Hebing. "Geschwindigkeiten bei kreisförmiger Kurvenfahrt/Stabilitäts- und Sicherheitsgrenze". In: *Der Verkehrsunfall* 20 (1982).

[154] R. Schmied et al. “Nonlinear MPC for Emission Efficient Cooperative Adaptive Cruise Control”. In: *IFAC-PapersOnLine* 48.23 (2015). 5th IFAC Conference on Nonlinear Model Predictive Control NMPC 2015, pp. 160–165. DOI: 10.1016/j.ifacol.2015.11.277.

[155] T. Schön, F. Gustafsson, and P. J. Nordlund. “Marginalized particle filters for mixed linear/nonlinear state-space models”. In: *IEEE Transactions on Signal Processing* 53.7 (2005), pp. 2279–2289. DOI: 10.1109/TSP.2005.849151.

[156] T. Schön. *Rao-Blackwellized particle filter - MATLAB code*. Visited on 16/01/2018. 2011. URL: http://user.it.uu.se/~thosc112/research/rao-blackwellized-particle.html.

[157] D. Schramm, M. Hiller, and R. Bardini. *Vehicle Dynamics - Modeling and Simulation*. Springer, Berlin, Heidelberg, 2018. DOI: 10.1007/978-3-662-54483-9.

[158] H. Schwetlick and T. Schütze. “Least squares approximation by splines with free knots”. In: *BIT Numerical Mathematics* 35.3 (1995), pp. 361–384. DOI: 10.1007/BF01732610.

[159] M. Shahrbanozadeh, G.-A. Barani, and S. Shojaee. “Simulation of flow through dam foundation by isogeometric method”. In: *Engineering Science and Technology, an International Journal* 18.2 (2015), pp. 185–193. DOI: 10.1016/j.jestch.2014.11.001.

[160] D. Shen, D. Karbowski, and A. Rousseau. “Fuel-Optimal Periodic Control of Passenger Cars in Cruise Based on Pontryagin’s Minimum Principle”. In: *IFAC-PapersOnLine* 51.31 (2018). 5th IFAC Conference on Engine and Powertrain Control, Simulation and Modeling E-COSM 2018, pp. 813–820. DOI: 10.1016/j.ifacol.2018.10.123.

[161] P. A. Sherar. “Variational Based Analysis and Modelling using B-splines”. PhD thesis. Cranfield University, School of Engineering, Applied Mathematics and Computing Group, 2003.

[162] C.-L. Shih and L.-C. Lin. “Trajectory Planning and Tracking Control of a Differential-Drive Mobile Robot in a Picture Drawing Application”. In: *Robotics* 6.3 (2017). DOI: 10.3390/robotics6030017.

[163] D. Simon. “Kalman filtering with state constraints: a survey of linear and nonlinear algorithms”. In: *Control Theory Applications, IET* 4.8 (2010), pp. 1303–1318. DOI: 10.1049/iet-cta.2009.0032.

[164] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley, 2006.

[165] N. Sina, V. Esfahanian, M. Hairi Yazdi, and S. Azadi. “Introducing the Modified Tire Power Loss and Resistant Force Regarding Longitudinal Slip”. In: *SAE International Journal of Passenger Cars - Mechanical Systems* 11 (2018). DOI: 10.4271/06-11-02-0014.

[166] P. Skoglar, U. Orguner, D. Tornqvist, and F. Gustafsson. “Road target tracking with an approximative Rao-Blackwellized Particle Filter”. In: *2009 12th International Conference on Information Fusion*. 2009, pp. 17–24.

[168] B. Stenlund and F. Gustafsson. “Avoiding windup in recursive parameter estimation”. In: *Preprints of reglermöte* (2002), pp. 148–153.

[169] J. A. Stiver, X. D. Koutsoukos, and P. J. Antsaklis. “An invariant-based approach to the design of hybrid control systems”. In: *International Journal of Robust and Nonlinear Control* 11.5 (2001), pp. 453–478. DOI: 10.1002/rnc.593.

[170] Y. Sun et al. “A hybrid algorithm combining EKF and RLS in synchronous estimation of road grade and vehicle mass for a hybrid electric bus”. In: *Mechanical Systems and Signal Processing* 68-69 (2016), pp. 416–430. DOI: 10.1016/j.ymssp.2015.08.015.

[171] A. Taghavipour, M. Vajedi, and N. L. Azad. “Ecological Cruise Control”. In: *Intelligent Control of Connected Plug-in Hybrid Electric Vehicles*. Cham: Springer International Publishing, 2019, 169–183. DOI: 10.1007/978-3-030-00314-2_9.

[172] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. SAE International. 2021. DOI: 10.4271/J3016_202104.

[173] A. T. Thorgeirsson and F. Gauterin. “Probabilistic Predictions with Federated Learning”. In: *Entropy* 23.1 (2021). DOI: 10.3390/e23010041.

[174] M. Treiber, A. Hennecke, and D. Helbing. “Congested traffic states in empirical observations and microscopic simulations”. In: *Phys. Rev. E* 62 (2000), pp. 1805–1824. DOI: 10.1103/PhysRevE.62.1805.

[175] N. Ueda Kenji and Yamashita. “On a Global Complexity Bound of the Levenberg-Marquardt Method”. In: *Journal of Optimization Theory and Applications* 147.3 (2010), pp. 443–453. DOI: 10.1007/s10957-010-9731-0.

[176] M. Unser. “Splines: a perfect fit for signal and image processing”. In: *IEEE Signal Processing Magazine* 16.6 (1999), pp. 22–38. DOI: 10.1109/79.799930.

[177] M. Unser, A. Aldroubi, and M. Eden. “B-spline signal processing. I. Theory”. In: *Signal Processing, IEEE Transactions on* 41.2 (1993), pp. 821–833. DOI: 10.1109/78.193220.

[178] A. Vahidi, A. Stefanopoulou, and H. Peng. “Recursive least squares with forgetting for online estimation of vehicle mass and road grade: Theory and experiments”. In: *Vehicle System Dynamics - VEH SYST DYN* 43 (2005), pp. 31–55. DOI: 10.1080/00423110412331290446.

[179] S. Van Vaerenbergh, M. Lázaro-Gredilla, and I. Santamaria. “Kernel Recursive Least-Squares Tracker for Time-Varying Regression”. In: *Neural Networks and Learning Systems, IEEE Transactions on* 23 (2012), pp. 1313–1326. DOI: 10.1109/TNNLS.2012.2200500.

[180] S. Van Vaerenbergh, I. Santamaria, W. Liu, and J. C. Principe. “Fixed-Budget Kernel Recursive Least-Squares”. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (2010).

[181] S. Van Vaerenbergh and I. Santamaría. “A Comparative Study of Kernel Adaptive Filtering Algorithms”. In: *2013 IEEE Digital Signal Processing (DSP) Workshop and IEEE Signal Processing Education (SPE)*. Software available at https://github.com/steven2358/kafbox/. 2013. DOI: 10.1109/DSP-SPE.2013.6642587.

[182] A. Vdovin, S. Bonitz, C. Landström, and L. Löfdahl. “Investigation of Wheel Ventilation-Drag using a Modular Wheel Design Concept”. In: *SAE International Journal of Passenger Cars - Mechanical Systems* 6 (2013). DOI: 10.4271/2013-01-0953.

[183] H.-G. Wahl. “Optimale Regelung eines prädiktiven Energiemanagements von Hybridfahrzeugen”. PhD thesis. KIT Scientific Publishing, Karlsruhe, 2015. DOI: 10.5445/KSP/1000048347.

[184] W. Wang and J. Zhao. “Inverse updating and downdating for weighted linear least squares using M-invariant reflections”. In: *Linear Algebra and its Applications* 291.1 (1999), pp. 185–199. DOI: 10.1016/S0024-3795(99)00007-5.

[185] H. Waschl, I. Kolmanovsky, M. Steinbuch, and L. del Re. *Optimization and Optimal Control in Automotive Systems*. Springer, Cham, 2014. DOI: 10.1007/978-3-319-05371-4.

[186] A. Weißmann, D. Görges, and X. Lin. “Energy-optimal adaptive cruise control combining model predictive control and dynamic programming”. In: *Control Engineering Practice* 72 (2018), pp. 125–137. DOI: 10.1016/j.conengprac.2017.12.001.

[187] H. Winner, S. Hakuli, F. Lotz, and C. Singer. *Handbook of Driver Assistance Systems - Basic Information, Components and Systems for Active Safety and Comfort*. Springer, Cham, 2016. DOI: 10.1007/978-3-319-12352-3.

[188] D.-M. Wu et al. “Fast velocity trajectory planning and control algorithm of intelligent 4WD electric vehicle for energy saving using time-based MPC”. In: *IET Intelligent Transport Systems* 13 (1 2019), pp. 153–159. ISSN: 1751-956X.

[189] Z. Wu, J. Li, J. Zuo, and S. Li. “Path Planning of UAVs Based on Collision Probability and Kalman Filter”. In: *IEEE Access* 6 (2018), pp. 34237–34245. DOI: 10.1109/ACCESS.2018.2817648.

[190] L. Xin, P. Hailong, and L. Jianqiang. “Trajectory prediction based on particle filter application in mobile robot system”. In: *2008 27th Chinese Control Conference*. 2008, pp. 389–393. DOI: 10.1109/CHICC.2008.4605472.

[191] N. M. Yatim and N. Buniyamin. “Development of Rao-Blackwellized Particle Filter (RBPF) SLAM Algorithm Using Low Proximity Infrared Sensors”. In: *9th International Conference on Robotic, Vision, Signal Processing and Power Applications*. Ed. by H. Ibrahim, S. Iqbal, S. S. Teoh, and M. T. Mustaffa. Springer Singapore, 2017, pp. 395–405.

[192] Y. Yuan and S. Zhou. “Sequential B-Spline Surface Construction Using Multiresolution Data Clouds”. In: *Journal of Computing and Information Science in Engineering* 12.2 (2012). DOI: 10.1115/1.4006000.

[193] X. Q. Yunbao Huang. “Dynamic B-spline surface reconstruction: Closing the Sensing-and-modeling loop in 3D digitization”. In: *Computer-Aided Design, Elsevier* (2007).

[194] R. Zhang and E. Yao. “Electric vehicles’ energy consumption estimation with real driving condition data”. In: *Transportation Research Part D: Transport and Environment* 41 (2015), pp. 177–187. DOI: 10.1016/j.trd.2015.10.010.

[195] S. Zhang, Y. Luo, K. Li, and V. Li. “Real-Time Energy-Efficient Control for Fully Electric Vehicles Based on an Explicit Model Predictive Control Method”. In: *IEEE Transactions on Vehicular Technology* 67.6 (2018), pp. 4693–4701. DOI: 10.1109/TVT.2018.2806400.

[196] S. Zhang and X. Zhuan. “Study on Adaptive Cruise Control Strategy for Battery Electric Vehicle Considering Weight Adjustment”. In: *Symmetry* 11.12 (2019). DOI: 10.3390/sym11121516.

[197] S. Zhang and R. Xiong. “Adaptive energy management of a plug-in hybrid electric vehicle based on driving pattern recognition and dynamic programming”. In: *Applied Energy* 155 (2015), pp. 68–78. DOI: 10.1016/j.apenergy.2015.06.003.

[198] X. Zhang, L. Xu, J. Li, and M. Ouyang. “Real-Time Estimation of Vehicle Mass and Road Grade Based on Multi-Sensor Data Fusion”. In: *2013 IEEE Vehicle Power and Propulsion Conference (VPPC)*. 2013, pp. 1–7.

[199] Y. Zhang et al. “Estimation of Electric Mining Haul Trucks’ Mass and Road Slope Using Dual Level Reinforcement Estimator”. In: *IEEE Transactions on Vehicular Technology* 68.11 (2019), 10627–10638.

[200] D. Zhao and H. Guo. “A Trajectory Planning Method for Polishing Optical Elements Based on a Non-Uniform Rational B-Spline Curve”. In: *Applied Sciences* 8.8 (2018). DOI: 10.3390/app8081355.

[201] K. Zhao, F. Ling, H. Lev-Ari, and J. G. Proakis. “Sliding window order-recursive least-squares algorithms”. In: *IEEE Transactions on Signal Processing* 42.8 (1994), pp. 1961–1972. DOI: 10.1109/78.301835.

[202] X. Zhao et al. “Model Selection for Parametric Surfaces Approximating 3D Point Clouds for Deformation Analysis”. In: *Remote Sensing* 10.4 (2018). DOI: 10.3390/rs10040634.

[203] Z.-S. Zhao et al. “Improved Rao-Blackwellized Particle Filter by Particle Swarm Optimization”. In: *Journal of Applied Mathematics* (2013).

[204] D. Zhou et al. “Online Energy Management Strategy of Fuel Cell Hybrid Electric Vehicles: A Fractional-Order Extremum Seeking Method”. In: *IEEE Transactions on Industrial Electronics* 65.8 (2018), pp. 6787–6799.

[205] F. Zhou, W.-J. He, and X.-Y. Fan. “Marginalized Particle Filter for Maneuvering Target Tracking Application”. In: *Advances in Grid and Pervasive Computing*. Ed. by P. Bellavista et al. Springer Berlin Heidelberg, 2010, pp. 542–551.

[207] M. Zimmer. *Durchgängiger Simulationsprozess zur Effizienzsteigerung und Reifegraderhöhung von Konzeptbewertungen in der Frühen Phase der Produktentstehung*. Springer Wiesbaden, 2015, pp. 115–130. DOI: 10.1007/978-3-658-11500-5.

Publications

Peer-reviewed journal papers

[74] J. Jauch*, J. Masino*, T. Staiger, and F. Gauterin. "Road Grade Estimation With Vehicle-Based Inertial Measurement Unit and Orientation Filter". In: *IEEE Sensors Journal* 18.2 (2018). * Equal contribution, pp. 781–789. DOI: 10.1109/JSEN.2017.2772305.

[78] J. Jauch, F. Bleimund, M. Frey, and F. Gauterin. "An Iterative Method Based on the Marginalized Particle Filter for Nonlinear B-Spline Data Approximation and Trajectory Optimization". In: *Mathematics* 7.4 (2019). DOI: 10.3390/math7040355.

[79] J. Jauch, F. Bleimund, M. Frey, and F. Gauterin. "Automated Longitudinal Control Based on Nonlinear Recursive B-Spline Approximation for Battery Electric Vehicles". In: *World Electric Vehicle Journal* 10.3 (2019). DOI: 10.3390/wevj10030052.

[80] J. Jauch, F. Bleimund, S. Rhode, and F. Gauterin. "Recursive B-spline approximation using the Kalman filter". In: *Engineering Science and Technology, an International Journal* 20.1 (2017), pp. 28–34. DOI: 10.1016/j.jestch.2016.09.015.

Source code

[75] J. Jauch. *ALC Matlab files*. 2019. DOI: 10.5281/zenodo.3334806. URL: http://github.com/JensJauch/ALC/releases.

[76] J. Jauch. *NRBA Matlab files*. 2019. DOI: 10.5281/zenodo.2604318. URL: http://github.com/JensJauch/nrba.

[77] J. Jauch. *RBA source code*. 2016. DOI: 10.5281/zenodo.157732. URL: http://github.com/JensJauch/RBA.

Technical reports

[41] D. Dörr, B. Fath, J. Jauch, and M. Frey. *Verbundprojekt: "Gesamtfahrzeugintegration innovativer Konzepte für ein effizientes und performantes E-Fahrzeug (e-volution)", Teil vorhaben KIT: "Assistenzsysteme für effiziente und performante Elektrofahrzeuge" : Schlussbericht*. Tech. rep. Karlsruher Institut für Technologie (KIT), 2017.

Supervised student theses

[29] F. Chang. “Entwicklung eines Algorithmus zur Schätzung des Radbremsmomentes in Abhängigkeit des Bremsdrucks”. Master thesis. Karlsruher Institut für Technologie, Institut für Fahrzeugsystemtechnik, 2016.

[32] Z. Chen. “Co-simulation of a real-time capabale vehicle model with a microscopic traffic-flow simulation”. Master thesis. Karlsruhe Institute of Technology, Institute of Vehicle System Technology, 2018.

[49] J. Fischer. “Methodische Entwicklung eines aktiven Aerodynamiksystems an einer Fahrzeugfront”. Master thesis. Karlsruher Institut für Technologie, Institut für Fahrzeugsystemtechnik, 2016.

[85] Y. Jin. “Entwicklung eines autonomen Fahrspurwechsels für LKW mit Einzelradbremsungsregelung”. Master thesis. Karlsruher Institut für Technologie, Institut für Fahrzeugsystemtechnik, 2018.

[98] C. Lee. “Adaptive Cruise Control mit einer modellprädiktiven Regelung”. Master thesis. Karlsruher Institut für Technologie, Institut für Fahrzeugsystemtechnik, 2016.

[167] T. Staiger. “Vehicle Based Road Grade Estimation”. Bachelor thesis. Karlsruhe Institute of Technology, Institute of Vehicle System Technology, 2017.

[206] S. Zhou. “Entwicklung einer modellprädiktiven Regelung zur optimalen Betriebsführung eines hybriden Batteriespeichers”. Master thesis. Karlsruher Institut für Technologie, Institut für Fahrzeugsystemtechnik, 2018.