Model-based control for automotive applications

Model-based control for automotive applications

Gerrit Naus

The research reported in this thesis is part of the research program of the Dutch Institute of Systems and Control (DISC). The author has successfully completed the educational program of the Graduate School DISC.

The research reported in this thesis is supported by TNO Automotive, Helmond, the Netherlands and DAF Trucks N.V., Eindhoven, the Netherlands.

G.J.L. Naus (2010). *Model-based control for automotive applications*. Ph.D. thesis, Eindhoven University of Technology, Eindhoven, the Netherlands.

A catalogue record is available from the Eindhoven University of Technology library.
ISBN: 978-90-386-2353-5

Cover design: I.M. van de Voort and G.J.L. Naus.
Reproduction: Ipskamp Drukkers B.V., Enschede, the Netherlands.

Model-based control for automotive applications

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op woensdag 3 november 2010 om 16.00 uur

door

Gerrit Jacobus Lambertus Naus

geboren te Roermond

Dit proefschrift is goedgekeurd door de promotor:

prof.dr.ir. M. Steinbuch

Copromotor:
dr.ir. M.J.G. van de Molengraft

Summary

Model-based control for automotive applications

The number of distributed control systems in modern vehicles has increased exponentially over the past decades. Today's performance improvements and innovations in the automotive industry are often resolved using embedded control systems. As a result, a modern vehicle can be regarded as a complex mechatronic system. However, control design for such systems, in practice, often comes down to time-consuming online tuning and calibration techniques, rather than a more systematic, model-based control design approach.

The main goal of this thesis is to contribute to a corresponding paradigm shift, targeting the use of systematic, model-based control design approaches in practice. This implies the use of control-oriented modeling and the specification of corresponding performance requirements as a basis for the actual controller synthesis. Adopting a systematic, model-based control design approach, as opposed to pragmatic, online tuning and calibration techniques, is a prerequisite for the application of state-of-the-art controller synthesis methods. These methods enable to achieve guarantees regarding robustness, performance, stability, and optimality of the synthesized controller. Furthermore, from a practical point-of-view, it forms a basis for the reduction of tuning and calibration effort via automated controller synthesis, and fulfilling increasingly stringent performance demands.

To demonstrate these opportunities, case studies are defined and executed. In all cases, actual implementation is pursued using test vehicles and a hardware-in-the-loop setup.

- Case I: Judder-induced oscillations in the driveline are resolved using a robustly stable drive-off controller. The controller prevents the need for re-tuning if the dynamics of the system change due to wear. A hardware-in-the-loop setup, including actual sensor and actuator dynamics, is used for experimental validation.
- Case II: A solution for variations in the closed-loop behavior of cruise control func-

tionality is proposed, explicitly taking into account large variations in both the gear ratio and the vehicle loading of heavy duty vehicles. Experimental validation is done on a heavy duty vehicle, a DAF XF105 with and without a fully loaded trailer.

- Case III: A systematic approach for the design of an adaptive cruise control is proposed. The resulting parameterized design enables intuitive tuning directly related to comfort and safety of the driving behavior and significantly reduces tuning effort. The design is validated on an Audi S8, performing on-the-road experiments.
- Case IV: The design of a cooperative adaptive cruise control is presented, focusing on the feasibility of implementation. Correspondingly, a necessary and sufficient condition for string stability is derived. The design is experimentally tested using two Citroën C4's, improving traffic throughput with respect to standard adaptive cruise control functionality, while guaranteeing string stability of the traffic flow.

The case studies consider representative automotive control problems, in the sense that typical challenges are addressed, being variable operating conditions and global performance qualifiers. Based on the case studies, a generic classification of automotive control problems is derived, distinguishing problems at i) a full-vehicle level, ii) an in-vehicle level, and iii) a component level. The classification facilitates a characterization of automotive control problems on the basis of the required modeling and the specification of corresponding performance requirements.

Full-vehicle level functionality focuses on the specification of desired vehicle behavior for the vehicle as a whole. Typically, the required modeling is limited, whereas the translation of global performance qualifiers into control-oriented performance requirements can be difficult. In-vehicle level functionality focuses on actual control of the (complex) vehicle dynamics. The modeling and the specification of performance requirements are typically influenced by a wide variety of operating conditions.

Furthermore, the case studies represent practical application examples that are specifically suitable to apply a specific set of state-of-the-art controller synthesis methods, being robust control, model predictive control, and gain scheduling or linear parameter varying control. The case studies show the applicability of these methods in practice. Nevertheless, the theoretical complexity of the methods typically translates into a high computational burden, while insight in the resulting controller decreases, complicating, for example, (online) fine-tuning of the controller. Accordingly, more efficient algorithms and dedicated tools are required to improve practical implementation of controller synthesis methods.

Contents

Nomenclature

Acronyms and abbreviations

acronym	description
ABS	anti-lock braking system
ADAS	advanced driver assistance system
ACC	adaptive cruise control
AMT	automated manual transmission
CACC	cooperative adaptive cruise control
CAN	controller area network
CC	cruise control
DAF	DAF Trucks N.V.
DK	DK iteration procedure
EBS	electronic braking system
ECU	electronic control unit
EHB	electro-hydraulic braking system
FTS	first tier supplier
GIS	geographical information system
GPS	global positioning system
GS	gain scheduling
HDV	heavy duty vehicle
HIL	hardware in the loop
HSV	Hankel singular value
IIO	incremental input-output
IP	intellectual property
IVS	integrated vehicle control
KYP	Kalman Yakubovich Popov
LFT	linear fractional transformation
LMI	linear matrix inequality
LP	linear program
LPV	linear parameter varying control

acronym	description
LS	loop shaping
MABX	Micro AutoBox
MACS	modular automotive control system
MIMO	multi input, multi output
MPC	model predictive control
mpQP	multi-parametric quadratic program
MPT	multi parametric toolbox
ODE	ordinary differential equation
OEM	original equipment manufacturer
P(I)D	proportional, (integral), and differential control
PSD	power spectral density
PWA	piecewise affine
QP	quadratic program
RAM	random access memory
RCP	rapid control prototyping
RGA	relative gain array
RTK	real-time kinematic
SG	stop-&-go
SISO	single input, single output
SQL	sequential loop closing
SVD	singular value decomposition
TNO	Netherlands organization for applied scientific research
UDP	user datagram protocol
VEHIL	vehicle hardware in the loop

Roman symbols and letters

symbol	description	value	unit
$\mathbf{A}$	state-space system matrix		
A_f	frontal area		m^2
a	acceleration		$\mathrm{m/s}^2$
$\mathbf{B}$	state-space input matrix		
$\mathbf{C}$	state-space output matrix		
C_{F_κ}	longitudinal slip stiffness		N
C_w	air drag coefficient		-
$\mathcal{C}$	constraint		
c_{rl}	rolling resistance constant		-
$\mathbf{D}$	state-space feedthrough matrix		
$D(s)$	transfer function delay model		
$D_x(s)$	scaling filter, $x \in \{p, s\}$		
d	rotational damping		N m s/rad
	translational damping		N s/m
$E(s)$	Laplace transform of $e(t)$		
e	error		
F	force		N
$F(s)$	transfer function feedforward filter		
F_x	longitudinal driving force		N
$\mathcal{F}_\infty$	largest positively invariant subset		
f	frequency		rad/s
	fuel		mg / stroke
$f(x)$	x-dependent variable		
G	gear		-
$G(s)$	transfer function system model		
$\mathcal{G}(s)$	string-stability transfer function		
g	gravitational constant	9.81	$\mathrm{m/s}^2$
$g(t)$	impulse response function $\mathcal{L}^{-1}(\mathcal{G}(s))$		
$g, \boldsymbol{g}(x)$	x-dependent variable, vector		
$H(s)$	transfer function system model		
	transfer function spacing policy dynamics		
$\mathcal{H}$	system model		
$\mathcal{H}_\infty$	$\mathcal{H}_\infty$ controller synthesis		
h	headway time		s
i	transmission ratio		-
	index number		
J	inertia		$\mathrm{kg\,m^2/rad}$
	optimization criterion		
j	jerk		$\mathrm{m/s}^3$
	imaginary number		

symbol	description	value	unit
$\mathbf{K}$	controller matrix		
$K(s)$	transfer function controller model		
$\mathcal{K}$	controller model		
k	rotational stiffness		N m/rad
	translational stiffness		N/m
	discrete time steps		-
	index number		
	gain		-
k_{MM}	modulus margin		dB
$L(s)$	open-loop transfer function		
l	vehicle length		m
$M(s)$	transfer function system model		
$\mathcal{M}$	system model		
m	mass		kg
$N(s)$	transfer function system model		
n, N	number		-
$\mathcal{O}$	control objective		
P	design parameter	$[0, 1]$	-
$\mathbf{Q}, Q$	(matrix) weighting		
R	weighting		
$\mathcal{R}_i$	region i		
$\mathcal{R}_o$	operating point		
r	radius		m
	reference		
$S(s)$	sensitivity function		
s	Laplace operator		
s_i	scaling factor		
T	torque		N m
$T(s)$	complementary sensitivity function		
T_s	sample time		s
$\mathcal{T}$	tuple of tuning parameters		
t	continuous time		s
$U(s)$	Laplace transform of $u(t)$		
$\boldsymbol{u}, u$	input (vector)		
v	velocity		m/s
v_{CC}	desired cruise control velocity		m/s
$W(s)$	weighting filter		
$\boldsymbol{w}_p, w_p$	exogenous input (vector)		
$X(s)$	Laplace transform of $x(t)$		
$\mathcal{X}_f$	feasible set		
$\boldsymbol{x}, x$	model state (vector)		
x	position		m

symbol	description	value	unit
$x_{r,0}$	distance at standstill		m
$\boldsymbol{y}, y$	output (vector)		
$\boldsymbol{z}_p, z_p$	exogenous output (vector)		

Greek symbols and letters

symbol	description	value	unit
α	road inclination		rad
α_{PM}	phase margin		rad
γ	$\mathcal{H}_\infty$ performance indicator		
$\boldsymbol{\Delta}$	uncertainty matrix		
Δ	operating range		
δ	perturbation		
	real-valued uncertainty		
ϵ	estimation error		-
η	fraction		-
	efficiency		-
θ	rotation		rad
κ	longitudinal slip		-
$\Lambda(s)$	transfer function model		
μ	friction coefficient		-
$\mu_{p,s}$	robust performance, stability		
$\Xi(s)$	transfer function model		
$\boldsymbol{\xi}$	model state (vector)		
ξ	parametric uncertainty		
ρ	air density	1.29	km/m^3
$\overline{\sigma}$	maximum singular value		
τ	delay time		s
	breakpoint		s/rad
ϕ	delay		s
$\boldsymbol{\chi}$	parameter vector		
ω	frequency		rad/s
	rotational velocity		rad/s
ω_n	measurement noise frequency bound		rad/s

Subscripts and indices

symbol	description	symbol	description
$_0$	constant	$_n$	nominal value
$_{(i,j)}$	element (i, j)	$_{nl}$	nonlinear
$_a$	augmented	$_O$	operating range

symbol	description
	air
br	brake
bw	bandwidth
c	convex
	comfort
	component
cl	clutch
D	differential
d	desired
	driveline
e	engine
	extended
eq	equivalent
f	final drive
g	gear box
h	high-frequent
	host vehicle
I	interpolation
K	controller
kin	kinetic
l	linear
m	mean
	mass
max	maximum
min	minimum
n	normal direction

symbol	description
o	operating point
	overshoot
P	proportional
p	performance
	powertrain
	perturbed
ps	propulsion shaft
r	resistance
	reduced
	relative
rl	road load
rr	radar range
rt	real target
s	drive shaft
	settling
	safety
	system
sl	slip
st	static
t	tachograph
	target vehicle
th	throttle
v	vehicle
	validation
vt	virtual target
w	wheel

Operations and notation

symbol	description
$\mathbf{0}$	zero matrix
$\mathcal{F}(\cdot)$	lower fractional transform
$\mathbf{I}$	identity matrix
$\mathcal{L}(\cdot)$	Laplace transform
$\mathbb{N}$	set of non-negative integers
$\mathbb{R}^{n_1 \times n_2}$	set of real matrices of dimension $n_1 \times n_2$
$\boldsymbol{x}, (\cdot)$	vector
$\mathbf{A}, (\cdot)$	matrix
$\boldsymbol{x}^T, \mathbf{A}^T$	vector or matrix transpose
$\dot{x}, (\ddot{x})$	(double) time derivative
$\tilde{\boldsymbol{x}}$	perturbation around $\overline{\boldsymbol{x}}$

symbol	description
$\overline{\boldsymbol{x}}$	nominal operating point
	maximum value
$\hat{x}$	estimate of x
$[\cdot]$	closed, continuous set
$\{\cdot\}$	set of discrete values
$\angle$	phase angle
$\lvert\cdot\rvert$	absolute value
	complex modulus
$\lvert\lvert\cdot\rvert\rvert_1$	1-norm over time
$\lvert\lvert\cdot\rvert\rvert_\infty$	maximum magnitude
$\forall$	for all values

CHAPTER 1

Introduction

Abstract - In this chapter, an introduction to the development of embedded automotive control functionality is given. The challenges in today's automotive control design process are discussed, motivating the research objectives and the case studies that are considered in this thesis. Based on the objectives, a research approach is presented and an outline of the thesis is given.

1.1 Embedded control functionality in the automotive industry

1.1.1 History and current research directions

The application of electronics and (control) software in the automotive industry has been increasing exponentially over the past decades. Traditionally, the automotive industry focused on mechanical, hydraulic, and, in the case of heavy-duty vehicles (HDVs), also pneumatic solutions. Driven by the development and opportunities of electronics and (control) software, today's market demands for new functionality in both passenger cars and HDVs are often resolved with embedded systems. Embedded systems are processors incorporating dedicated software functionality, which are embedded as part of a larger hardware system. The hardware incorporates electronics and mechanical parts, including actuators and sensors. In the automotive industry, these processors are called electronic control units (ECUs) (Larses, 2005).

Initially induced by environmental issues and thereafter strongly driven by safety and comfort demands, application of embedded systems in the automotive industry has expanded enormously since the late 1970s. Legislation required a decrease of emissions and fuel consumption, which resulted in the development of catalytic converters and so-called electronic diesel control to control the ignition of diesel engines more accurately. The application of electronic diesel control in commercially available vehicles initiated a paradigm shift in the automotive industry. The time line in Figure 1.1 illustrates this shift from the use of mechanical and hydraulic solutions to the use of embedded electronics and control software in the development of new functionality.

Embedded systems often offer less weight, allow more compact and flexible packaging, and, most importantly, software allows adding functionality to existing hardware, enabling more functionality than mechanical systems. As a result, new functionality can be implemented more quickly and easily. Hence, embedded control functionality enables OEMs to differentiate between their vehicle models in a cost effective manner (Pretschner et al., 2007; Ward and Fields, 2000; Heinecke et al., 2004). The number of systems and functionalities in a vehicle that rely on embedded control software has increased exponentially in the past decades. Modern vehicles may contain over 70 separate ECUs to handle all embedded electronics and software functionality, moreover, it is estimated that currently more than 80 percent of all automotive innovation stems from electronics and software functionality (Mössinger, 2010; Leen and Heffernan, 2002). This trend is expected to continue for several more decades, whereas before 1978 a vehicle contained only mechanical and hydraulic parts (Broy, 2006).

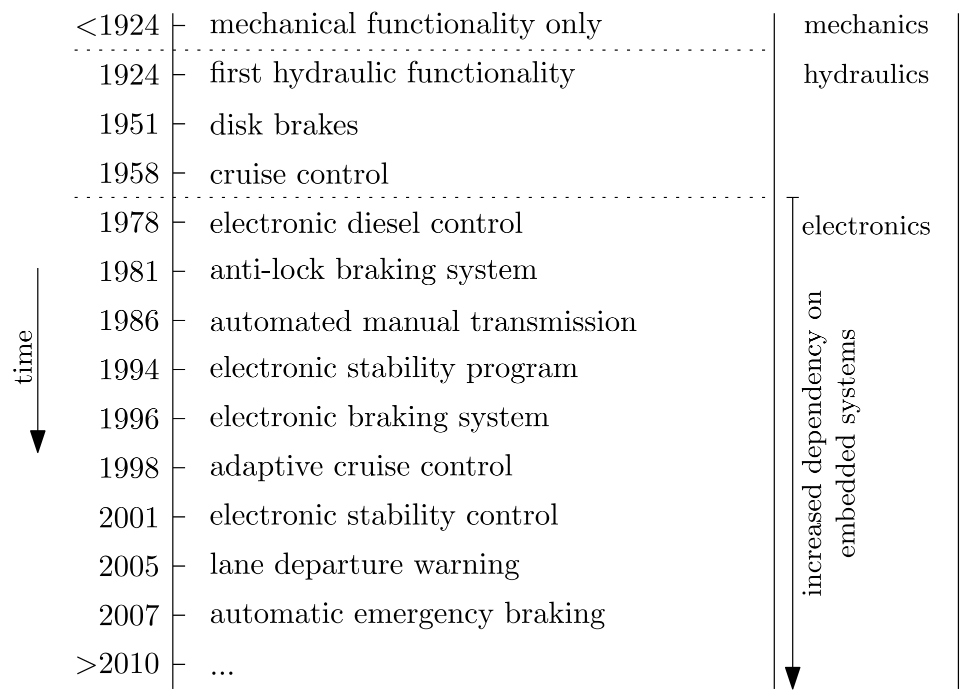


Figure 1.1: Time line with some of the key developments in commercially available functionality in the automotive industry (Leen and Heffernan, 2002; Larses, 2005; Rijkswaterstaat, 2007; Broy, 2006; WABCO, 2010).

In Figure 1.2, some examples of standard systems and functionalities that are present in today's commercially available passenger cars and heavy-duty vehicles are shown. The systems and functionalities are classified into six generic domains, namely the powertrain, the driveline, the chassis, the body, infotainment systems, and so-called advanced driver assistance systems (ADASs) (Navet et al., 2005). The main developments in the automotive industry can be related to these domains. They are driven by increasingly stringent performance demands in the fields of safety, environment, mobility, driver comfort, and costs (Mössinger, 2010; Guzzella, 2009).

Innovations in infotainment and body control systems are primarily driven by driver comfort and have resulted in an exponential growth in in-vehicle electronics (Bosch, 2007; Mössinger, 2010). Examples are the 'electrification' of door locks, windows, mirrors, and seat adjustment, as well as climate control, automatic wipers, and automatic headlight control. A major trend is the integration of consumer electronics and entertainment systems into vehicles (Cassius and Kun, 2007). So-called telematic and infotainment systems combine audio, video, wireless connectivity, navigation, global positioning, and up-to-date route information.

In the chassis domain, especially the development of active safety systems has received much attention in the past decades (see, e.g., the time line in Figure 1.1). Various active safety functions in the chassis domain are standard in today's commercially available vehicles, such as the anti-lock braking system, electronic stability control, and, in the body domain, airbags (OICA, 2006). Still, the number of fatalities and injuries world-

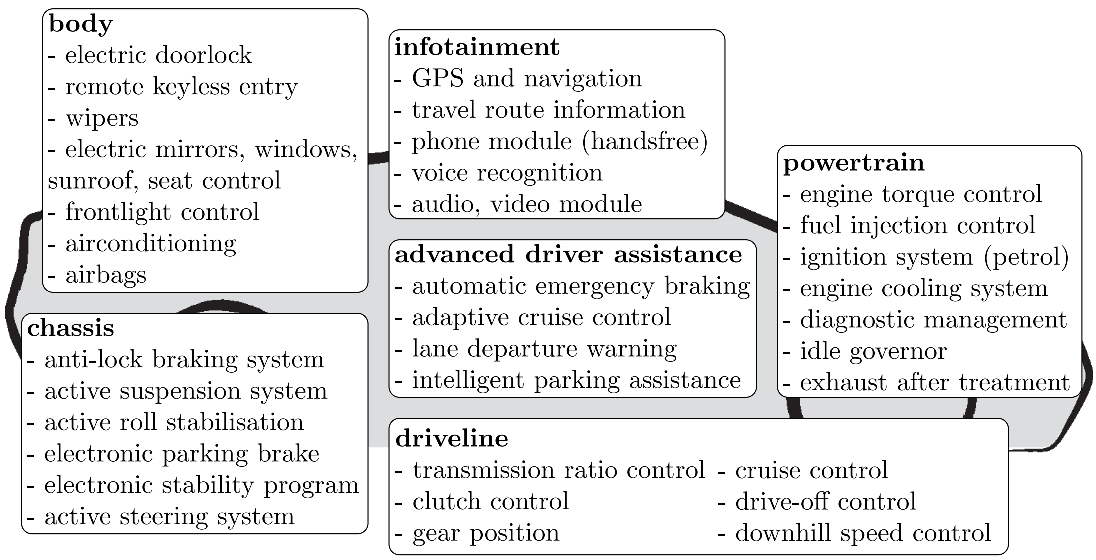


Figure 1.2: Examples of electronics and software functionality that are standard in a modern passenger car or heavy-duty vehicle (Mössinger, 2010; Guzzella, 2009; Bosch, 2007).

wide is too large, and the corresponding costs are enormous, driving an ongoing development of active safety systems (World Health Organization, 2009). Current research focuses on the integration and combination of chassis control systems, such as the anti-lock braking system, electronic stability control, traction control, roll-stability control, and yaw moment control, using active front and rear wheel driving and steering, active final-drive control, and (semi-)active suspension systems (Yu et al., 2008; Chang and Gordon, 2008).

A new generation of active safety systems is based on advanced driver assistance systems (ADASs) (Lu et al., 2005). Current driver assistance systems are primarily intended as comfort systems, relieving a driver's work load. Examples are present in different domains of a modern vehicle, such as cruise control, electronic power steering, automated transmissions, and route planning and navigation. The use of situational awareness in ADAS functionality facilitates an increased focus on active safety instead of comfort (Guzzella, 2009; Nagai, 2007). This is illustrated by the introduction of and research into new active safety functionalities, such as automatic emergency braking or collision mitigation systems, active seatbelt control, and collision avoidance systems (Lu et al., 2005; Laan, 2009). Other examples of advanced driver assistance systems are adaptive cruise control, lane departure warning, lane keeping, and intelligent parking assist systems.

Ongoing innovations in the powertrain and driveline domain are driven by energy efficiency, performance optimization and reducing emissions (Guzzella, 2009; Cook et al., 2006; Sun et al., 2005). International directives on NO_x, HC, soot, and CO_2 are increasingly stringent (Buckland and Cook, 2005). Furthermore, societal demands on efficiency, fuel economy, and performance are continuously increasing. Examples of extensive research areas are turbo charging, after treatment systems, exhaust gas recirculation, valve timing control, throttle control, fuel pressure control, and optimization of the transmission, for example, the continuously variable transmission (Meulen et al., 2009). Furthermore, a ceaseless search for 'clean' alternative energy sources, such as electricity, hydrogen, bio fuels, alcohol-based fuels, and fuel cells, is ongoing (Guzzella, 2009; Chan, 2007). In particular the hybridization and electrification of the powertrain, combining a traditional internal combustion engine and an electric motor-generator, is an active field of research (see, e.g., Hofman et al., 2007; Keulen et al., 2009a). Research activities focus on optimal energy management or powersplit control, regenerative braking, auxiliary control, engine downsizing, start-stop control, and route-based optimization combining geographic information system, global positioning system and route planning.

Innovations in active safety systems, driver assistance systems, and advanced driver assistance systems are enabled by by-wire technology and innovative sensor technology (Larses, 2005). By-wire technology extends or replaces part of originally mechanical functionality by embedded systems, facilitating control of a single system by multiple functionalities. Examples are brake-by-wire, shift-by-wire, and steer-by-wire functionality. In so-called full by-wire systems, the total system is controlled electronically, including the power transfer using electro-mechanical or electro-hydraulic actuators. These actuators replace originally direct mechanical or hydraulic links. The application of by-wire systems has become standard in modern vehicles, although application of full by-wire systems is, until now, often restricted by legislation.

Simultaneous to the increase in by-wire technology, the number of in-vehicle sensors has increased exponentially (Ahmed et al., 2007; Broy, 2006). Both innovative sensor technology and developments in the field of estimators and observers enable increased situational awareness, extensive vehicle state estimates, and driver monitoring (Kolmanovsky and Winstead, 2006). Especially wireless communication is regarded as a future major step to improve safety and, in particular, mobility. The demand for individual mobility will only increase, while traffic jams are a major burden already. It is estimated that the traffic problem in the Netherlands currently costs 3 billion euro per year (Netherlands Institute for Transport Policy Analysis - KIM, 2008). For example, research into co-operative adaptive cruise control and platooning indicates the possibilities for increased traffic throughput and improved traffic flows when vehicle-to-vehicle and infrastructure-to-vehicle communication is employed (Arem et al., 2006).

1.1.2 Control architecture

As indicated in the previous section, the number of embedded control functions in the automotive industry has increased exponentially in the past decades. In a modern vehicle, over 250 distinct software functions are present. Today, this functionality is distributed over multiple ECUs throughout the vehicle, whereas formerly, the ECUs represented stand-alone functional units. To increase performance, decrease vehicle weight and increase reliability, data bus systems have replaced direct wiring (Pretschner et al., 2007; Richter and Ernst, 2006). The data bus systems form in-vehicle networks, interconnecting all ECUs, actuators and sensors. Still, the wiring harness of a modern passenger car may have up to 4.000 parts, weigh as much as 40 kg and contain more than 1900 wires for up to 4 km of wiring (Navet, 2009). As a result, complexity of both the design and the integration of the functionality has increased significantly.

To master the complexity of today's in-vehicle networks, up to 5 different bus systems are present in a modern vehicle. Each bus system is specialized for a specific domain of the vehicle, such as the powertrain, the chassis, the body, and the driveline (Nolte et al., 2005; Stroop and Stolpe, 2006; Navet et al., 2005). Developed in the 1980s, the controller area network (CAN) is the most widespread in-vehicle network (Kiencke et al., 1986; Leen and Heffernan, 2002). CAN, and other standardized in-vehicle networks accommodate event-triggered communication. Driven by the increasing demand for more complex, dependable and safety-critical functionality, research currently focuses on the development of time-triggered communication protocols, such as FlexRay and the time-triggered

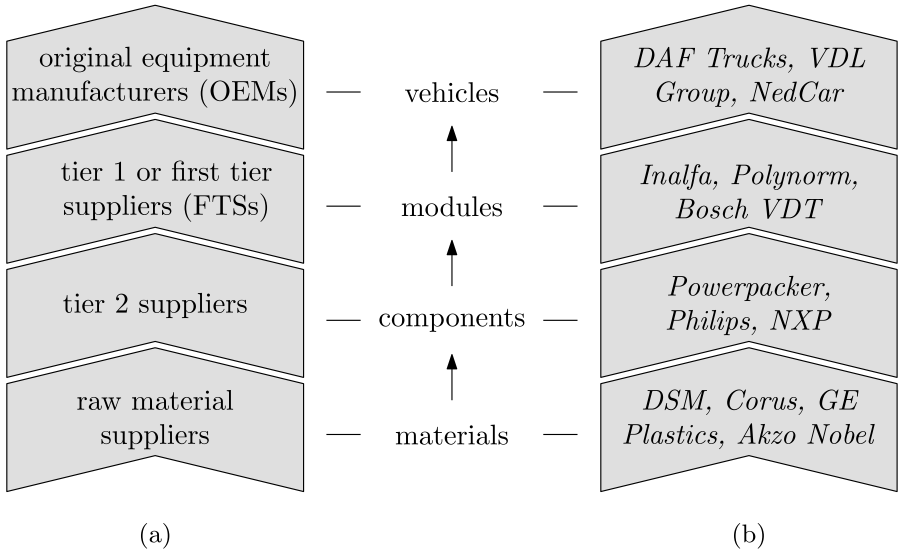


Figure 1.3: (a) Schematic representation of the automotive supply chain. (b) The automotive supply chain in the Netherlands (Wismans, 2007).

CAN protocol (Kandasamy et al., 2005; Flexray, 2002). The large number of systems and functionalities in combination with both the distributed character and the complexity of today's systems and functionalities, make a modern vehicle a complex mechatronic systems (Mössinger, 2010; Leen and Heffernan, 2002).

To handle this complexity, development of innovative functionality and new systems is done by specialized suppliers, so-called first tier suppliers (FTSs). The vehicle producers, i.e., the original equipment manufacturers (OEMs), typically focus on specific core competencies, being the specification and integration of all systems and functionalities, the development of the engine, the styling of the vehicle, and the marketing of the vehicle (Pretschner et al., 2007; Fröberg et al., 2007).

Formerly, OEMs were responsible for the development of the main systems in a vehicle and relatively little development was solely done by suppliers. Starting in the 1980s, tier 1, or FTSs became responsible for larger systems, designed in co-development with the OEMs. The make-and-deliver-to-order trend of the 1990s, in combination with the increasing complexity of the systems made the OEMs focus on core competencies even more. As a result, today, the development of the main systems and functionalities is done by specialized FTSs, while the OEMs are challenged with the specification and the integration of all functionalities and systems (Ward and Fields, 2000; Richter and Ernst, 2006). This has resulted in a highly vertical supply chain, which is shown schematically in Figure 1.3(a). As an example, the automotive supply chain in the Netherlands is shown in Figure 1.3(b) (Wismans, 2007).

As a result of the vertical supply chain, the amount of proprietary technology in a vehicle is large. This holds in particular for embedded (control) software functionality. FTSs often supply more or less black-box systems to the OEMs. Hence, it is difficult if not impossible for the OEMs to localize errors or modify parts of a system, while it is difficult for the FTSs to design optimal systems. This effect is illustrated by the V-cycle development process (Das V-Modell, 2006). The V-cycle is commonly used in the automotive industry to represent development processes, (see, e.g., Huisman and Veldpaus, 2005; Gietelink, 2007).

In Figure 1.4, a V-cycle for the design of an automotive control system is shown. The name of the V-cycle development process is related to the steps in the design flow, which can be ordered to form the shape of the letter V. As a result, different levels of abstraction can be distinguished, coupling a specification or design step (steps 1 to 3 in Figure 1.4) and a corresponding validation or verification step (steps 5 to 7 in Figure 1.4). Per level, different validation and verification tools are available, such as test drives and hardware-in-the-loop (HIL) tests (see Figure 1.4). HIL tests involve verification of the embedded implementation of a system using a model of the rest of the vehicle and its environment (Kluge et al., 2009; Schuette and Waeltermann, 2005). Only at the level of the functional validation, the functionality is built-in into a vehicle and test drives are performed.

The effect of the vertical supply chain can be recognized in the transition between the development steps for which the OEMs are responsible and the steps for which the FTSs are responsible (see Figure 1.4). Specialized FTSs are responsible for the actual design of a system (steps 3 to 5 in Figure 1.4), while the OEMs focus on specification and integration of systems and functionality (steps 1 to 2, and 6 to 7 in Figure 1.4). Due to intellectual property (IP) issues, FTSs supply more-or-less black-box systems to the OEMs. For the integration of these systems, insight in the stability, optimality of performance, and

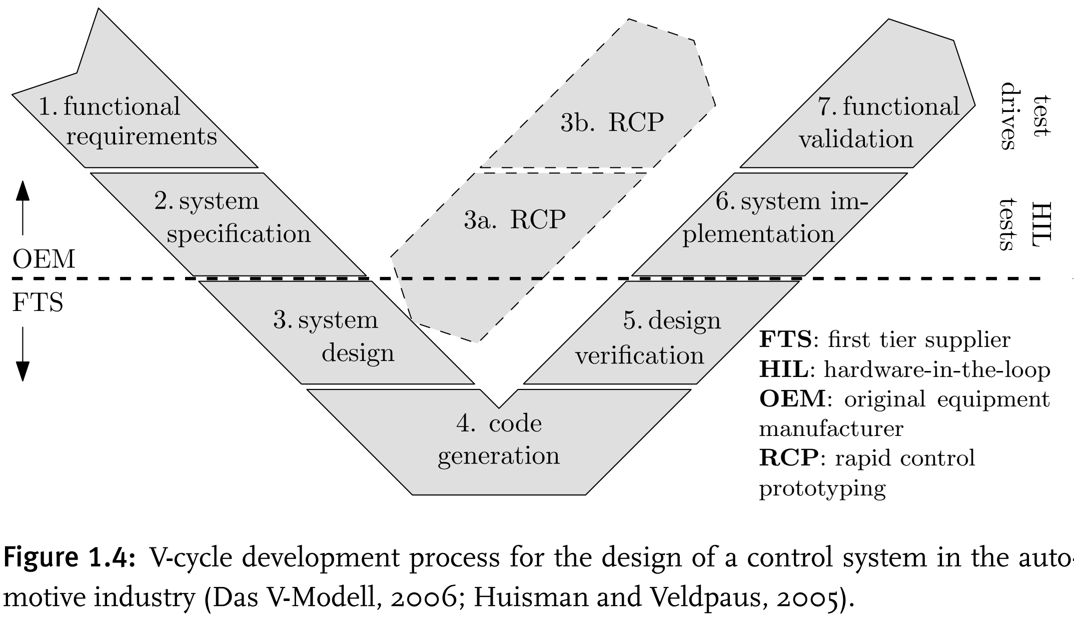


Figure 1.4: V-cycle development process for the design of a control system in the automotive industry (Das V-Modell, 2006; Huisman and Veldpaus, 2005).

robustness of the resulting system are often difficult to assess. As a result, the OEMs employ time-consuming tuning and calibration procedures for the integration of all systems and functionalities. To reduce the integration time, insight in the system designs is crucial.

Testing of a control system design in an early stage, i.e., before the final embedded code is generated, using an HIL setup or performing actual test drives, is called rapid control prototyping (RCP). RCP facilitates improved insight in the design, the achievable performance, and the functionality of a design in an early stage of the development process. Using RCP, the results of HIL tests and test drives can be used directly as feedback in the steps 1, 2 and 3 of the development process. Hence, RCP embodies an optional step in the V-cycle development process, as indicated in Figure 1.4. It enables FTSs to test their systems in an early stage and OEMs to be more involved in the development process, gaining more insight in the systems. RCP was introduced on the automotive market in the mid-1990's. Nowadays, RCP is widely adopted as a solution to handle the increasingly complex control design process in the automotive industry (see, e.g., Schuette and Waeltermann, 2005; Lee et al., 2004).

1.1.3 Control design

Theory vs practice

In theory, control design approaches are well defined. Consider for example a general control configuration as is depicted in Figure 1.5, where $\mathcal{P}$ is a generalized plant model, $\boldsymbol{w}_p(t)$ and $\boldsymbol{z}_p(t)$ are exogenous inputs and outputs, respectively, $\mathcal{K}$ is a controller, $\boldsymbol{u}(t)$ represents the control signals, and $\boldsymbol{y}(t)$ represents the controller input signals (Skogestad and Postlethwaite, 2005). The generalized plant model $\mathcal{P}$ combines a model of the system and the performance requirements, including nonlinearities, time variations, uncertainties, and a model of the disturbances. The exogenous inputs $\boldsymbol{w}_p(t)$ represent user-defined reference signals or commands, disturbances and noise. The exogenous outputs $\boldsymbol{z}_p(t)$ represent the error signals to be minimized, i.e., the performance variables. Hence, the transfer from $\boldsymbol{w}_p(t)$ to $\boldsymbol{z}_p(t)$ is a measure for the performance of the controlled system, indicating to what extent the system behavior matches the desired performance requirements, for example, tracking of a user-defined velocity, or damping of vibrations in the driveline of a vehicle.

Based on this general control configuration, Skogestad and Postlethwaite (2005) propose a general control design approach, listing the essential steps in the development of a control system, see Table 1.1. Following steps 1 to 8, a generalized plant model $\mathcal{P}$ including a performance channel $\boldsymbol{w}_p(t) \mapsto \boldsymbol{z}_p(t)$ is derived. Following steps 9 to 12, the controller $\mathcal{K}$ is synthesized and the resulting closed-loop system is evaluated. Actual implementation and testing of the controller is done in steps 13 and 14. In practice, however, steps 2 to 3,

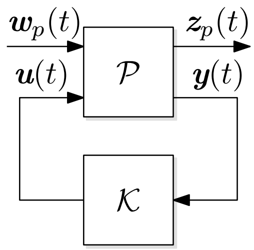


Figure 1.5: General control configuration, where $\mathcal{P}$ is a generalized plant model, $\boldsymbol{w}_p(t)$ and $\boldsymbol{z}_p(t)$ are exogenous inputs and outputs, respectively, $\mathcal{K}$ is a feedback controller, $\boldsymbol{u}(t)$ represent the control signals, and $\boldsymbol{y}(t)$ the controller input signals (Skogestad and Postlethwaite, 2005).

8, and 10 to 12 are often omitted (the shaded steps in Table 1.1) (Skogestad and Postlethwaite, 2005). The remaining steps lack, in essence, the use of a model of the system and the specification of performance requirements that can be used in a systematic model-based control design approach. Following these steps results in an approach that is often referred to as online tuning.

The automotive industry is a typical example where online tuning methods are often adopted (Heinecke et al., 2004; Coelingh et al., 2002; Naus, 2007a,b). Appropriate

Table 1.1: General control design approach (Skogestad and Postlethwaite, 2005).

step	action
1.	Study the system (process, plant) to be controlled and obtain initial information about the control objectives.
2.	Model the system and simplify the model, if necessary.
3.	Scale the variables and analyze the resulting model; determine its properties.
4.	Decide which variables are to be controlled (controlled outputs).
5.	Decide on the measurements and manipulated variables: what sensors and actuators will be used and where will they be placed?
6.	Select the control configuration.
7.	Decide on the type of controller to be used.
8.	Decide on performance requirements, based on the overall control objectives.
9.	Design a controller.
10.	Analyze the resulting controlled system to see if the requirements are satisfied; and if they are not satisfied modify the requirements or the type of controller.
11.	Simulate the resulting controlled system, on either a computer or a pilot plant.
12.	Repeat from step 2, if necessary.
13.	Choose hardware and software and implement the controller.
14.	Test and validate the control system, and tune the controller on-line, if necessary.

Table 1.2: Typical challenges in automotive control problems (Kolmanovsky, 2008; Naus, 2007a,b).

	challenge
1.	variable operating conditions
2.	global performance qualifiers
3.	IP issues resulting from a vertical supply chain

control-oriented models and well-defined performance requirements that would enable a more systematic control design approach are often lacking. As a result, online tuning techniques are adopted to fill (feedforward) lookup tables, and tune (gain-scheduled) PID feedback controllers. Furthermore, logic rules and heuristic control methods are adopted to take into account changing operating conditions (Kolmanovsky, 2008). As a result, OEMs typically employ time-consuming tuning and calibration procedures to integrate all systems and functionality. This tuning and these procedures have to be repeated for every change in the dynamics or in the performance requirements.

From a control point of view, typical disadvantages of online tuning techniques are a lack of guarantees regarding robustness, performance, stability, and optimality. These disadvantages can be overcome by adopting a systematic, model-based control design approach using available controller synthesis methods.

Typical challenges in automotive control problems

The use of pragmatic, online tuning techniques instead of a more systematic control design approach can, at least partly, be related to the complexity that is induced by the vertical supply chain and corresponding IP issues, which are discussed in the previous section. Besides that, typical challenges in automotive control problems complicating the modeling and the specification of performance requirements are induced by variable operating conditions and global performance qualifiers (see Table 1.2) (Kolmanovsky, 2008; Naus, 2007a,b).

The abundance and the variety of operating conditions of a vehicle and the in-vehicle systems are large. As a result, variations in operating conditions form a major challenge in designing automotive control systems. Define the state vector $\boldsymbol{\xi}(t)$ and a vector of real and integer parameters $\boldsymbol{\chi}(t)$, characterizing the system dynamics of $\mathcal{P}$. Assume that stable operating conditions of a system are defined by constant system inputs $\boldsymbol{u}(t) = \overline{\boldsymbol{u}}$, a constant system state $\boldsymbol{\xi}(t) = \overline{\boldsymbol{\xi}}$, constant system dynamics $\boldsymbol{\chi}(t) = \overline{\boldsymbol{\chi}}$, and a constant system output $\boldsymbol{y}(t) = \overline{\boldsymbol{y}}$. The dynamics $\overline{\boldsymbol{\chi}}$ of in-vehicle systems may change, e.g., as a function of temperature variations, loading conditions, wear, the vehicle velocity, the gear ratio, or the engine rotational velocity. Furthermore, a vehicle is a mass-produced product. Small inter-vehicle differences in the systems are inevitable, which results in different dynamics per vehicle. Control systems have to account for these variations.

Variations in the external inputs $\overline{\boldsymbol{u}}$ influence in particular the desired driving behavior. Examples are variable traffic situations, such as normal driving and emergency situations, variable road conditions, such as mountainous regions and flat roads, and changing driver behavior, such as sportive and comfortable driving. If variations in the operating conditions can be measured, they can be taken into account explicitly in the controller design. Otherwise, the variations have to be regarded as uncertainties or unknown disturbances.

Global performance qualifiers can be thought of as general, non-control-oriented performance requirements that are not bound to a specific control problem. Typically, global performance qualifiers are not naturally translated into control-oriented performance requirements that can be used to quantify closed-loop performance. Furthermore, priority or weighting of the qualifiers is typically driver dependent. Examples are safe and comfortable driving, high traffic throughput, fuel economic driving with zero emissions, high vehicle acceleration and deceleration capabilities, and low costs. For specific control problems, global performance qualifiers have to be translated into control-oriented performance requirements and setpoints, where driver-dependent tuning is an important aspect.

The resulting requirements are often conflicting and restricted by legislation or physical limitations. For example, small inter-vehicle distances are favorable for a high traffic throughput, whereas safety requires large inter-vehicle distances. The inter-vehicle distance can be considered as a setpoint, which is restricted by legislation and by the maximum vehicle acceleration and deceleration capabilities. Other examples of limitations follow from legislation on emissions and safety, such as international directives on NO_x, HC, soot, and CO_2, a minimal inter-vehicle distance, a maximum allowable automatic deceleration, and a maximum velocity for heavy-duty vehicles. Physical limitations are, for example, limited acceleration and deceleration capabilities due to engine and brake system limitations, a minimum engine rotational velocity to prevent engine stalling, limited friction forces defining the tire-road contact characteristics, and a minimum fuel consumption and emissions.

As a result of global performance qualifiers, the performance channel $\boldsymbol{w}_p(t) \mapsto \boldsymbol{z}_p(t)$ and, correspondingly the generalized plant model $\mathcal{P}$, are not defined unambiguously for specific control problems. Furthermore, variable operating conditions result in operating-point dependency and time variations in both the dynamics of the generalized plant model $\mathcal{P}$ and the performance channel $\boldsymbol{w}_p(t) \mapsto \boldsymbol{z}_p(t)$. Hence, modeling and specification of performance requirements from a control point of view are complicated by the presence of global performance qualifiers and variable operating conditions.

Application examples

In literature, application examples and case studies solving specific automotive control problems are readily available. These examples provide a systematic, model-based control design approach for specific applications, indicating the possibilities for performance improvements. Furthermore, these examples demonstrate that controller synthesis methods are available that are particularly suitable to handle the typical challenges in the design of automotive control systems, such as variable operating conditions, constraints, or conflicting performance requirements. Finally, in various cases, the application examples show that practical applicability of the methods is feasible.

Focus of state-of-the-art controller synthesis methods often is on the theoretical problem formulation, rather than practical implementation. Practical implementation issues, possibly limiting the practical applicability of a method, are, for example, real-time computational limitations an complexity of the actual controller synthesis. Typically, nonlinear and robust controller synthesis methods are adopted in the application examples and case studies (Johansson and Rantzer, 2003; Kiencke and Nielsen, 2005). Examples are model predictive control (MPC), gain scheduling (GS) or linear parameter varying control (LPV), and robust control.

To handle nonlinear or operating-point dependent dynamics, gain scheduling (GS) or linear parameter varying (LPV) techniques are often adopted (Rugh and Shamma, 2000; Leith and Leithead, 2000). Classical GS is commonly adopted in practice, in combination with online tuning techniques (Kolmanovsky, 2008; Naus, 2007a,b). Based on experience and insight, scheduling parameters are chosen to schedule the controller parameters for specific operating conditions. Closed-loop performance and stability guarantees are evaluated by trial-and-error via extensive testing. More recent LPV techniques enable a-priori guarantees, however, often at the cost of a more involved controller synthesis. Some recent application examples of LPV controller synthesis methods in literature are air-to-fuel ratio control (Alfieri et al., 2009), lane guidance (Hingwe et al., 2002), power steering (McCann and Le, 2008), and air charge control (Kwiatkowski et al., 2009).

Model predictive control (MPC) is particularly suitable to handle constraints. Besides that, different, possibly conflicting, control objectives can be taken into account in a systematic manner. Standard MPC, requiring much online computing power and large computation effort, is especially widespread in process control, where high sampling times are often not required (Maciejowski, 2002). Recent developments on explicit MPC methods enable offline computation of the controller, and, as a result, higher online sampling rates, making MPC suitable for solving automotive control problems (Bemporad et al., 2002b). Examples in literature are idle velocity regulation (Di Cairano et al., 2008), active steering and braking for autonomous vehicles (Borelli et al., 2005), powertrain control (Saerens et al., 2008), air path management (Iwadare et al., 2007), and variable valve actuation (Bengtsson et al., 2006).

Robust control is in particular suitable to account for unmeasured uncertainties or variations in the operating conditions (Zhou et al., 1996). For example, as a vehicle is a mass-produced product with a long life span, uncertain variabilities in the dynamics are present due to mechanical differences and wear. These variabilities can be handled appropriately in a robust control framework, see, for example, Baumann et al. (2005), designing an anti-jerk controller to prevent oscillations in the driveline. Other examples are the design of vehicle stability control (Yin et al., 2007; Güvenç et al., 2009), and active suspension control (Gaspar et al., 2003). Furthermore, robust control methods are often adopted as a basis for LPV controller synthesis. Examples are the design of an hybrid power management strategy (Inagaki et al., 2007), an active suspension system (Leite and Peres, 2005), and an anti-lock braking system (Baslamisli et al., 2007).

1.2 Problem formulation

1.2.1 Research objectives

The main goal of this thesis is to contribute to a paradigm shift from the application of pragmatic, online tuning techniques to the application of a systematic, model-based control design approach in the automotive industry. A systematic, model-based control design approach implies the use of control-oriented modeling and the specification of corresponding performance requirements as a basis for the actual controller synthesis. In practice, online tuning and calibration techniques are often adopted instead.

The use of a systematic, model-based control design approach is a prerequisite for the application of state-of-the-art controller synthesis methods. These methods enable to achieve guarantees regarding robustness, performance, stability, and optimality of the synthesized controller. Accordingly, from a practical point-of-view, a systematic, model-based control design approach forms a basis for, e.g., fulfilling increasingly stringent performance demands, and automated controller synthesis, reducing tuning and calibration effort.

To achieve this goal, the following research objectives are defined.

- Demonstrate the possibilities of and opportunities for application of a systematic, model-based control design approach for automotive control problems, validating the availability of state-of-the-art controller synthesis methods that are specifically suitable to cope with the typical challenges in automotive control problems according to Table 1.2.
- Evaluate in what sense the typical challenges in automotive control problems according to Table 1.2 limit or complicate the application of a systematic, model-based

control design approach and derive guidelines to cope with these challenges, specifically focusing on the development of control-oriented models and the specification of corresponding performance requirements.

- Assess the practical applicability of state-of-the-art controller synthesis methods for control problems in the automotive industry, and identify typical limitations of the methods, focusing on practical implementation.

1.2.2 Research approach

The adopted research approach targets to acquire insight in the properties of control problems in the automotive industry via several case studies. In literature, application examples and case studies solving specific automotive control problems are readily available. Analogously, several case studies are considered in this thesis. An overview of the case studies is given in Table 1.3.

First, the case studies represent practical application examples that are specifically suitable to apply a specific set of state-of-the-art controller synthesis methods, being robust control, model predictive control (MPC), and gain scheduling (GS) or linear parameter varying (LPV) control. Literature indicates these synthesis methods to be specifically suitable to handle the typical challenges according to Table 1.2. Moreover, actual control problems, rather than theoretical application examples, are considered, which enables to demonstrate the possibilities and opportunities for application of a model-based control design approach for actual automotive control problems. Each of the four cases has a specific focus, comprising driver assistance and advanced driver assistance systems, powertrain control, the use of vehicle state estimators, and inter-vehicle communication. Accordingly, the case studies target contributing to active fields of research, addressing the first research objective.

Second, the case studies are chosen to be representative examples of automotive control problems, in the sense that the typical challenges according to Table 1.2 are considered.

Table 1.3: Overview of the case studies.

case	title
I.	Robust control of a clutch system to prevent judder-induced driveline oscillations.
II.	Gain scheduling and linear parameter varying control design for heavy-duty vehicle cruise control (CC).
II.	Design and implementation of parameterized adaptive cruise control (ACC): an explicit model predictive control approach.
IV.	String-stable cooperative adaptive cruise control (CACC) design and experimental validation, a frequency-domain approach.

Focus of this research is in particular on variable operating conditions and global performance qualifiers.

- Variable operating conditions are considered in the cases I and II. In case I, the effect of wear is considered, which introduces time-varying dynamics. As this effect cannot be measured, a robust controller synthesis is proposed as a solution. In case II, the operating conditions vary as a result of variable loading and gear shifting. These variations can be measured and are incorporated explicitly in the control design using GS and LPV controller synthesis methods.
- In the cases III and IV, global performance qualifiers are considered. In case III, conflicting requirements are translated into operating-point dependent constraints. Depending on the inputs and the state of the system, different constraints are active. Adopting an MPC controller synthesis method, the constraints are explicitly taken into account in the controller synthesis. In case IV, traffic throughput is considered as a global performance qualifier, which is translated into a sufficient condition that is valid for each individual vehicle. This sufficient condition imposes a constraint on the dynamics of each vehicle, whereas the global performance qualifier imposes a constraint on the total traffic dynamics. Hence, this sufficient condition can be used as a basis for a decentralized controller design.

The case studies are used to acquire insight in the properties of control problems in the automotive industry. Focus is in particular on the required control-oriented modeling and the specification of corresponding performance requirements. Based on this insight, the definition of a more generic classification of automotive control problems is pursued, thus addressing the second research objective.

Third, in all cases, focus is on implementation in practice, thus addressing the third research objective. The case studies are defined and executed in close cooperation with DAF Trucks N.V.[1] and TNO Automotive[2]. DAF is a Dutch OEM, producing heavy duty vehicles (HDVs). TNO Automotive is a Dutch institute for applied research, targeting the development of innovative automotive functionality. DAF and TNO facilitate actual implementation of the results in practice. Using rapid control prototyping, in all cases, practical implementation issues are evaluated via application of the resulting controller on a real vehicle and on hardware-in-the-loop setups, addressing the third research objective.

In the remainder of this section, the case studies are detailed. For each case, the problem formulation and the proposed approach are indicated.

[1] DAF Trucks N.V., P.O. box 90065, 5600 PT, Eindhoven, the Netherlands

[2] TNO Science and Industry, Business Unit Automotive, P.O. box 756, Helmond, the Netherlands

Case I: Robust control of a clutch system to prevent judder-induced driveline oscillations

An automated manual transmission (AMT) consists of an automated gearbox in combination with an automated dry plate or lock-up clutch system. Especially in HDVs, an AMT is a standard system nowadays, which is often applied. A typical problem in AMTs is the effect of clutch judder, which is a friction-induced vibration between masses in sliding contact. Clutch judder results in undesirable vibrations and oscillations in the driveline. Clutch judder may occur when the clutch is closed, which is done automatically when driving off. The causes for clutch judder are variation in the friction characteristics of the clutch-facings material as well as mechanical tolerances and misalignment in the driveline. The conditions of the clutch-facings material and of the tolerances in the driveline may change as a function of, for example, temperature, wear, and moist. Consequently, clutch judder is a commonly encountered phenomenon in clutches. To cope with the clutch judder phenomenon, a robustly stable feedback controller is designed using a robust controller synthesis method. The controller is based on a model of the driveline of an HDV. The model incorporates an uncertainty model for unmodeled friction dynamics which induce clutch judder.

Case II: Gain scheduling and linear parameter varying control design for heavy-duty vehicle cruise control

Cruise control (CC) is a widespread, commercially available functionality, which, nowadays, can be regarded as a standard automotive control system. Focusing in particular on heavy-duty vehicles (HDVs), a large operating range has to be taken into account when designing a CC system. For example, the mass of a typical HDV varies in between 7000 and 40000 kg. Commonly, this operating range is not explicitly taken into account in the controller design. As a result, the design of standard CC systems is conservative and closed-loop behavior varies over the operating range. Recent research advances on active parameter and state estimators as well as the increase in advanced electronics that become standard in vehicles, enable accurate estimates of the vehicle mass. Gain scheduling (GS) and linear parameter varying (LPV) controller synthesis approaches are adopted to incorporate the time-varying mass explicitly in the design of a CC system. Four different controller synthesis methods are compared, varying from classical GS to more recent LPV techniques. The controller design is based on a mass-dependent LPV model of an HDV, which is derived via physical modeling. Focus is on the comparison of the theoretical comprehensiveness and the practical applicability of the methods.

Case III: Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach

Adaptive cruise control (ACC) is an extension of the classic cruise control, targeting automatic vehicle following. Considering the corresponding driving behavior, ACC systems are generally designed to have specific key characteristics, such as safety, comfort, fuel economy and traffic-flow efficiency. These characteristics typically impose conflicting control objectives and introduce constraints, thus complicating the controller design. Furthermore, driver acceptance of the system requires ACC behavior to mimic human driving behavior to some extent, which is driver specific, time varying, and also situation dependent. A systematic procedure is presented to incorporate the desired key characteristics and the situation-dependency in the design of the ACC. The resulting ACC is parameterized by the key characteristics safety and comfort, with at most one tuning variable for each characteristic. An MPC controller synthesis is adopted to cope with the conflicting controller requirements, the constraints, and the situation dependency of the performance requirements.

Case IV: String-stable cooperative adaptive cruise control design and experimental validation, a frequency-domain approach

Decreasing inter-vehicle distances promises significant benefits such as an increased traffic throughput and a reduced aerodynamic drag force, thus decreasing fuel consumption. If either drivers are encouraged to decrease the inter-vehicle distance, or commercially available adaptive cruise control (ACC) functionality is employed, undesired oscillations in the traffic flow, so-called string unstable driving behavior may occur. When standard ACC functionality is extended with wireless inter-vehicle communication, driving at small inter-vehicle distances is possible, while maintaining string stability. The result is called cooperative adaptive cruise control (CACC). Although practical implementation of CACC is challenging, it is technically possible. However, it is difficult to specify the benefit for individual vehicles. A frequency-domain based definition of string stability is derived, targeting performance specification for individual vehicles within everyday traffic. A CACC system is designed, focusing on the feasibility of implementation within the current infrastructure. The inter-vehicle spacing is used as a performance specification, considering guaranteed string stability as a constraint. Considering the minimal inter-vehicle spacing, the performance of the CACC system is compared to the performance of a standard ACC system.

1.2.3 Contributions and outline

The main goal of this thesis is to contribute to a paradigm shift from the application of pragmatic, online tuning techniques to the application of a systematic, model-based control design approach in the automotive industry. The first contribution of this thesis involves a classification of automotive control problems. The classification facilitates a characterization on the basis of the required modeling and the specification of performance requirements. Automotive control problems at a full-vehicle level, at an in-vehicle level, and at a component level are distinguished. The classification is based on insight that is acquired via both the results of relevant case studies (see Chapters 2 to 5), and experience at DAF Trucks N.V. and TNO Automotive (Naus, 2007a,b).

Second, following the classification, a discussion on the limitations, points-of-attention and guidelines for control-oriented modeling and the specification of corresponding performance requirements is presented. Focus is on managing the typical challenges in automotive control problems according to Table 1.2. In this research, variable operating conditions and global performance qualifiers are considered. The classification and the corresponding discussion are presented in Chapter 6.

A third contribution of this thesis involves the practical application of the proposed control concepts. A hardware-in-the-loop setup, a DAF XF105, an Audi S8 and two Citroën C4's are used. Both the possibilities and the limitations for practical applicability of the adopted controller synthesis methods are identified (see Chapter 6).

Finally, as the case studies involve actual control problems in the automotive industry, specific contributions to active fields of research are obtained in each case. In Chapter 2, the effect of clutch judder, in particular for heavy duty vehicles (HDVs), is considered. The contribution involves the design of a robustly stable feedback controller to actively damp judder-induced driveline oscillations during drive-off maneuvers. Furthermore, experimental validation on a hardware-in-the-loop setup of a heavy-duty vehicle is presented. The chapter is based on Naus et al. (2010c). Related results are reported in Beenakkers (2007) and Naus et al. (2008c).

A solution for variations in the closed-loop behavior of cruise control functionality for heavy duty vehicles is proposed in Chapter 3. The contribution of this chapter is a comparison of relevant GS and LPV controller synthesis methods for the design of a cruise control for HDVs, targeting to expose the limitations of the classical gain scheduling methods that are often applied in practice and to assess the practical applicability of more recent linear parameter varying methods. Accordingly, a DAF XF105 is used for experimental validation. Related results are reported in Diepen (2009).

A systematic procedure for the design and tuning of the vehicle-independent part of an adaptive cruise control (ACC) is presented in Chapter 4. The contribution is the design of an ACC which is parameterized by the key characteristics, with at most one tuning variable for each characteristic. Hence, after the parameterization, the specific setting of the ACC can easily be changed, possibly even by the driver. Next to presenting this systematic design approach, the implementation of the ACC on an Audi S8 and the results of on-the-road experiments are discussed. The chapter is based on Naus et al. (2010b). In Naus et al. (2010e) and Keulen et al. (2009b,c), it is demonstrated that the framework is generic in the sense that different global performance qualifiers are considered. Related results are reported in Naus et al. (2008a), Naus et al. (2008b), Bleek (2007) and Reichardt (2007).

Finally, in Chapter 5, the design of a cooperative adaptive cruise control (CACC) is presented. The contribution of this research involves, first, the design of a CACC system focusing on the feasibility of implementation and the definition of a corresponding sufficient, frequency-domain condition for string stability of heterogeneous traffic. Second, implementation on two Citroën C4's and corresponding experimental validation of the proposed CACC system are discussed. The chapter is based on (Naus et al., 2010d). Preliminary and related results are reported in Vugts (2010), Naus et al. (2010a), and Naus et al. (2009a).

The thesis is closed with a summary of the main conclusions and recommendations on model-based control for automotive applications in Chapter 7.

CHAPTER 2

Robust control of a clutch system to prevent judder-induced driveline oscillations[1]

Abstract - Oscillations in the driveline of a vehicle, specifically originating from the clutch system, are referred to as clutch judder. Typically, judder is a result of wear-induced variations in the friction characteristics of the clutch facings material. In this chapter, the design of a robust controller to prevent judder-induced oscillations is presented. A DK iteration procedure, combining $\mathcal{H}_\infty$ controller synthesis and μ-analysis, is adopted for the robust controller design. The model for the clutch judder is based on and validated with measurements on a heavy-duty vehicle. Both simulations and hardware-in-the-loop (HIL) experiments are performed to evaluate the feasibility of the control concept.

2.1 Introduction

Focus of this research lies on heavy duty vehicles (HDVs) incorporating an automated manual transmission (AMT). An AMT typically consists of a dry plate or lock-up clutch system in combination with a gearbox. The clutch transfers torque from the engine to the driveline, which is schematically depicted in Figure 2.1.

Oscillations in the driveline specifically originating from the clutch are referred to as clutch (engagement) judder (Centea et al., 1999). In general, judder is a friction-induced

[1]This chapter is based on G. J. L. Naus, M. A. Beenakkers, R. G. M. Huisman, M. J. G. van de Molengraft and M. Steinbuch (2010). Robust control of a clutch system to prevent judder-induced driveline oscillations. *Veh. Syst. Dyn.* (accepted for publication).

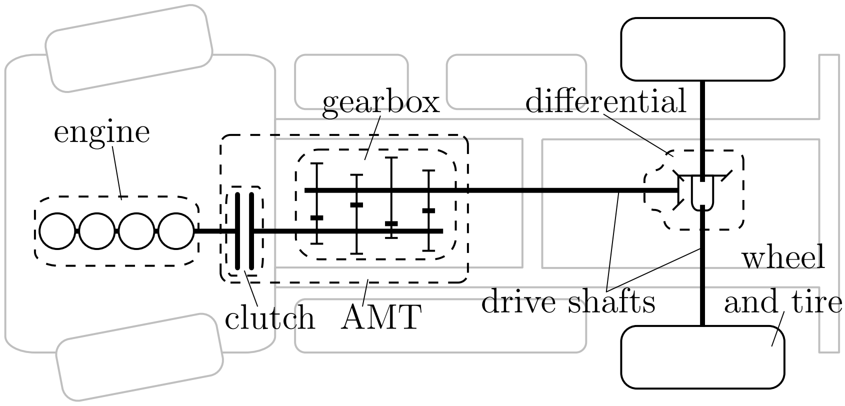


Figure 2.1: Schematic representation of a typical HDV driveline.

vibration between masses in sliding contact and can be regarded as an unstable mode of the system dynamics (see, e.g., Heckmann, 2006). In this case, the clutch plates are in sliding contact, while the engine inertia on the one side and the gearbox inertia on the other side represent the masses, see Figure 2.1 (Crowther et al., 2004; Winkel et al., 2004). Judder is a well-known phenomenon in clutches, occurring in particular during drive-off maneuvers. The resulting oscillations in the driveline inherit the first resonance frequency of the driveline, introducing undesired dynamic loads, increasing slip and wear effects in the clutch, and reducing driver comfort (Bostwick and Szadkowski, 1998).

Generally speaking, two main causes of clutch judder can be distinguished. First, variation in the friction characteristics of the clutch-facings material, and, second, mechanical tolerances and misalignments in the driveline (Winkel et al., 2004). The effect of judder may be solved by i) changing the vehicle driveline properties by means of mechanical adjustments, e.g., increasing damping and stiffness of the shafts or improving the characteristics of the clutch-facings material, or ii) application of feedback control to actively damp the oscillations in the driveline and stabilize the system. This research focuses on the latter solution.

Much research regarding active damping of driveline oscillations focuses on control of the engine output, as the engine can be regarded as an easy-to-use actuator (e.g., Bruce et al., 2005). Judder however, occurs in the slipping clutch phase, when the engine is partially decoupled from the driveline. Hence, this research focuses on the design of a controller utilizing the clutch actuator rather than the engine, where the clutch actuator provides the clamping force to close the clutch. Moreover, focus of this research is on achieving stable system dynamics to prevent driveline oscillations, rather than damping of the oscillations afterwards.

Comparable research focuses on the design of feedforward filters instead of a feedback controller (Winkel et al., 2004; Weik et al., 2004). As soon as the judder phenomenon

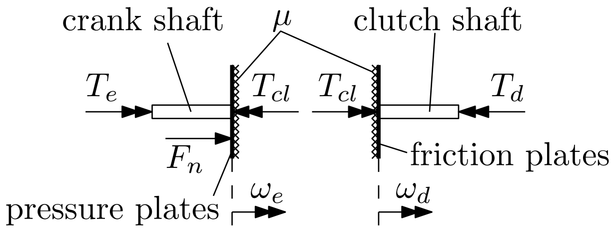


Figure 2.2: Schematic representation of a dry-clutch system, where $T_e(t)$ is the torque delivered by the engine, $\omega_e(t)$ and $\omega_d(t)$ are the corresponding rotational velocities, $T_d(t)$ is the torque delivered by the driveline, $F_n(t)$ is the clamping force, μ is the friction coefficient of the clutch-facings material, and $T_{cl}(t)$ is the torque transferred by the clutch. For clarity, the time-dependency of the signals is omitted in the figure.

is detected, a learning feedforward filter is initialized and the oscillations are opposed by the resulting feedforward signal. Although some successful results are reported, experimental validation showed that such a feedforward controller is unable to stabilize the driveline in all working conditions.

The contribution of this chapter comprises the design of a robustly stable feedback controller to actively damp judder-induced driveline oscillations during drive-off maneuvers (Beenakkers, 2007). The judder model used is validated using measurements on a real HDV. Both simulations and hardware-in-the-loop (HIL) experiments are performed to validate the feasibility of the control concept.

The problem formulation is presented in Section 2.2. The modeling and the controller synthesis are discussed in Sections 2.3 and 2.4, respectively. In Section 2.5, the results of simulations and experiments are discussed, and the chapter is closed with conclusions and recommendations.

2.2 Problem formulation

2.2.1 Modeling of the clutch system

In Figure 2.2, a schematic representation of a dry-clutch system is shown. In reality, both the pressure and the clutch plates consist of several plates, which are clamped together. The pressure plates are mounted to the crank shaft, which is connected to the engine delivering a torque $T_e(t)$. The friction plates are mounted to the clutch shaft, which is connected to the driveline delivering a torque $T_d(t)$. The plates rotate with a velocity $\omega_e(t)$ and $\omega_d(t)$, corresponding to the engine and the driveline rotational velocity, respectively. To close the clutch, a clamping force $F_n(t)$ is provided by the clutch actuator, resulting in a torque $T_{cl}(t)$ that is transferred by the clutch system.

The plate facings are covered with a material with a high friction coefficient μ. Friction-induced judder models typically comprise a combination of static and kinetic friction, μ_{st} and μ_{kin}, respectively, yielding (Crowther et al., 2004; Centea et al., 1999)

$$\mu = \operatorname{sign}(\omega_{sl}(t))\,\mu_{st} + \mu_{kin}\,\omega_{sl}(t) \tag{2.1}$$

where $\omega_{sl}(t) = \omega_e(t) - \omega_d(t)$ the difference in rotational velocity between the pressure plates and the friction plates of the clutch, and

$$\operatorname{sign}(\omega_{sl}(t)) = \begin{cases} 1, & \text{for} \quad \omega_{sl}(t) > 0 \\ 0, & \text{for} \quad \omega_{sl}(t) = 0 \\ -1, & \text{for} \quad \omega_{sl}(t) < 0 \end{cases} \tag{2.2}$$

Focus of this research is restricted to drive-off maneuvers. More specifically, focus is on driving-off on a flat road, starting from standstill. This implies $\omega_e(t) \leq \omega_d(t)$ and hence

$$\mu = \mu_{st} + \mu_{kin}\,\omega_{sl}(t) \tag{2.3}$$

Assuming uniform pressure across the surface of the clutch plates, the torque that is transferred by the clutch, $T_{cl}(t)$, is given by

$$T_{cl}(t) = F_n(t) r_m n_{cl} \mu \tag{2.4}$$

where the constants r_m and n_{cl} are the mean clutch radius and the number of clutch plates, respectively, μ is as defined in (2.3), and $F_n(t) \geq 0$ is the clamping force.

2.2.2 Clutch judder

Oscillations in the driveline specifically originating from the clutch are called clutch judder. Clutch judder is a result of unstable dynamics that are induced by variations in the friction characteristics of the clutch-facings material, for example due to wear. Define

$$\mu_{st}^* = r_m n_{cl} \mu_{st} \tag{2.5}$$

$$\mu_{kin}^* = r_m n_{cl} \mu_{kin} \tag{2.6}$$

Combining (2.3) through (2.6) yields the torque that is transferred by the clutch

$$T_{cl}(t) = F_n(t)\mu_{st}^* + F_n(t)\mu_{kin}^*\omega_{sl}(t) \tag{2.7}$$

This result shows that $F_n(t)\mu_{kin}^*$ can be regarded as a damping term. In this damping term, the value of the kinetic friction coefficient μ_{kin}, which is included in μ_{kin}^* (2.6), may vary as a function of temperature, wear, moist, etc. (Winkel et al., 2004). Typical values for μ_{kin} are (Bostwick and Szadkowski, 1998)

$$\mu_{kin} \in M_{kin,O} = [-0.001,\ 0.001] \tag{2.8}$$

Hence, focusing on drive-off maneuvers on a flat road, where $\omega_{sl}(t) \geq 0$ and $T_{cl}(t) \geq 0$, and with $r_m n_{cl}$ a positive constant and $F_n(t) \geq 0$, instabilities are induced for $\mu_{kin} < 0$.

Oscillations in the driveline that are a result of this instability are called clutch judder. This research targets the design of a robustly stable feedback controller to actively stabilize the system dynamics, thus preventing judder-induced driveline oscillations, and, if necessary, damp the resulting oscillations. Robustness is required for variations in the kinetic friction coefficient μ_{kin}. As the actual value of μ_{kin} is not measured and neither is easy to estimate, it is regarded as an uncertain variability.

2.3 Modeling

2.3.1 Modeling of the driveline

In Figure 2.3, a mass-spring-damper model of the HDV driveline is shown. The engine torque $T_e(t)$ and the clamping force $F_n(t)$ are the available control inputs of the model. Taking into account the gearbox and the final-drive ratios, all inertias are lumped into an engine, a driveline and a vehicle inertia, which are represented by J_e, J_d and J_v, respectively. The vehicle inertia J_v includes all external loads. The corresponding rotational velocities are represented by $\omega_e(t)$, $\omega_d(t)$ and $\omega_v(t)$, respectively. Finally, the damping and stiffness of the drive shafts are represented by d_s and k_s, respectively, and the engine damping is represented by d_e.

Standard, a tachograph at the output of the gearbox is used to measure the vehicle velocity. In the mass-spring-damper model in Figure 2.3, this corresponds to the rotational velocity $\omega_d(t)$ of the driveline inertia J_d. Furthermore, the engine rotational velocity $\omega_e(t)$ is measured. Hence, when the clutch is opened and the engine is decoupled from the driveline, rotational velocities are measured on both sides of the clutch system.

Based on the mass-spring-damper model of the HDV driveline, define the state $\boldsymbol{x}(t) = (\omega_e(t), \omega_d(t), \omega_v(t), \theta(t))^T$, where $\theta(t)$ represents the winding of the drive shafts, the input vector $\boldsymbol{u}(t) = (T_e(t), F_n(t))^T$, and the output vector $\boldsymbol{y}(t) = (\omega_e(t), \omega_d(t))^T$. Correspondingly, a nonlinear, multi-input, multi-output (MIMO) model description $\mathcal{M}_{nl}$ is derived,

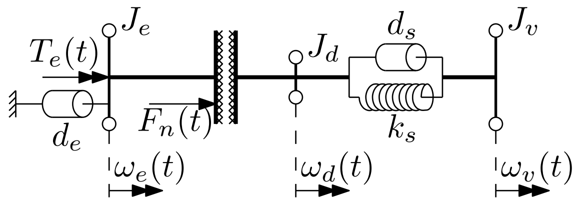


Figure 2.3: Schematic representation of a mass-spring-damper model of an HDV driveline, where $T_e(t)$ is the engine torque, $F_n(t)$ is the clamping force, J_e, J_d and J_v are the engine, the driveline and the vehicle inertias, respectively, $\omega_e(t)$, $\omega_d(t)$ and $\omega_v(t)$ are the corresponding rotational velocities, respectively, d_s and k_s are the damping and the stiffness of the drive shafts, respectively, and d_e is the engine damping.

which is given by

$$\mathcal{M}_{nl} : \left\{ \begin{array}{rcl} J_e\dot{\omega}_e(t) & = & T_e(t) - d_e\omega_e(t) - T_{cl}(t) \\ J_d\dot{\omega}_d(t) & = & T_{cl}(t) + k_s\theta(t) + d_s\dot{\theta}(t) \\ J_v\dot{\omega}_v(t) & = & -k_s\theta(t) - d_s\dot{\theta}(t) \\ \dot{\theta}(t) & = & \omega_v(t) - \omega_d(t) \end{array} \right. \tag{2.9}$$

where $T_{cl}(t) = T_{cl}(t, F_n(t)\omega_{sl}(t))$ is according to (2.7), rendering the model $\mathcal{M}_{nl}$ (2.9) nonlinear.

Accordingly, define the operating point $(F_n(t),\ \omega_{sl}(t))$, with

$$F_n(t) \in F_{n,O} = [F_{n,min},\ F_{n,max}] = [0,\ 8000]\ \text{N} \tag{2.10}$$

$$\omega_{sl}(t) \in \Omega_{sl,O} = [\omega_{sl,min}, \omega_{sl,max}] = [0,\ 84]\ \text{rad/s} \triangleq [0,\ 800]\ \text{rpm} \tag{2.11}$$

The nonlinear model (2.9) is linearized around the nominal operating point $(\overline{F}_n,\ \overline{\omega}_{sl})$, with

$$\overline{F}_n = \frac{1}{2}(F_{n,min} + F_{n,max}) \tag{2.12}$$

$$\overline{\omega}_{sl} = \frac{1}{2}(\omega_{sl,min} + \omega_{sl,max}) \tag{2.13}$$

and the corresponding nominal working conditions $\boldsymbol{x}(t) = \overline{\boldsymbol{x}}$, $\boldsymbol{u}(t) = \overline{\boldsymbol{u}}$ and $\boldsymbol{y}(t) = \overline{\boldsymbol{y}}$. Focusing on (small) perturbations around these nominal working conditions $\tilde{\boldsymbol{x}}(t) = \delta\boldsymbol{x}(t)$, $\tilde{\boldsymbol{u}}(t) = \delta\boldsymbol{u}(t)$ and $\tilde{\boldsymbol{y}}(t) = \delta\boldsymbol{y}(t)$ yields a linearized model description

$$\mathcal{M}_l : \left\{ \begin{array}{rcl} \dot{\tilde{\boldsymbol{x}}}(t) & = & \mathbf{A}(\overline{F}_n\mu^*_{kin})\tilde{\boldsymbol{x}}(t) + \mathbf{B}(\overline{\omega}_{sl}\mu^*_{kin})\tilde{\boldsymbol{u}}(t) \\ \tilde{\boldsymbol{y}}(t) & = & \mathbf{C}\tilde{\boldsymbol{x}}(t) + \mathbf{D}\tilde{\boldsymbol{u}}(t) \end{array} \right. \tag{2.14a}$$

where

$$\begin{aligned}
\mathbf{A}(\overline{F}_n\mu^*_{kin}) &= \begin{pmatrix} \frac{-d_e-\overline{F}_n\mu^*_{kin}}{J_e} & \frac{\overline{F}_n\mu^*_{kin}}{J_e} & 0 & 0 \\ \frac{\overline{F}_n\mu^*_{kin}}{J_d} & \frac{-d_s-\overline{F}_n\mu^*_{kin}}{J_d} & \frac{d_s}{J_d} & \frac{k_s}{J_d} \\ 0 & \frac{d_s}{J_v} & \frac{-d_s}{J_v} & \frac{-k_s}{J_v} \\ 0 & -1 & 1 & 0 \end{pmatrix} \\
\mathbf{B}(\overline{\omega}_{sl}\mu^*_{kin}) &= \begin{pmatrix} \frac{1}{J_e} & \frac{-\overline{\omega}_{sl}\mu^*_{kin}-\mu^*_{st}}{J_e} \\ 0 & \frac{\overline{\omega}_{sl}\mu^*_{kin}+\mu^*_{st}}{J_d} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \\
\mathbf{C} &= \left(\mathbf{I}^{2\times2}, \mathbf{0}^{2\times2}\right), \qquad \mathbf{D} = \left(\mathbf{0}^{2\times2}\right)
\end{aligned} \tag{2.14b}$$

2.3.2 Sensor, actuator and communication dynamics

Compared to practice, the theoretical models $\mathcal{M}_{nl}$ (2.9) and $\mathcal{M}_l$ (2.14) lack actuator and sensor dynamics. Furthermore, a significant communication delay is present due to the use of CAN-bus communication. Measurement results indicate that the actuator and sensor dynamics are relatively fast compared to the dynamics of the driveline. Hence, following Chen (1997), only a limited bandwidth of both the actuators and the sensors that are used, is modeled via low-pass filters. The communication delay due to CAN-bus communication is modeled using a Padé approximation.

For the sake of clarity, the additional sensor, actuator and communication dynamics are not taken into account in the models and the derivations presented in this chapter. Focus is on the discussion of the control design approach, which does not change when these additional dynamics are taken into account. The results, however, do change. Hence, the results that are shown in the figures do incorporate these additional dynamics. For example, in Figure 2.4, an additional time-delay is clearly visible in the phase plot.

2.3.3 Model validation

For a driveline with closed clutch, $\omega_e(t) = \omega_d(t)$ holds. Considering the nonlinear model description $\mathcal{M}_{nl}$ (2.9), the order of the model reduces and a linear SISO model $\mathcal{M}_p : T_e(t) \mapsto \omega_e(t)$ results:

$$\mathcal{M}_p : \begin{cases} (J_e + J_d)\dot{\omega}_e(t) &= T_e(t) - d_e\omega_e(t) + k_s\theta(t) + d_s\dot{\theta}(t) \\ J_v\dot{\omega}_v(t) &= -k_s\theta(t) - d_s\dot{\theta}(t) \\ \dot{\theta}(t) &= \omega_v(t) - \omega_e(t) \end{cases} \tag{2.15}$$

The model described by (2.15) is compared to experimental measurement results obtained with a real HDV, utilizing frequency response measurement techniques (Zalm

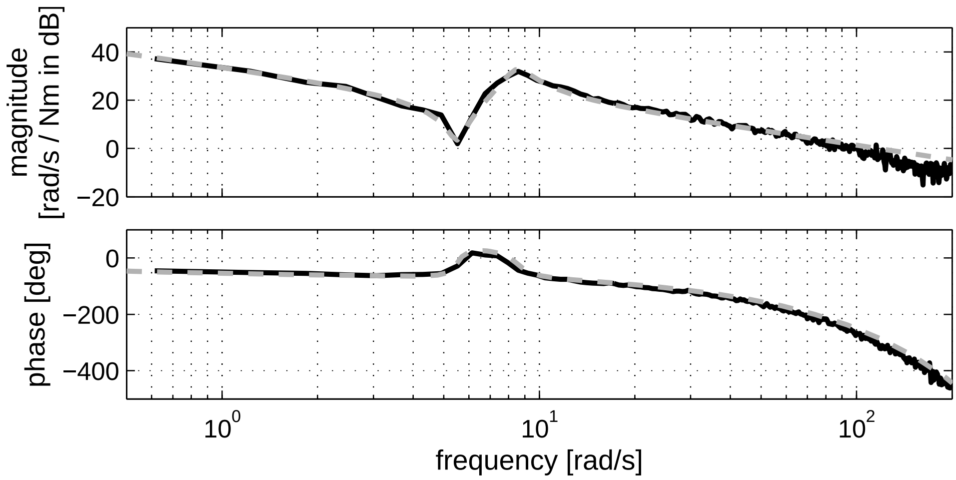


Figure 2.4: Bode plot of the linear model $\mathcal{M}_p : T_e(t) \mapsto \omega_e(t)$ (2.15), including actuator, sensor, and communication dynamics (dashed grey), compared to measurement results of a real truck (solid black).

et al., 2008). On the basis of these measurements, the actual parameter values of $\mathcal{M}_p$ are estimated by fitting the model onto the measurement results, which is shown in Figure 2.4. These values are then used in the models $\mathcal{M}_{nl}$ and $\mathcal{M}_l$, describing the system with a slipping clutch. Comparison of the measured frequency response and the fitted model shows, first, that the model describes the main characteristics of the dynamics appropriately, and second, that the fitted model resembles the measurement results well.

In Figure 2.5, the response of $\omega_d(t)$ to a step in the clamping force $F_n(t)$ for constant $T_e(t) = \overline{T_e}$ is shown. A simulation of the model $\mathcal{M}_{nl}$ (2.14) is compared to measurements obtained with a real HDV. The results confirm that both the eigenfrequency of the driveline and the corresponding damping are modeled appropriately. Moreover, the eigenfrequency coincides with the resonance peak of the frequency response function shown in Figure 2.4.

2.3.4 Model characteristics

Regarding the linear model $\mathcal{M}_l$ (2.14), the terms in the upper left 3×3 block of $\mathbf{A}(\overline{F}_n \mu^*_{kin})$ can be regarded as damping terms. Assume for the moment $d_e = d_s = 0$. The sign and size of these damping terms then depend on the sign and the size of $\overline{F}_n$ and $\mu^*_{kin} = r_m n_{cl} \mu_{kin}$ (2.6). As mentioned before, μ_{kin} (2.8) may vary as a result of wear, moist or temperature. With $r_m > 0$ and $n_{cl} > 0$, this may result in a negatively valued μ^*_{kin}. As $\overline{F}_n \geq 0$ holds, this induces negatively valued damping terms. In Figure 2.6, the frequency response function $F_n(t) \mapsto \omega_d(t)$ of the model $\mathcal{M}_l$ (2.14) is shown for two different values

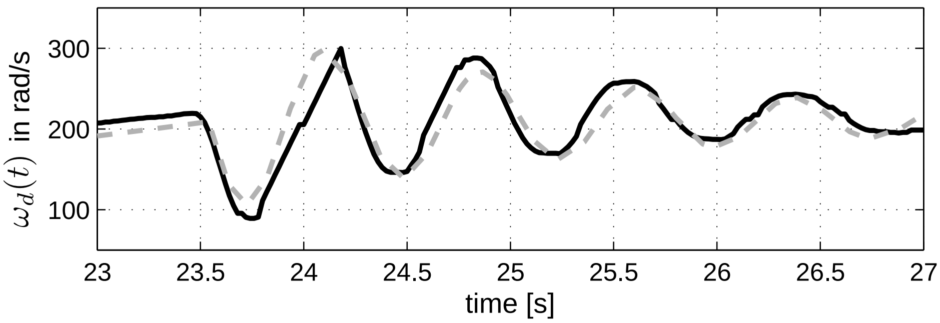


Figure 2.5: Step response of the model $\mathcal{M}_{nl} : F_n(t) \mapsto \omega_d(t)$ for constant $T_e(t) = \overline{T_e}$ (dashed grey), compared to measurement results (solid black). The step input follows from suddenly opening the clutch while driving-off.

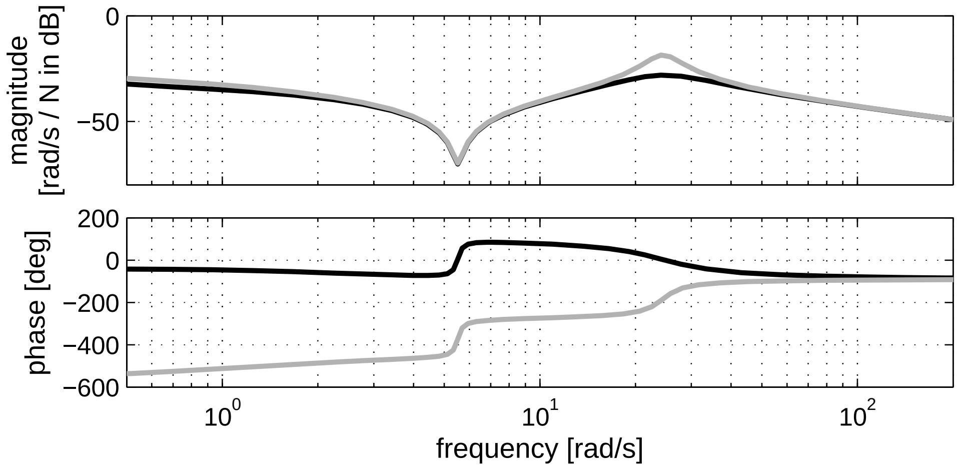


Figure 2.6: Bode plot of the linearized model $F_n(t) \mapsto \omega_d(t)$ for $\mu_{kin} > 0$ (black) and $\mu_{kin} < 0$ (grey).

of μ_{kin}, and constant $T_e(t) = \overline{T_e}$. For $\mu_{kin} > 0$, the model is stable, whereas for $\mu_{kin} < 0$ three unstable poles are present. As a result, the standard Bode phase-gain relationship does not hold for the unstable model (Skogestad and Postlethwaite, 2005). Indeed, both at low frequencies and at about 25 rad/s, opposite behavior can be observed. The phase of the unstable model increases rather than decreases where the unstable poles are located (see Figure 2.6). Hence, $\mu_{kin} < 0$ indeed leads to instability, which explains the term 'negatively damped' or 'self-induced' judder (Centea et al., 1999; Bostwick and Szadkowski, 1998; Yamada and Ando, 1995).

The goal of this research is to robustly stabilize the system dynamics. As discussed in the introduction, the clamping force $F_n(t)$, rather than the engine torque $T_e(t)$ will be used as the control variable. This approach is based on a Hankel singular value (HSV) analysis of the model $\mathcal{M}_l$ (2.14). The HSV ratio, i.e. the ratio of the largest and the smallest HSV of a linear system, enables to assess the observability and controllability of this system. For a large ratio, the system incorporates modes that are less controllable or observable (Skogestad and Postlethwaite, 2005). The smallest HSV corresponds to the eigenfrequency of the driveline, which is excited by the effect of clutch judder. The HSV analysis indicates that the engine, using the engine torque $T_e(t)$ will not be able to attenuate driveline oscillations appropriately, whereas the clutch actuator, using the clamping force $F_n(t)$ is. This is a logical result, as the engine is (partly) decoupled from the rest of the driveline when the clutch is slipping, and the torque that is transferred by the clutch is directly controlled by the clutch actuator.

Consequently, the clamping force $F_n(t)$ is used as a control signal to stabilize the system and attenuate possible driveline oscillations. Furthermore, the engine torque $T_e(t)$ is used to enable control of the engine rotational velocity when the clutch is opened, i.e., when the driveline is decoupled from the engine.

Furthermore, a relative gain array (RGA) analysis shows that the dynamics of the MIMO model $\mathcal{M}_l$ (2.14) can be regarded as decoupled in the relevant frequency range, facilitating the design of two single-input, single-output (SISO) controllers. An RGA analysis provides a measure for the interaction in a model (Bristol, 1966). In case the RGA is close to identity, crosswise interactions are relatively small. In Figure 2.7, the magnitudes of the diagonal and the off-diagonal RGA terms of $\mathcal{M}_l$ (2.14) are shown (note that both the two diagonal and the two off-diagonal elements of the RGA matrix of this two-input, two-output model are equal). As is discussed further on, the desired closed-loop bandwidth of the system is in the order of 10^1 rad/s. At this frequency, the RGA is close to identity. Hence, the MIMO model $\mathcal{M}_l$ (2.14) can be regarded as decoupled in the relevant frequency range.

2.4 Controller synthesis

2.4.1 Sequential loop closing

The RGA analysis shows that the off-diagonal interaction of the MIMO model $\mathcal{M}_l$ (2.14) is small in the relevant frequency range (see Section 2.3.4). Consequently, the model $\mathcal{M}_l$ can be regarded as a combination of two SISO models, and a diagonal controller $K(s) = \mathrm{diag}\,(K_e(s), K_d(s))$ instead of a full MIMO controller is designed. As we are dealing with two linear SISO models with some small interaction, sequential loop closing (SQL) techniques are adopted for the design of the controller (Hovda and Skogestad, 1994). In this way, the (small) off-diagonal interaction terms are accounted for, whereas this is not the case when two separate SISO controllers would be designed.

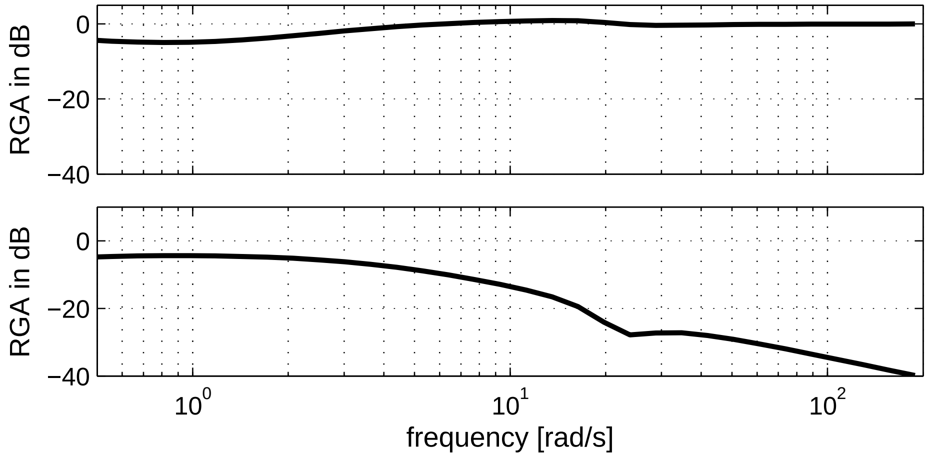


Figure 2.7: Result of the RGA analysis of $\mathcal{M}_l$ (2.14). In the upper plot, the magnitude of the diagonal elements is shown. In the lower plot, the magnitude of the off-diagonal elements is shown. Note that both the diagonal and the off-diagonal elements of the RGA matrix of this two-input, two-output model are equal.

Transforming the state-space representation $\mathcal{M}_l$ (2.14) to a transfer function $H(s)$ yields, with slight abuse of notation

$$\begin{pmatrix} \omega_e \\ \omega_d \end{pmatrix} = \underbrace{\begin{pmatrix} H_e(s) & H_{ed}(s) \\ H_{de}(s) & H_d(s) \end{pmatrix}}_{H(s)} \begin{pmatrix} T_e \\ F_n \end{pmatrix} \tag{2.16}$$

where $H(s)$ is partitioned into four transfer functions. In Figure 2.8, a schematic representation of $H(s)$ in combination with the diagonal controller $K(s)$ is shown. The references for the engine and driveline rotational velocities are denoted $\omega_{e,d}(t)$ and $\omega_{d,d}(t)$, respectively.

Using SQL techniques, all loops in the model are closed sequentially by designing corresponding SISO controllers. From the HSV analysis it is concluded that the clamping force $F_n(t)$ is used to stabilize the system and attenuate possible driveline oscillations (see Section 2.3.4). Hence, $K_d(s)$ is designed first to stabilize the model. Next, performance of the system is optimized by the design of $K_e(s)$. When designing $K_e(s)$, the already designed $K_d(s)$ has to be accounted for, which yields an equivalent model $H_e^*(s)$. The model $H_e^*(s)$ follows directly from the scheme in Figure 2.8, yielding

$$H_e^*(s) = H_e(s) - \frac{H_{ed}(s)K_d(s)H_{de}(s)}{1 + H_d(s)K_d(s)} \tag{2.17}$$

The design of $K_e(s)$ is relatively easy compared to the design of $K_d(s)$, as $H_e^*(s)$ represents a stable system. Furthermore, stabilization of the driveline is the main topic of this

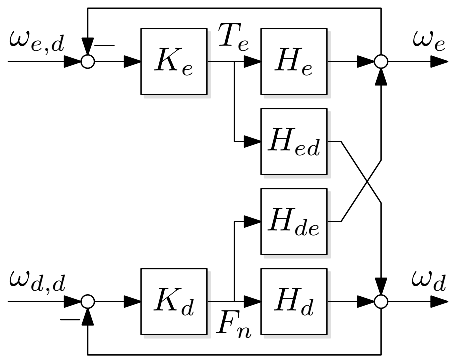


Figure 2.8: Schematic representation of the MIMO model $H(s)$ and a corresponding diagonal controller $K(s) = \text{diag}(K_e(s), K_d(s))$, where, for clarity, both the frequency-domain dependency of the models and the time-dependency of the signals are omitted.

research. Therefore, the remainder of this chapter focuses on the design of $K_d(s)$, and the design of $K_e(s)$ will not be discussed further.

In designing $K_d(s)$, only $H_d(s)$ has to be taken into account, as $K_e(s) = 0$ holds in this step of the controller design. The corresponding state-space model $\mathcal{M}_d$, with state vector $\boldsymbol{x}_d(t) = \tilde{\boldsymbol{x}}(t)$, input $u_d(t) = \tilde{F}_n(t)$ and output $y_d(t) = \tilde{\omega}_d(t)$ (see (2.14)), is defined by:

$$\mathcal{M}_d : \begin{cases} \dot{\boldsymbol{x}}_d(t) &= \mathbf{A}_d(\overline{F}_n\mu^*_{kin})\boldsymbol{x}_d(t) + \mathbf{B}_d(\overline{\omega}_{sl}\mu^*_{kin})u_d(t) \\ y_d(t) &= \mathbf{C}_d\boldsymbol{x}_d(t) + \mathbf{D}_d u_d(t) \end{cases} \tag{2.18a}$$

where

$$\begin{aligned} &\mathbf{A}_d(\overline{F}_n\mu^*_{kin}) = \mathbf{A}(\overline{F}_n\mu^*_{kin}) \\ &\mathbf{B}_d(\overline{\omega}_{sl}\mu^*_{kin}) = \left(\tfrac{-\overline{\omega}_{sl}\mu^*_{kin}-\mu^*_{st}}{J_e}, \tfrac{\overline{\omega}_{sl}\mu^*_{kin}+\mu^*_{st}}{J_d}, 0, 0\right)^T \\ &\mathbf{C}_d = (0,1,0,0)\,, \qquad \mathbf{D}_d = 0 \end{aligned} \tag{2.18b}$$

2.4.2 Uncertainty modeling

Focusing on the design of a controller $K_d(s)$ for the model $\mathcal{M}_d$ (2.18), robust stability with respect to the uncertain, variable parameter μ_{kin} (2.8) is required. Furthermore, this should hold for the entire operating range $(F_n(t),\, \omega_{sl}(t)) \in F_{n,O} \times \Omega_{sl,O}$ (see Section 2.3.1). However, in this specific case, the operating point $(F_n(t),\, \omega_{sl}(t))$ only influences the size and the sign of the uncertainty, whereas the rest of the dynamics are independent of the operating point. The operating point at which the model is linearized, $(\overline{F}_n,\, \overline{\omega}_{sl})$, can be regarded as a gain for the uncertain parameter μ_{kin}, via the terms $\overline{F}_n\mu^*_{kin}$ and $\overline{\omega}_{sl}\mu^*_{kin}$.

Consequently, to achieve robust stability for the entire operating range, the operating range $F_{n,O} \times \Omega_{sl,O}$ is included in the uncertainty. As is discussed further on, a constant, operating-point independent performance requirement is used, validating this approach.

However, the resulting ranges of variation for the uncertainties in the model are relatively large. Hence, to reduce conservatism of the controller design, the fact that only a specific (drive-off) maneuver is considered, is exploited.

During a drive-off maneuver on a flat road, the clamping force $F_n(t)$ will typically increase from $F_{n,min}$ to $F_{n,max}$ (2.10), while the rotational slip velocity $\omega_{sl}(t)$ will typically decrease from $\omega_{sl,max}$ to $\omega_{sl,min}$ (2.11). Consequently, a coupling between $F_n(t)$ and $\omega_{sl}(t)$ is present. To decrease conservatism, this coupling is taken into account using a linear relation between the variables of the operating point at which the model is linearized, yielding

$$\overline{\omega}_{sl} = \omega_{sl,max} - \frac{\omega_{sl,max}}{F_{n,max}}\overline{F}_n \tag{2.19}$$

For simplicity, a linear relationship is used. Other relationships can be adopted analogously.

Accordingly, define the uncertainties

$$\xi_1 = \mu_{kin} \in M_{kin,O} \tag{2.20}$$

$$\xi_2 = \overline{F}_n \in F_{n,O} \tag{2.21}$$

where $M_{kin,O}$ is as in (2.8) and $F_{n,O}$ is as in (2.10). Substituting (2.19) through (2.21) in the model $\mathcal{M}_d$ (2.18), yields

$$\mathcal{M}_d : \left\{ \begin{array}{rcl} \dot{\boldsymbol{x}}_d(t) & = & \mathbf{A}_d(\xi_1\xi_2)\boldsymbol{x}_d(t) + \mathbf{B}_d(\xi_1, \xi_1\xi_2)u_d(t) \\ y_d(t) & = & \mathbf{C}_d\boldsymbol{x}_d(t) + \mathbf{D}_d u_d(t) \end{array} \right. \tag{2.22}$$

Next, define the upper and lower bounds $\xi_i^- \leq \xi_i \leq \xi_i^+$, the nominal values $\xi_{i,n} = \frac{1}{2}(\xi_i^- + \xi_i^+)$, and the scaling factors $s_i = \frac{1}{2}(\xi_i^+ - \xi_i^-)$, $i \in \{1, 2\}$. Using normalized real-valued perturbations $\delta_i \in [-1, 1]$, $i \in \{1, 2\}$, the parametric uncertainties are decomposed into a nominal part and an uncertain part

$$\begin{array}{rcl} \xi_1 & = & \xi_{1,n} + \delta_1 s_1 \\ \xi_2 & = & \xi_{2,n} + \delta_2 s_2 \end{array} \tag{2.23}$$

Substituting this in the state-space model $\mathcal{M}_d$ (2.22), yields the perturbed state-space model $\mathcal{M}_{d,p}$

$$\mathcal{M}_{d,p} : \left\{ \begin{array}{rcl} \dot{\boldsymbol{x}}_d(t) & = & (\mathbf{A}_n + \mathbf{A}_\delta)\boldsymbol{x}_d(t) + (\mathbf{B}_n + \mathbf{B}_\delta)u_d(t) \\ y_d(t) & = & \mathbf{C}_n\boldsymbol{x}_d(t) + \mathbf{D}_n u_d(t) \end{array} \right. \tag{2.24}$$

where $\mathbf{A}_n = \mathbf{A}_d(\xi_{1,n}\xi_{2,n})$, $\mathbf{A}_\delta = \mathbf{A}_d(\xi_1\xi_2) - \mathbf{A}_n$, and, analogously, $\mathbf{B}_n = \mathbf{B}_d(\xi_{1,n}, \xi_{1,n}\xi_{2,n})$, $\mathbf{B}_\delta = \mathbf{B}_d(\xi_1, \xi_1\xi_2) - \mathbf{B}_n$. Furthermore, $\mathbf{C}_n = \mathbf{C}_d$ and $\mathbf{D}_n = \mathbf{D}_d$.

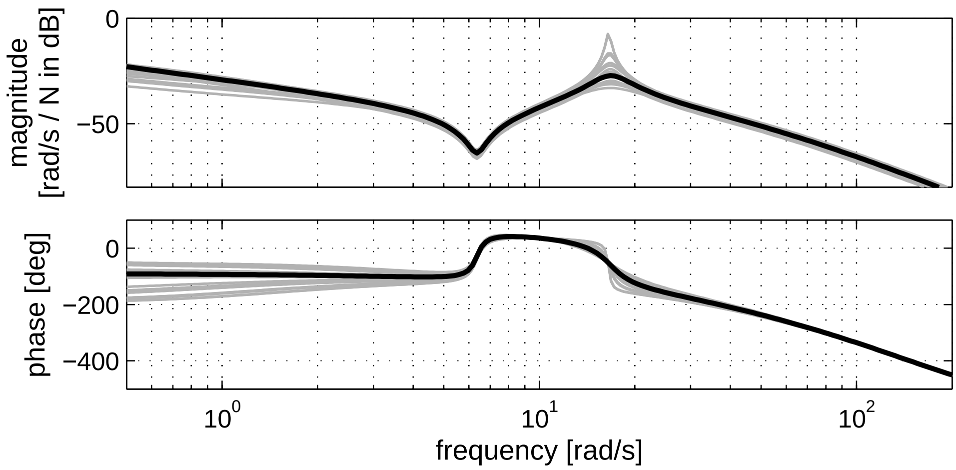


Figure 2.9: Bode plot of the perturbed model $\mathcal{M}_{d,p}$ (2.24), for $\xi_1 \in M_{kin,O}$ (2.8) and $\xi_2 \in F_{n,O}$ (2.10), including actuator and sensor dynamics as well as communication delay. The corresponding nominal model is plotted in black.

In Figure 2.9, Bode plots of $\mathcal{M}_{d,p}$ are shown for variable δ_1 and δ_2, including the stable, nominal model $\mathcal{M}_{d,p,n} = \mathcal{M}_{d,p}(\delta_1 = \delta_2 = 0)$. The plots indicate that at most only one unstable pole is present in the model: at low frequencies, the standard Bode phase-gain relationship does not hold for the unstable model. Comparing this to the results in Figure 2.6 shows that the maximum number of unstable poles is decreased. This is a result of reducing the variations of the uncertainties, which indicates that taking into account the coupling (2.19) indeed reduces conservatism.

2.4.3 Linear fractional transformation

For the purpose of robust control design, a linear fractional transformation (LFT) of the model $\mathcal{M}_{d,p}$ (2.24) is made (Doyle, 1984). Define the uncertainty matrix $\mathbf{\Delta} = \mathrm{diag}(\delta_1, \delta_2)$ and the corresponding input and output vectors $\boldsymbol{y}_\Delta(t) = (y_{\delta_1}(t), y_{\delta_2}(t))^T$ and $\boldsymbol{u}_\Delta(t) = (u_{\delta_1}(t), u_{\delta_2}(t))^T = \mathbf{\Delta}\boldsymbol{y}_\Delta(t)$, respectively. Following Steinbuch et al. (1992), the corresponding augmented model $\mathcal{M}_{d,a}$, with input and output vectors $\boldsymbol{u}_a(t) = (\boldsymbol{u}_\Delta(t), u_d(t))^T$ and $\boldsymbol{y}_a(t) = (\boldsymbol{y}_\Delta(t), y_d(t))^T$, respectively, is defined as

$$\mathcal{M}_{d,a}: \begin{cases} \begin{pmatrix} \dot{\boldsymbol{x}}_d(t) \\ y_d(t) \end{pmatrix} = \begin{pmatrix} \mathbf{A}_n & \mathbf{B}_n \\ \mathbf{C}_n & \mathbf{D}_n \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_d(t) \\ u_d(t) \end{pmatrix} + \mathbf{B}_\Delta \boldsymbol{u}_\Delta(t) \\ \boldsymbol{y}_\Delta(t) = \mathbf{C}_\Delta \begin{pmatrix} \boldsymbol{x}_d(t) \\ u_d(t) \end{pmatrix} + \mathbf{D}_\Delta \boldsymbol{u}_\Delta(t) \end{cases} \tag{2.25}$$

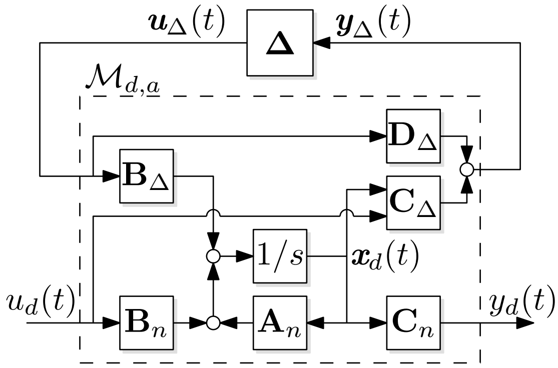


Figure 2.10: LFT representation of the perturbed model $\mathcal{M}_{d,p}$, with $u_d(t) = F_n(t)$ and $y_d(t) = \omega_d(t)$.

The matrices $\mathbf{B}_\Delta$, $\mathbf{C}_\Delta$ and $\mathbf{D}_\Delta$ follow from substituting $\boldsymbol{u}_\Delta(t) = \boldsymbol{\Delta}\boldsymbol{y}_\Delta(t)$ in (2.25), and combining the result with the original perturbed model (2.24). This yields

$$\mathcal{M}_{d,a} : \begin{cases} \dot{\boldsymbol{x}}_d(t) &= \mathbf{A}_n\boldsymbol{x}_d(t) + \mathbf{B}_a\boldsymbol{u}_a(t) \\ \boldsymbol{y}_a(t) &= \mathbf{C}_a\boldsymbol{x}_d(t) + \mathbf{D}_a\boldsymbol{u}_a(t) \end{cases} \tag{2.26a}$$

where

$$\mathbf{B}_a = \begin{pmatrix} -\frac{s_1}{J_e} & -\frac{\xi_{1,n}}{J_e} & \\ \frac{s_1}{J_d} & \frac{\xi_{1,n}}{J_d} & \mathbf{B}_n \\ 0 & 0 & \\ 0 & 0 & \end{pmatrix}, \quad \mathbf{C}_a = \begin{pmatrix} \xi_{2,n} & -\xi_{2,n} & 0 & 0 \\ s_2 & -s_2 & 0 & 0 \\ & \mathbf{C}_n & & \end{pmatrix}$$
$$\mathbf{D}_a = \begin{pmatrix} 0 & 1 & -\frac{\omega_{sl,max}}{F_{n,max}}\xi_{2,n} + \omega_{sl,max} \\ 0 & 0 & -\frac{\omega_{sl,max}}{F_{n,max}}s_2 \\ 0 & 0 & \mathbf{D}_n \end{pmatrix} \tag{2.26b}$$

Notice the element $\mathbf{D}_{a,(1,2)}$, which is required to generate the product of the two uncertain parameters in the original matrices (2.24). In Figure 2.10, the LFT representation of the perturbed model $\mathcal{M}_{d,p}$ consisting of the augmented model $\mathcal{M}_{d,a}$ and the uncertainty matrix $\boldsymbol{\Delta}$ is shown schematically.

2.4.4 Performance demands

The desired performance of the closed-loop system depends on the corresopnding reference signal $\omega_{d,d}(t)$. The closed-loop sensitivity $S(s)$ is an indicator for this performance as it represents the transfer from $r(t) = \omega_{d,d}(t)$ to the error signal $y_K(t) = r(t) - y_d(t) = \omega_{d,d}(t) - \omega_d(t)$. Using the $\mathcal{H}_\infty$ loop shaping concept, the desired closed-loop performance of the controlled system is prescribed by the design of an output-weighting performance

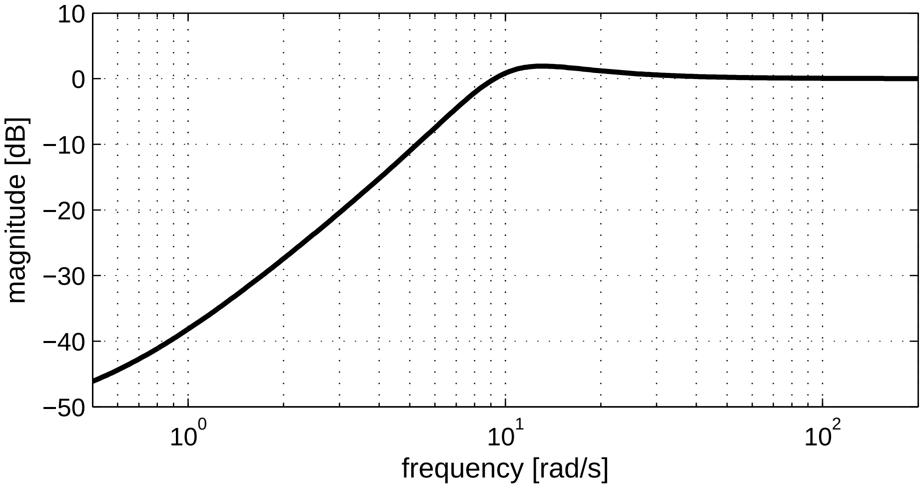


Figure 2.11: Bode magnitude plot of $W_p(j\omega)^{-1}$.

filter $W_p(s)$, yielding the performance requirement

$$|S(j\omega)| < \frac{1}{|W_p(j\omega)|}, \qquad \forall\omega \tag{2.27}$$

where $W_p(s)$ is a second-order low-pass filter to ensure low-frequent tracking of $\omega_{d,d}(t)$ and high-frequent noise suppression

$$W_p(s) = \frac{s^2 + 0.7 f_{bw} s + f_{bw}^2}{(s + 0.05 f_{bw})^2} \tag{2.28}$$

where f_{bw} the desired closed-loop bandwidth frequency. In Figure 2.11, the magnitude plot of $W_p(s)^{-1}$ is shown.

Incorporating $W_p(s)$ in the augmented model $\mathcal{M}_{d,a}$, yields the model $\mathcal{M}^*_{d,a}$ with the input vector $\boldsymbol{u}(t) = (\boldsymbol{u}_\Delta(t), r(t), u_K(t))^T$ and the output vector $\boldsymbol{y}(t) = (\boldsymbol{y}_\Delta(t), z_p(t), y_K(t))^T$, where $u_K(t) = u_d(t)$ and, with slight abuse of notation, $z_p(t) = W_p(s) y_K(t)$. Combining this result with the controller $K_d(s)$ yields the closed-loop problem setup, which is schematically shown in Figure 2.12. The goal is to design a controller $K_d(s)$, which ensures robust stability as well as robust performance for the closed-loop system, while minimizing the transfer from $r(t)$ to $z_p(t)$.

2.4.5 Robust performance and stability analysis

The matrix $\boldsymbol{\Delta}$ represents a structured uncertainty incorporating real perturbations only. Define the set of block-diagonal matrices $\mathcal{D}$ whose structure is compatible to the structure of $\boldsymbol{\Delta}$. Utilizing scalings $D(s) \in \mathcal{D}$, a μ-analysis provides the least conservative robust

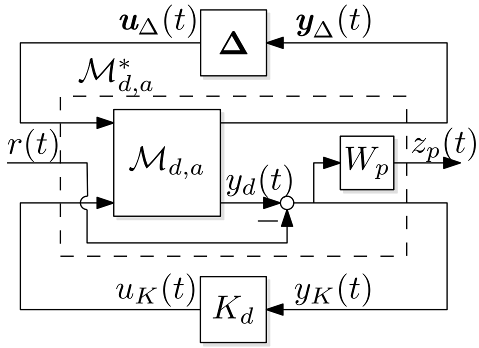


Figure 2.12: Schematic representation of the control problem setup for the controller $K_d = K_d(s)$ with input $y_K(t) = r(t) - y_d(t)$ and output $u_K(t) = F_n(t)$. The augmented plant model $\mathcal{M}^*_{d,a}$ combines the model $\mathcal{M}_{d,a}$ and the performance filter $W_p = W_p(s)$, with input $r = \omega_{d,d}(t)$ and output $z_p(t) = W_p(s)y_K(t)$.

performance and stability conditions (Skogestad and Postlethwaite, 2005). The scalings reduce the conservativeness of the conditions by exploiting the fact that $\boldsymbol{\Delta}$ incorporates real perturbations only.

Transforming the state-space representation $\mathcal{M}_{d,a}$ to a transfer function $H_{d,a}(s)$, yields

$$\begin{aligned} H_{d,a}(s) &= \mathbf{C}_a \left(s\mathbf{I} - \mathbf{A}_n\right)^{-1} \mathbf{B}_a + \mathbf{D}_a \\ &= \left(\begin{array}{c|c} H_{d,a,11}(s) & H_{d,a,12}(s) \\ \hline H_{d,a,21}(s) & H_{d,a,22}(s) \end{array}\right) \end{aligned} \tag{2.29}$$

the augmented model $H^*_{d,a}(s)$ incorporating the performance filter $W_p(s)$ then becomes

$$H^*_{d,a}(s) = \left(\begin{array}{cc|c} H_{d,a,11}(s) & \mathbf{0} & H_{d,a,12}(s) \\ -W_p(s)H_{d,a,21}(s) & W_p(s) & -W_p(s)H_{d,a,22}(s) \\ \hline -H_{d,a,21}(s) & 1 & -H_{d,a,22}(s) \end{array}\right) \tag{2.30}$$

Given a robust controller $K_d(s)$, define the lower LFT of $H^*_{d,a}(s)$ and $K_d(s)$ by

$$N(s) = \mathcal{F}(H^*_{d,a}(s), K_d(s)) \tag{2.31a}$$

$$= H^*_{d,a,11}(s) + H^*_{d,a,12}(s)K_d(s)\left(1 - H^*_{d,a,22}(s)K_d(s)\right)^{-1} H^*_{d,a,21}(s) \tag{2.31b}$$

Subsequently, if nominal stability of the model $N(s, K_d(s))$ is guaranteed, robust performance is achieved for

$$\mu_p(N(j\omega, K_d(j\omega))) < 1, \qquad \forall\omega \tag{2.32}$$

where

$$\mu_p(N(j\omega, K_d(j\omega))) \leq \min_{D_p(j\omega)\in\mathcal{D}} \overline{\sigma}\left(D_p(j\omega)N(j\omega, K_d(j\omega))D_p(j\omega)^{-1}\right), \quad \forall\omega \tag{2.33}$$

defines the performance of the system via the upper bound on the scaled singular value of $N(s, K_d(s))$.

Robust stability is achieved if nominal stability is guaranteed, and

$$\mu_s(M(j\omega)) < 1, \qquad \forall\omega \tag{2.34}$$

where $M(j\omega) = N_{(1,1)}(j\omega)$ is the transfer function from the output to the input of the perturbation matrix $\boldsymbol{\Delta}$, and μ_s is defined by

$$\mu_s(M(j\omega)) \leq \inf_{D_s(j\omega)\in\mathcal{D}} \overline{\sigma}\left(D_s(j\omega)M(j\omega)D_s(j\omega)^{-1}\right), \qquad \forall\omega \tag{2.35}$$

which forms a generalization of the upper bound on the scaled structured uncertainty $\boldsymbol{\Delta}$, with scalings $D_s(s) \in \mathcal{D}$.

2.4.6 DK-iteration

A robust controller is designed using a DK-iteration procedure, combining $\mathcal{H}_\infty$-synthesis and μ-analysis (Skogestad and Postlethwaite, 2005). The DK-iteration procedure involves a sequence of minimizations, alternating between the controller $K_d(s)$, for fixed scalings $D_p(s)$ associated with the scaled upper bound μ_p (2.33), and the scalings $D_p(s)$, for a fixed controller $K_d(s)$.

1. For a given (initial) controller $K_d(j\omega)$, compute the scaling $D_p(j\omega) \in \mathcal{D}$ minimizing $\mu_p(N(j\omega), K_d(j\omega))$, for all ω.
2. Fit the magnitude of each element of $D_p(j\omega)$ to a stable, minimum-phase transfer function $D_p^*(j\omega)$.
3. For fixed $D_p^*(j\omega)$, synthesize a $\mathcal{H}_\infty$-controller $K_d(j\omega)$ for the scaled problem

$$\min_{K_d(j\omega)} \left(||D_p^*(j\omega)N(j\omega, K_d(j\omega))D_p^*(j\omega)^{-1}||_\infty\right), \qquad \forall\omega \tag{2.36}$$

Iterating continues until either satisfactory performance μ_p is achieved or the $\mathcal{H}_\infty$-norm no longer decreases. As the iteration procedure may converge to a local minimum, the choice for the initial conditions on the controller $K_d(s)$ is important. In this case, an initial, stable design for $K_d(s)$ is based on insight in the system and standard loop shaping techniques.

2.5 Results

2.5.1 Controller evaluation

The results of the controller synthesis are discussed in this section. As discussed in Section 2.3.2, the additional actuator and sensor dynamics as well as the communication delay are taken into account in the results that are shown. In Figure 2.13, the values of $|\mu_p(j\omega)|$ and $|\mu_s(j\omega)|$ are shown. Given the fact that nominal stability of $N(j\omega)$ is guaranteed, $\forall\,\omega$, robust stability of the closed-loop system is achieved, as $|\mu_s(j\omega)| < 0$ dB holds (see Section 2.4.5).

The desired performance, however, may not be met as $|\mu_p(j\omega)| > 0$ dB. This is confirmed by the magnitude of the sensitivity function $S(s)$ for variable ξ_1 and ξ_2, which is shown in Figure 2.14. In general, the characteristics of the closed-loop sensitivity follow the desired characteristics of the performance weighting relatively well. However, the performance demand (2.27) is not met, which indicates that the desired performance is not achieved.

2.5.2 Simulation results

In Figure 2.15, simulation results of the new control concept and the original implementation are shown. A drive-off maneuver of a 40000 kg truck on a flat road is simulated. In the original implementation, an open-loop feedforward controller, steering the clamping force $F_n(t)$ is used. In practice, this is a built-in controller of the AMT. During a drive-off maneuver, the clamping force $F_n(t)$ is increased via a predefined trajectory. For the new control concept, a reference trajectory $\omega_{d,d}(t)$ is used. In the simulation with the original

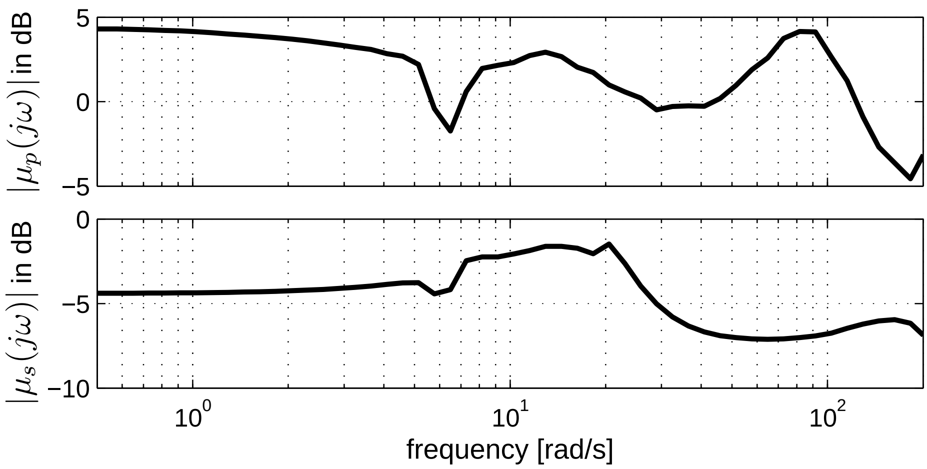


Figure 2.13: Bode magnitude plots of $\mu_p(j\omega)$ (upper plot) and $\mu_s(j\omega)$ (lower plot).

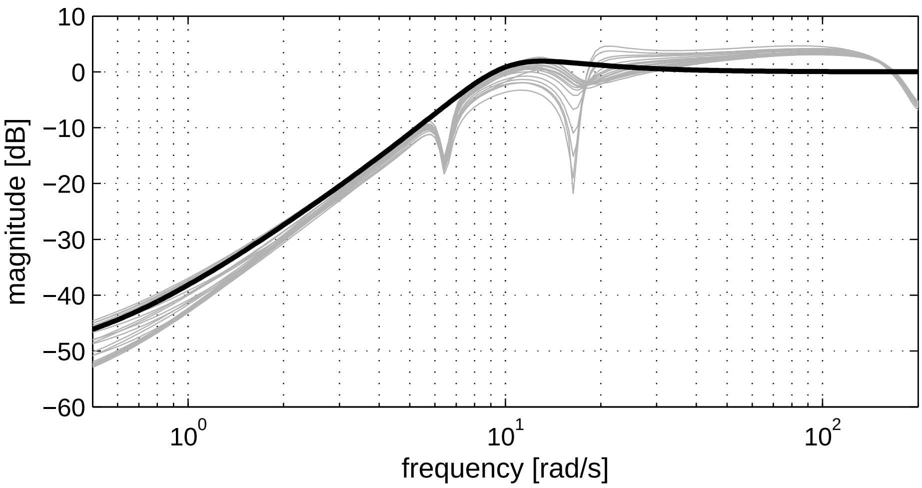


Figure 2.14: Bode magnitude plots of the performance weighting $W_p(j\omega)^{-1}$ (black) and the sensitivity $S(j\omega)$ for variable ξ_1 and ξ_2 (grey).

implementation, a feedforward control signal is used that ensures closing of the clutch in a time that corresponds to the time this reference trajectory takes.

In the simulations both $F_n(t)$ and $\omega_d(t)$ follow the corresponding reference trajectories almost exactly. Therefore, the reference trajectories are not included in Figure 2.15. Tracking becomes worse for faster drive-off situations. Figure 2.14 already indicates that the desired closed-loop bandwidth is not achieved for all situations. For faster drive-off situations, the required closed-loop bandwidth becomes higher. Eventually, this leads to tracking problems, in the sense that the reference trajectory $\omega_{d,d}(t)$ cannot be followed exactly, or that overshoot is present. Furthermore, in both cases a relatively simple feedback controller $K_e(s)$ is used to control the engine rotational velocity, using a constant reference trajectory of 550 rpm.

The results indicate the presence of judder in the original implementation, whereas oscillations are successfully prevented by the new concept. The feedforward controller in the original implementation does not prevent initiation of the judder-induced oscillations, which amplify until clutch lockup is reached, i.e., when $\omega_d(t) = \omega_e(t)$ holds. By stabilizing the system, the feedback controller of the new control concept actually prevents initiation of the oscillations. Hence, as no other disturbances are modeled and a smooth reference trajectory is used, judder-induced oscillations are effectively prohibited. As a result, the frequency content of the control signal $F_n(t)$ is small around the eigenfrequency of the driveline. The step change in the clamping force $F_n(t)$ at about 16 s in the simulation results with the new control concept is a result of switching both the model and the controller that are used in the simulation at the moment of clutch lockup.

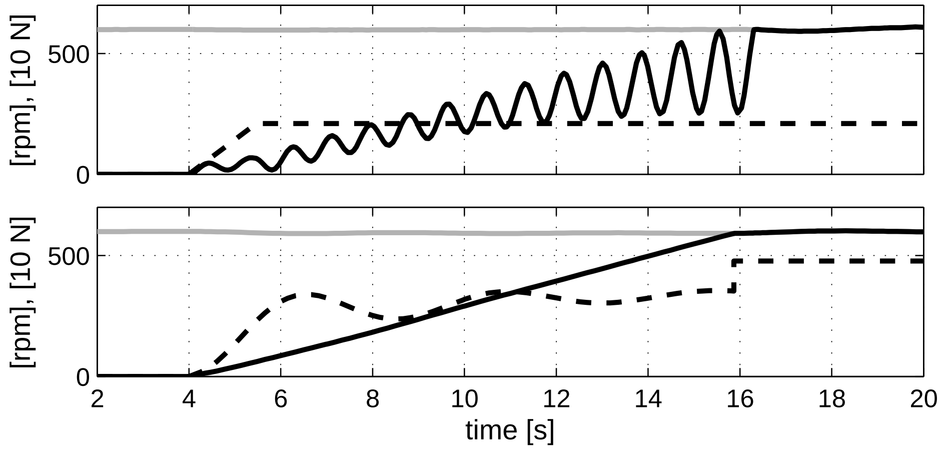


Figure 2.15: Simulation results of the new control concept (lower figure) in comparison with the original implementation (upper figure), with $\omega_d(t)$ in solid black, $F_n(t)$ in dashed black and $\omega_e(t)$ in solid grey.

2.5.3 HIL experiments

A hardware-in-the-loop (HIL) setup combines the electronic and software components of a real truck, such as the engine CPU, the clutch actuator, and a simulation model of the truck hardware, such as the driveline, the chassis and the truck body (see Figure 2.16). In this way, an HIL setup allows to test new actuators, sensors or control concepts, without the need to perform tests in a real truck (Huisman and Veldpaus, 2005). Hence, HIL experiments provide useful insight before a new system is implemented in a real truck. Using a dSpace MicroAutoBox, the new control concept is implemented and tested on the DAF HIL setup. The models of the clutch system and the driveline are included in the corresponding simulation model. Focus is on the working of the concept when actual actuators, sensors and CAN bus communication are used.

In Figure 2.17, the results of HIL experiments with the new control concept and the original implementation are shown. Again, a drive-off maneuver of a 40000 kg truck on a flat road is simulated. In this case, the reference trajectories $\omega_{e,d}(t)$ and $\omega_{d,d}(t)$ for the engine rotational velocity and the driveline rotational velocity, respectively, are included (see the lower part of Figure 2.17). The reference trajectories for $F_n(t)$ and $\omega_e(t)$ in the original implementation follow from the AMT and are unknown. This also holds also for $F_n(t)$ in both the original implementation and the for the new concept from the moment clutch lockup is reached. At that moment, the engine takes over control of $\omega_e(t) = \omega_d(t)$, and the (original) AMT controller prescribes a clamping force $F_n(t)$.

In the new control concept, $\omega_e(t)$ shows a step change when the drive-off maneuver is started. This is a result of a mismatch between the desired engine idle velocity during

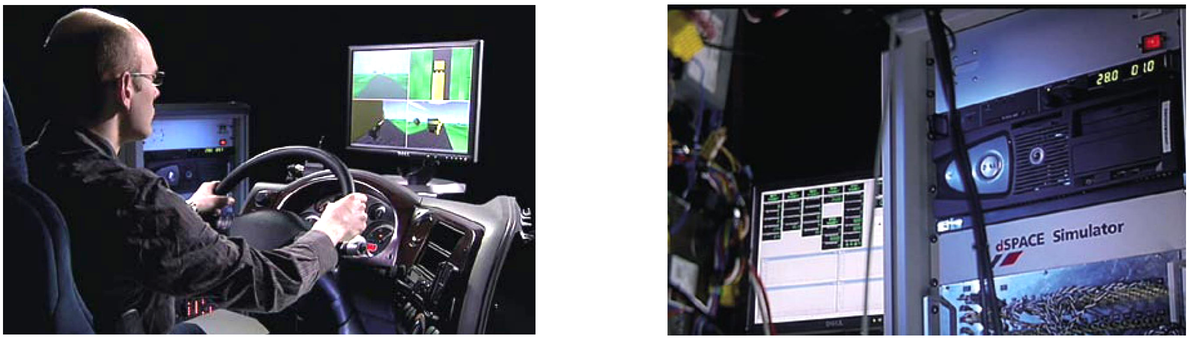


Figure 2.16: Illustration of the DAF HIL setup.

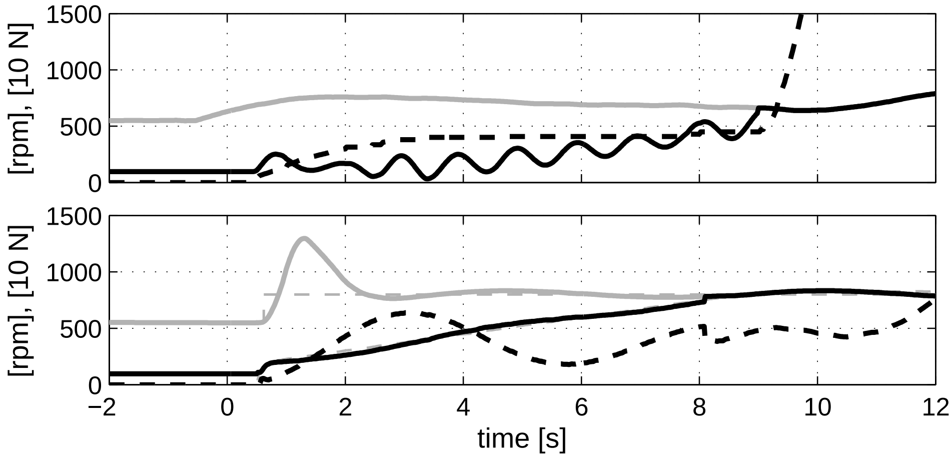


Figure 2.17: HIL measurement results of the new control concept (lower figure) in comparison with the original implementation (upper figure), with ω_d in solid black, F_n in dashed black, ω_e in solid grey, and both $\omega_{e,d}(t)$ and $\omega_{d,d}(t)$ in dashed grey.

driving-off, which equals 800 rpm, and the initial condition used in the simulation model that is implemented in the HIL setup, which equals 550 rpm. The feedback controller $K_e(s)$ is not able to handle this step change appropriately, resulting in a peak in $\omega_e(t)$.

Furthermore, in comparison to the simulation results (see Figure 2.15), the time in which clutch-lockup is reached, is almost twice as small. Moreover, the final setpoint for the rotational velocity $\omega_d(t)$ is increased to 800 rpm. As a result, in the resulting tracking performance, small deviations from the reference trajectory $\omega_{d,d}(t)$ can be observed, which are not present in the simulations. However, corresponding to the simulation results, the HIL experiments show that judder-induced oscillations are present in the original implementation, whereas oscillations are successfully prevented by the new concept. This validates the design of the new concept.

2.6 Conclusions and recommendations

A model of an HDV driveline is designed and validated using measurements obtained with a real HDV driveline. The model incorporates an uncertainty model for unmodeled friction dynamics, which induce instabilities. These instabilities are the cause of undesired driveline oscillations, which are referred to as clutch judder. Accordingly, a robustly stable feedback controller is designed, stabilizing the dynamics for the entire operating range. Simulations and HIL experiments show that the controller is indeed successful in suppressing judder-induced driveline oscillations, validating the feasibility of the new control concept. Accordingly, the implementation on a real vehicle is an issue for future work.

The desired performance is not achieved for the complete envelope of working conditions, which can be related to the limited closed-loop actuator bandwidth and the communication delay in the system. Given the parameter uncertainties, it will be difficult to improve the closed-loop performance such that it meets the current performance requirement. One of the solutions is to focus on the definition of a more feasible requirement. For example, a working-condition dependent requirement might enable improvement of the closed-loop performance for specific working conditions using a linear parameter varying (LPV) controller design approach. Another solution is to investigate whether either the closed-loop dynamics of the actuator can be improved or the communication delay can be reduced.

CHAPTER 3

Gain scheduling and linear parameter varying control design for heavy-duty vehicle cruise control

---

Abstract - This chapter presents the comparison of classical gain scheduling (GS) and more recent linear parameter varying (LPV) controller synthesis methods for the design of a heavy-duty vehicle cruise control. Focus is on exposing limitations of the classical GS methods that are often applied in practice, and assessing the practical applicability of more recent LPV methods. Classical GS methods include various non-automated design steps, whereas with more recent LPV methods a controller can be generated automatically. For heavy duty vehicles, the loading of a vehicle typically varies in a large range, influencing the vehicle dynamics. The loading is constant during operation, but can vary in a continuous range in between operations. As the loading is measured online, these variations can explicitly be taken into account in the design of the cruise control via a GS or LPV controller synthesis. Using the cruise control design as an application example, four GS and LPV controller synthesis methods are compared. A DAF XF105 truck is used for experimental validation. The results indicate that application of classical GS techniques is complicated by the lack of an explicit performance measure for the non-automated design steps in the methods. For more recent LPV methods, the increasing theoretical complexity can limit practical implementation, as, first, the computational burden increases, and, second, insight in the resulting controller decreases.

3.1 Introduction

Control design and tuning techniques applied in industry often differ from state-of-the-art techniques coming from the academic world. In practice, well-known and proven classical techniques are often preferred over more recent techniques, which are often theoretically more complex. The automotive industry is a typical example where, traditionally, focus is on mechanical, hydraulic and in the case of heavy duty vehicles also pneumatic solutions. However, it is estimated that today more than 80 percent of all automotive innovations stem from electronics and software (Leen and Heffernan, 2002; Grimm, 2003). In practice, online tuning techniques are often employed, which are time-consuming and lack both stability and performance guarantees. To reduce tuning effort and to enable a-priori guarantees, the need for application of more recent control solutions becomes increasingly important.

Gain scheduling (GS) and linear parameter varying (LPV) controller synthesis methods provide powerful approaches to handle a wide class of nonlinear systems and systems with measurable environmental or parameter time variations. In comparison with robust controller synthesis, in which possible variations in the dynamics are regarded as unknown uncertainties, the result is, in general, less conservative, enabling increased performance. As a result, GS and LPV controller synthesis methods have received much attention in the last two decades and a wide range of approaches has been presented in literature (Rugh and Shamma, 2000; Leith and Leithead, 2000).

In practice, classical gain-scheduling techniques are often applied, including various non-automated design steps and lacking closed-loop stability or performance guarantees. More recent LPV controller synthesis methods are often theoretically more complex, but the controller can be generated automatically. Furthermore, a-priori closed-loop stability guarantees and a-priori specification of the desired performance are possible. In this research, different GS and LPV controller synthesis methods are compared for a practical application. Focus is on exposing the limitations of the classical GS methods that are often applied in practice and assessing the practical applicability of more recent LPV methods.

Cruise control is a widespread, commercially available functionality, which, nowadays, can be regarded as a standard automotive control functionality (Kiencke and Nielsen, 2005). Specifically focusing on heavy duty vehicles (HDVs), the loading of the vehicle typically varies in a large range, influencing the vehicle dynamics. The vehicle loading is constant during operation, but can vary in a continuous range in between operations. For example, the loading of a typical HDV can vary in between 0 and 40000 kg. Recent research advances on active parameter and state estimators enable accurate estimates of the vehicle loading (Kolmanovsky and Winstead, 2006; McIntyre et al., 2009). Moreover, commercially available vehicle control systems such as the electronic braking system (EBS) already provide an estimate of the vehicle mass including the vehicle loading (WABCO, 2010). Hence, in this research, it is assumed that the loading can be estimated or measured online. As a result, the design of cruise control for HDVs is a particularly suitable example to compare the application of different GS and LPV controller synthesis methods for a practical application.

The contribution of this chapter is a comparison of relevant GS and LPV controller synthesis methods for the design of a cruise control for HDVs, targeting to expose the limitations of the classical GS methods that are often applied in practice and to assess the practical applicability of more recent LPV methods.

The controller design is based on identification and validation experiments that are performed with a DAF XF105. For reasons of confidentiality, exact vehicle parameter values are not reported.

In Section 3.2, a concise overview of different GS and LPV controller synthesis methods is given. The cruise control problem and the modeling are presented in Section 3.3. Application of different GS and LPV controller synthesis methods and experimental results with an HDV are discussed in Section 3.4 and 3.5, respectively. The chapter is closed with conclusions in Section 3.6.

3.2 Gain scheduling and linear parameter varying controller synthesis methods

In this section, an overview of different gain scheduling (GS) and linear parameter varying (LPV) controller synthesis methods is given. Focus is on control problems with a single measured scheduling parameter, which is constant during operation, but which can vary in a continuous range in between operations. Accordingly, the scheduling parameter is defined as:

$$\delta(t) = \delta \in \Delta \tag{3.1}$$

where $\Delta \in \mathbb{R}$ the continuous operating range for δ. Hence, a time-invariant system is considered, although the relevant dynamics are influenced by $\delta \in \Delta$. As δ is measured online, GS or LPV controller synthesis methods are applicable to take this influence into account.

GS and LPV controller synthesis methods can be classified into three main approaches: classical GS, more recent LPV techniques, and fuzzy scheduling. Fuzzy scheduling is often regarded as an extension of classical GS, specifically targeting robustness for fast parameter variations (Guerra and Vermeiren, 2001). As focus is on a constant scheduling parameter, fuzzy scheduling methods will not be discussed further in this research. Based on an overview of classical GS and more recent LPV techniques (Sections 3.2.1 and 3.2.2), different methods are selected for comparison (Section 3.2.3).

3.2.1 Classical gain scheduling

Classical gain scheduling approaches are based on a set of discretely-parameterized controllers $\mathcal{K}_o(\delta)$, $\delta \in \Delta_O$, corresponding to a finite grid of operating points $\Delta_O \subset \Delta$. Via interpolation, the controllers are embedded into a continuously parameterized family of controllers $\mathcal{K}(\delta)$, $\delta \in \Delta$ (Rugh and Shamma, 2000; Leith and Leithead, 2000).

Well-established, linear control design techniques can be adopted to synthesize $\mathcal{K}_o(\delta)$, $\delta \in \Delta_O$, and parameter-dependent modeling may even be omitted, adopting scheduling on the basis of insight and experience. As a result, classical GS is easily and often applied in practice. In this research, classical GS using both manual loop shaping and $\mathcal{H}_\infty$ controller synthesis techniques are considered.

For the interpolation, the set of controllers $\mathcal{K}_o(\delta)$, $\delta \in \Delta_O$, should be of a fixed structure and preferably of a limited order and complexity. Various non-automated interpolation approaches based on insight and experience, as well as more theoretically justified methods are reported in literature, e.g., (Paijmans et al., 2006; Nichols et al., 1993; Wijnheijmer et al., 2006) and (Stilwell and Rugh, 2000; Chang and Rasmussen, 2008; Claveau et al., 2007), respectively. However, a generally valid, systematic approach for the interpolation lacks and many pitfalls are present (Tóth et al., 2007; Leith and Leithead, 2000; Rugh and Shamma, 2000). To emphasize the differences between classical GS and more recent LPV methods, a non-automated interpolation methods is adopted in this research, which is based on insight.

A disadvantage of classical GS is the lack of a-priori closed-loop stability and performance guarantees. Possibilities to analyze stability afterwards are, for example, provided by Lyapunov stability analysis or robust μ analysis. The latter approach provides a suitable framework to guarantee stability in the case of constant scheduling parameters (Khalil, 2002; Skogestad and Postlethwaite, 2005). In this research, both methods are addressed. Closed-loop performance is, in general, validated by means of extensive simulations and tests (Rugh and Shamma, 2000).

3.2.2 LPV controller synthesis

Linear parameter varying (LPV) controller synthesis methods target a systematic design approach to obtain LPV controllers with a-priori closed-loop stability guarantees and the possibility to specify a-priori desired closed-loop performance. The basis of the controller synthesis is a robust performance specification, combining an asymptotical stability specification and a performance specification. For stable linear time-invariant systems, the $\mathcal{H}_\infty$ norm of the system model transfer function is often adopted as a performance measure, yielding an extended $\mathcal{H}_\infty$ problem specification (Rugh and Shamma, 2000; Scherer et al., 1997).

The basis for the robust performance specification is an LPV model of the generalized plant $\mathcal{P}(\delta)$, $\delta \in \Delta$, combining a model of the plant at hand and performance requirements, where $\mathcal{P}(\delta)$ is defined as:

$$\begin{pmatrix} \dot{\boldsymbol{\xi}}(t) \\ \boldsymbol{z}_p(t) \\ \boldsymbol{y}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{A}(\delta) & \mathbf{B}_w(\delta) & \mathbf{B}_u(\delta) \\ \mathbf{C}_z(\delta) & \mathbf{D}_{zw}(\delta) & \mathbf{D}_{zu}(\delta) \\ \mathbf{C}_y(\delta) & \mathbf{D}_{yw}(\delta) & \mathbf{D}_{yu}(\delta) \end{pmatrix} \begin{pmatrix} \boldsymbol{\xi}(t) \\ \boldsymbol{w}_p(t) \\ \boldsymbol{u}(t) \end{pmatrix} \tag{3.2}$$

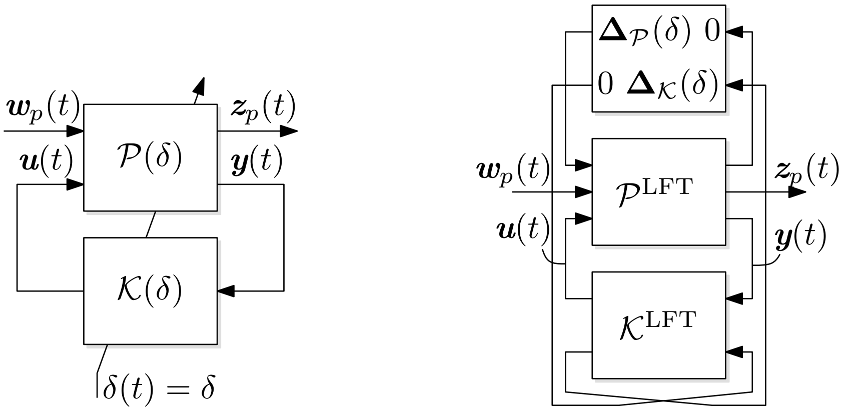
with the state $\boldsymbol{\xi}(t) \in \mathbb{R}^{n_\xi}$, the control input $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$, the model output $\boldsymbol{y}(t) \in \mathbb{R}^{n_y}$, the performance channel $\boldsymbol{w}_p(t) \mapsto \boldsymbol{z}_p(t)$, n_ξ, n_u and n_y the size of the state vector, the input and the output, respectively, and $\mathbf{A}$, $\mathbf{B}_j$, $\mathbf{C}_i$, and $\mathbf{D}_{ij}$, $j \in \{w, u\}$, $i \in \{z, y\}$, are the corresponding state-space matrices (Skogestad and Postlethwaite, 2005). A schematic representation of the closed-loop system with $\mathcal{P}(\delta)$ (3.2) and a corresponding LPV controller $\mathcal{K}(\delta)$, $\delta \in \Delta$, is shown in Figure 3.1(a). The design of SISO as well as MIMO LPV models has received much attention, and is still an open field of research (De Caigny et al., 2009; Paijmans et al., 2008; Marcos and Balas, 2004).

The robust performance specification leads to a set of matrix inequality constraints. The controller synthesis comes down to finding a (parameter-dependent) Lyapunov function $\mathbf{X}(\delta) \in \mathbb{R}^{n_\xi \times n_\xi}$ and a parameter-dependent controller $\mathcal{K}(\delta) \in \mathbb{R}^{(n_\xi + n_y) \times (n_\xi + n_u)}$, $\delta \in \Delta$, solving these constraints. Three problems are encountered. First, the constraints are nonlinear in the decision variables $\mathcal{K}(\delta)$ and $\mathbf{X}(\delta)$. To arrive at a set of linear matrix inequalities (LMIs) forming a convex optimization problem, a nonlinear change of variables is required (Scherer et al., 1997; Masubuchi et al., 1995),

$$(\mathbf{X}(\delta), \mathcal{K}(\delta)) \mapsto (\mathbf{X}'(\delta), \mathcal{K}'(\delta)) \tag{3.3}$$

introducing the new decision variables $\mathbf{X}'(\delta)$ and $\mathcal{K}'(\delta)$. Second, a continuous parameter range $\delta \in \Delta$ is considered, which gives infinitely many constraints. Third, the δ-parameterization of the decision variables is unknown, yielding infinitely many solutions.

To handle the latter two problems, two frameworks have been developed. If only a single or a few scheduling parameters are present, the LPV framework is appropriate. Either the operating range is gridded and a solution is computed for a grid $\Delta_O \subset \Delta$ (Becker and Packard, 1994; Apkarian and Gahinet, 1995; Wu et al., 1996), or affine or polytopic parameter dependencies are assumed for the plant, the controller, and the Lyapunov function, computing a solution at the corner points of a corresponding convex polytope $\Delta_c \supseteq \Delta$ (Apkarian et al., 1995; Yu and Sideris, 1995; De Caigny et al., 2010). If many scheduling parameters are present, both gridding and assuming affine parameter dependencies are less suitable, as the corresponding operating spaces are of high order, and, in general, Δ_c is an overestimate of Δ. In this research, the gridding approach is considered, which can be regarded as the most generic, widely adopted LPV controller synthesis method.



(a) (b)

Figure 3.1: (a) Closed-loop LPV system, and (b) LFT representation of a closed-loop LPV system, with the performance channel $w_p(t) \mapsto z_p(t)$, the control input $u(t)$, the model output $y(t)$, and $\mathcal{P}(\delta)$ and $\mathcal{K}(\delta)$, $\delta \in \Delta$ a parameter-dependent plant model and controller, respectively. $\mathcal{P}^{\mathrm{LFT}}$ and $\mathcal{K}^{\mathrm{LFT}}$ are the corresponding LFT representations, where $\Delta_{\mathcal{P}}(\delta)$ and $\Delta_{\mathcal{K}}(\delta)$ are structured uncertainty matrices, representing the parameter dependencies.

The linear fractional transformation (LFT) framework, using a small-gain analysis (see Figure 3.1(b)), is appropriate for many (as well as for only a single or a few) scheduling parameters. However, rational (and affine) parameter-dependencies are assumed and a constant, parameter-independent Lyapunov function is used, guaranteeing robustness for infinitely fast parameter variations (Scherer, 2001).

Research on combinations of the LFT framework and parameter-dependent Lyapunov functions targets the use of the well-established small-gain problem definition for problems with bounded parameter variation rates. Examples are the use of LFT Lyapunov functions (Wu and Dong, 2006; Helmersson, 1996), and the use of an extension of the Kalman-Yakubovich-Popov (KYP) lemma (Dinh et al., 2005; Iwasaki and Hara, 2005). The latter specifically focuses on constant scheduling parameters and provides a-priori stability and performance guarantees. Consequently, this approach is considered as the final approach for evaluation in this research.

Characterizing LPV problems based on i) the type of parameter-dependency, ii) the number of scheduling parameters, and iii) the parameter variation rate, either the LPV or the LFT framework, or research on combinations of both frameworks is applicable. The main disadvantages of the LPV and LFT frameworks are the often rapidly increasing computational burden for more complex models, a non-intuitive synthesis which leaves only little room for manual fine-tuning afterwards, and the introduction of conservatism in the controller synthesis. As a result, the number of applications that is reported in literature is limited (e.g., Hingwe et al., 2002; Dettori and Scherer, 2002; Groot Wassink et al., 2004).

3.2.3 Problem formulation

In this research, various GS and LPV controller synthesis methods are compared. Based on the overview of the previous sections, methods varying from methods that are merely based on non-automated design steps to fully automated methods are compared:

A **non-automated:** classical gain scheduling using manual loop shaping ($\mathcal{K}_{\mathrm{LS}}(\delta)$),

B **semi-automated:** classical gain scheduling using $\mathcal{H}_\infty$ controller synthesis ($\mathcal{K}_{\mathcal{H}_\infty}(\delta)$),

C **automated, limited guarantees:** LPV controller synthesis using an extended $\mathcal{H}_\infty$ problem definition and gridding ($\mathcal{K}_{\mathrm{LPV}}(\delta)$),

D **automated, full guarantees:** LPV controller synthesis using an extended $\mathcal{H}_\infty$ problem definition and an extension of the KYP-lemma ($\mathcal{K}_{\mathrm{KYP}}(\delta)$).

Method A is often applied in practice. In Method B, the use of $\mathcal{H}_\infty$ techniques facilitates automatic synthesis of a set of operating-point dependent controllers. Accordingly, Method B forms the step from classical GS to more recent LPV methods using the extended $\mathcal{H}_\infty$ problem definition. Method C is the most generic, widely adopted LPV controller synthesis method. Method D is a dedicated method for LPV problems with a single variable but constant scheduling parameter, targeting to minimize conservatism in the controller synthesis.

3.3 Problem setup and modeling

3.3.1 System overview

A standard cruise control system for a typical heavy duty vehicle (HDV) with an automated manual transmission (AMT) is considered. The engine is the actuator, enabling automatic propulsion, with as input a fuel request $f_{e,d}(t)$ in mg/stroke. A tachograph at the output of the gearbox measures the vehicle velocity $v(t)$ in m/s. Furthermore, resistance forces $F_r(t)$ are acting on the vehicle. In Figure 3.2, a schematic representation of a typical HDV with the relevant elements for cruise control is shown.

Variations in both the vehicle loading and the actual gear ratio influence the relevant dynamics for cruise control. To account for variations in the loading, a gain scheduling (GS) or linear parameter varying (LPV) control design approach is pursued using the measured vehicle mass as a scheduling parameter. To account for variations in the gear ratio, a controller is designed per gear ratio. The number of gears is limited, and switching between gears is accompanied by opening of the driveline via the clutch, which, in effect, means that the cruise control functionality is switched on and off. Hence, focus is on deriving a

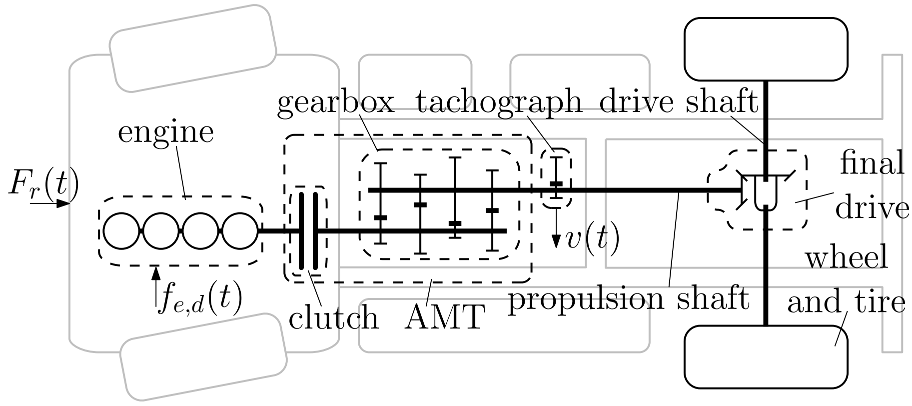


Figure 3.2: Schematic representation of the relevant elements for cruise control in a typical HDV with an automated manual transmission (AMT), where $f_{e,d}(t)$ is the fuel request, $v(t)$ is the measured vehicle velocity, and $F_r(t)$ are the resistance forces.

set of gear-ratio dependent LPV models $\mathcal{P}_{\mathrm{HDV}}(\delta) : f_{e,d}(t) \mapsto v(t)$, with the vehicle mass as a scheduling parameter δ.

3.3.2 Performance requirements

Focus of this research is on the design of GS and LPV feedback controllers. To assess the performance of these controller designs, their disturbance rejection capabilities are evaluated. For a cruise control system, the main disturbances are changes in the cruise control set velocity and environmental disturbance forces, such as road inclination and the aerodynamic drag force. From practice, only performance requirements related to changes in the cruise control set velocity are available. The absence of performance requirements related to environmental disturbance forces may be related to the fact that these disturbances are difficult to measure, prohibiting quantitative performance evaluation of the controllers. For example, estimators for the road inclination are nowadays available, however, it is no standard functionality in commercially available HDVs (McIntyre et al., 2009). Focus of this research is on comparison of GS and LPV controller synthesis methods using the cruise control problem as a practical application example. Hence, in this research, changes in the cruise control set velocity are considered. For clarity, step changes are considered, although a filtered, more smooth reference signal is probably preferable for implementation in a final application.

The performance requirements that are assumed in this research are listed in Table 3.1. The requirements in Table 3.1 are closely related to practice and can be regarded as generic requirements for a cruise control design for an HDV. The desired closed-loop bandwidth ω_{bw} is defined as the frequency at which the magnitude of the open-loop frequency response function crosses 0 dB in the downwards sense. The settling time is the time it

Table 3.1: Performance requirements.

specification	variable	value	unit
closed-loop bandwidth	ω_{bw}	0.5	rad/s
settling time	t_s	7.0	s
robustness, modulus margin	k_{MM}	6.0	dB
measurement noise	ω_n	> 1.0	rad/s
maximum steady-state error*	e_{ss}	0.15	m/s
maximum overshoot*	o_{max}	0.5	m/s
maximum fuel request	$f_{e,max}$	100.0	mg/stroke

*For a maximum step disturbance of 10.0 m/s.

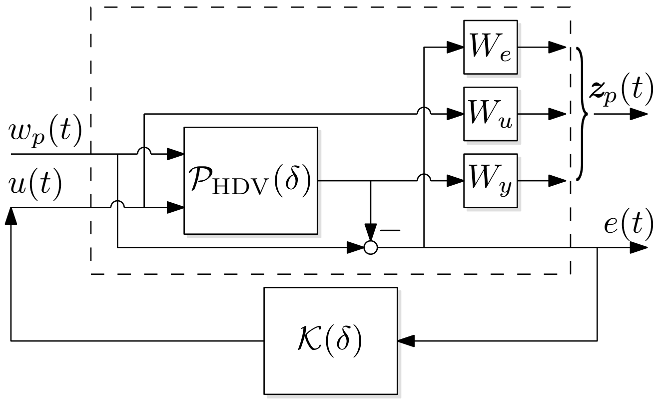


Figure 3.3: Schematic representation of the mixed-sensitivity framework for the cruise control problem specification (Skogestad and Postlethwaite, 2005).

takes for the system transients to decay to a small value, which is within the steady-state error band e_{ss}. A modulus margin k_{MM} is adopted for robustness. The measurement noise frequency ω_n indicates the frequency beyond which the controller should exhibit roll-off to avoid amplification of sensor noise and to ensure a certain level of robustness against high-frequent model uncertainties. Furthermore, the performance requirements are based on a maximum step disturbance of 10.0 m/s, which corresponds to the maximum velocity range that is covered by one gear.

Consider the controller synthesis methods that are compared in this research (see Section 3.2.3). Methods B to D are based on a standard or an extended $\mathcal{H}_\infty$ problem specification. To translate the performance requirements into a suitable measure of performance for these methods, a mixed-sensitivity framework is used and weighting filters $W_e(s)$, $W_u(s)$ and $W_y(s)$ are defined. In Figure 3.3, a schematic representation of the mixed-sensitivity framework is shown (Skogestad and Postlethwaite, 2005), where $w_p(t) = r(t)$ is the desired cruise control velocity, $e(t) = r(t) - v(t)$ is the tracking error, $\boldsymbol{z}_p(t)$ contains the outputs to be minimized, and $u(t) = f_{e,d}(t)$ is the fuel request.

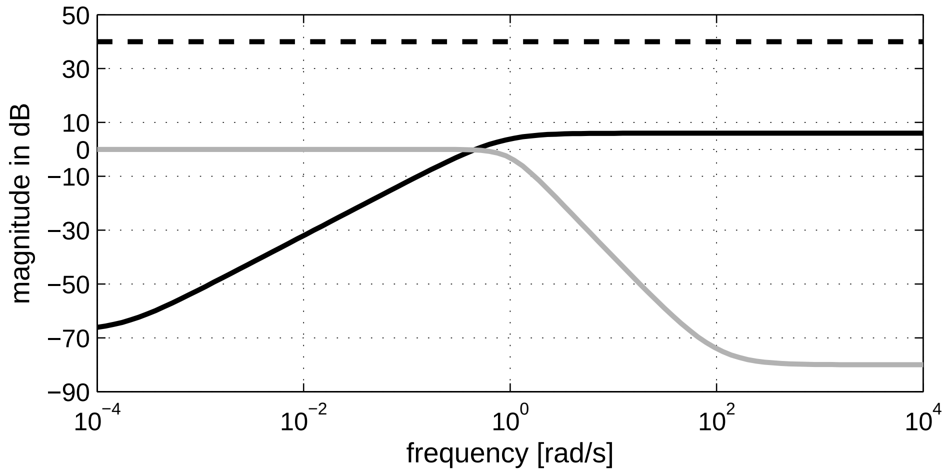


Figure 3.4: Bode magnitude plots of $W_e(j\omega)^{-1}$ (solid black), $W_u(j\omega)^{-1}$ (dashed black), and $W_y(j\omega)^{-1}$ (solid grey) (3.4).

The designs of the weighting filters are based on the performance requirements in Table 3.1, using standard filter designs (see, e.g., Skogestad and Postlethwaite, 2005; Hu et al., 1996):

$$\begin{aligned} W_e(s) &= \frac{s/10^{k_{MM}/20} + \omega_{bw}}{s + \omega_{bw}/e_{ss}} \\ W_u(s) &= f_{e,max}^{-1} \\ W_y(s) &= \frac{\omega_h^2}{\omega_n^2}\frac{s^2 + 2d_0\omega_n s + \omega_n^2}{s^2 + 2\omega_h s + \omega_h^2} \end{aligned} \tag{3.4}$$

where $\omega_h = 10\omega_n$ is introduced to obtain a proper filter $W_y(s)$, and $d_0 = 0.7$ is a damping term to tune the maximum singular value of the complementary sensitivity. Assuming a second-order response, the maximum singular value of the complementary sensitivity function determines the maximum overshoot. In Figure 3.4, Bode magnitude plots of the weighting filters (3.4) are shown.

3.3.3 Modeling of actuator, sensor, and communication network

A schematic overview of the relevant elements for cruise control are shown in Figure 3.5, with $T_e(t)$ the gross engine torque, and $\omega_{eq}(t)$ the gearbox output shaft rotational velocity. Communication of the measured velocity signal $v'(t)$ is done via the built-in controller area network (CAN) of the vehicle, yielding a delayed velocity measurement $v'(t)$ that is available for feedback control.

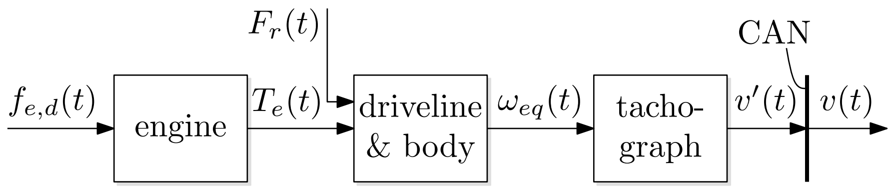


Figure 3.5: Schematic representation of the relevant elements for cruise control in a typical HDV, where $T_e(t)$ is the engine torque, $\omega_{eq}(t)$ is the gearbox output shaft rotational velocity, $v'(t)$ is the tachograph output, and $v(t)$ is the velocity measurement that is available for feedback control.

Based on measurement results with a real HDV, it is concluded that the engine as actuator and the tachograph as sensor have a (closed-loop) bandwidth of about 100 rad/s (Zalm et al., 2008). As this is relatively high with respect to the desired closed-loop bandwidth (see Table 3.1), both the engine and the tachograph are modeled using a first-order low-pass filter with time constant $\tau = 0.01$ s/rad. Furthermore, the engine model includes a gain k_e in Nm/(mg/stroke), transforming the fuel request $f_{e,d}(t)$ into a torque. The result is a first-order model $\mathcal{P}_e : f_{e,d}(t) \mapsto T_e(t)$

$$\mathcal{P}_e : \begin{cases} \dot{x}_e(t) &= -\frac{1}{\tau}x_e(t) + \frac{1}{\tau}f_{e,d}(t) \\ T_e(t) &= k_e x_e(t) \end{cases} \tag{3.5}$$

where $x_e(t)$ is the state of the model.

Analogously, the tachograph model includes a gain r_w/i_f in m/rad, converting the measured rotational velocity of the gearbox output shaft $\omega_{eq}(t)$ into a longitudinal vehicle velocity $v'(t)$, where r_w is the wheel radius and i_f is the ratio of the final drive. Furthermore, a model for the communication delay $\tau_{\text{CAN}} = 0.06$ s representing the CAN-bus communication is included in the tachograph model (see Figure 3.5). Targeting a closed-loop bandwidth of 0.5 rad/s (see Table 3.1), a first-order Padé-approximation is sufficient to model the delay. Hence, the resulting model for the tachograph is given by $\mathcal{P}_t : \omega_{eq}(t) \mapsto v(t)$

$$\mathcal{P}_t : \begin{cases} \dot{\boldsymbol{x}}_t(t) &= \begin{pmatrix} -\frac{1}{\tau} & 0 \\ \frac{r_m}{i_f} & -\frac{2}{\tau_{\text{CAN}}} \end{pmatrix} \boldsymbol{x}_t(t) + \begin{pmatrix} \frac{1}{\tau} \\ 0 \end{pmatrix} \omega_{eq}(t) \\ v(t) &= \begin{pmatrix} -\frac{r_m}{i_f}, & \frac{4}{\tau_{\text{CAN}}} \end{pmatrix} \boldsymbol{x}_t(t) \end{cases} \tag{3.6}$$

where $\boldsymbol{x}_t(t)$ is the state of the model, and $v(t) = v'(t + \tau_{\text{CAN}})$ is the delayed velocity measurement.

3.3.4 Modeling of driveline and vehicle body

The dynamics of the elements in the driveline can be modeled using linear elements if gear shifting and saturations are not taken into account, see, e.g., (Zalm et al., 2008;

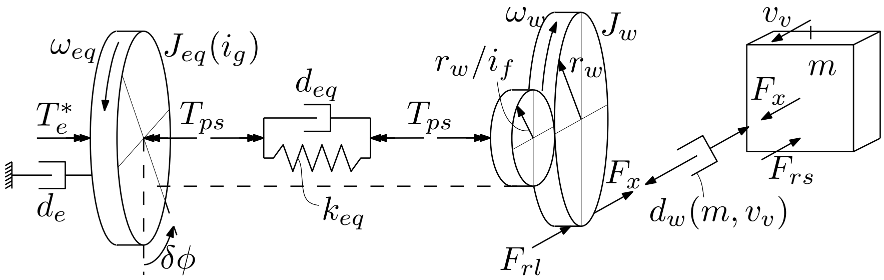


Figure 3.6: Schematic representation of the dynamical elements in the driveline and the vehicle body. For the sake of clarity, time dependencies are omitted.

Bruce, 2004; Lu and Hedrick, 2005). Assuming that the clutch is closed when cruise control functionality is enabled, all inertia, stiffness and damping in the driveline are lumped into equivalent variables $J_{eq} = J_{eq}(i_g(t))$, k_{eq} and d_{eq}, respectively, where $i_g(t)$ is the gear ratio. A schematic representation of the result is shown in Figure 3.6, where the resistance forces and the tire-road contact are also included. The corresponding rotational velocity $\omega_{eq}(t)$ equals the output of the gearbox, which is the input for the tachograph (see Figure 3.2). The corresponding lumped engine torque $T_e^*(t)$ equals

$$T_e^*(t) = \eta_g i_g(t) T_e(t) \tag{3.7}$$

where η_g is the gearbox efficiency. The torque in the propulsion shaft is given by:

$$T_{ps}(t) = \frac{1}{i_f^2 \eta_f} \left(k_{eq}\delta\phi(t) + d_{eq}\left(\omega_{eq}(t) - i_f \omega_w(t)\right)\right) \tag{3.8}$$

where i_f is the ratio of the final drive and η_f is the corresponding efficiency, $\delta\phi(t)$ is the torsion in the drive shafts, and $\omega_w(t)$ is the rotational velocity of the wheels.

The resistance forces $F_r(t)$ acting on the vehicle, include the aerodynamic drag force $F_a(t)$, the road inclination $F_\alpha(t)$, and the rolling resistance $F_{rl}(t)$

$$F_a(t) = 0.5 C_w \rho A_f v_v(t)^2 \tag{3.9}$$
$$F_\alpha(t) = m(t) g \sin\alpha(t) \tag{3.10}$$
$$F_{rl}(t) = m(t) g c_{rl} \cos\alpha(t) \tag{3.11}$$

where C_w is the aerodynamic drag coefficient, ρ is the air density, A_f is the frontal area of the truck, $v_v(t)$ is the actual vehicle velocity, g is the gravitational constant, $\alpha(t)$ is the road inclination, c_{rl} is the rolling resistance constant, $m(t)$ is the vehicle mass, and with $F_{rs}(t) = F_a(t) + F_\alpha(t)$ (see Figure 3.6).

Finally, the tire-road contact is modeled using a nonlinear damper model $d_w(m(t), v_v(t))$. Assuming only small values of pure longitudinal slip, the driving force $F_x(t)$, which the driven wheels exercise on the road, equals (Pacejka, 2002)

$$F_x(t) = C_{F_\kappa}(t)\kappa(t) \tag{3.12}$$

where $C_{F_\kappa}(t)$ is the longitudinal slip stiffness, and $\kappa(t)$ is the longitudinal slip

$$\kappa(t) = -\frac{v_v(t) - \omega_w(t)r_w}{v_v(t)} \tag{3.13}$$

This stiffness can be approximated by (Pacejka, 2002; Miège and Popov, 2005)

$$C_{F_\kappa}(t) = C_{F_\kappa,0}F_n(t) \tag{3.14}$$

where $C_{F_\kappa,0}$ is the longitudinal slip stiffness coefficient, and $F_n(t)$ is the normal force

$$F_n(t) = \eta_m m(t) g \tag{3.15}$$

where η_m is the mass fraction working on the driven wheels, $m(t)$ is the vehicle mass, and g is the gravitational constant. Combining (3.12) through (3.15) yields

$$F_x(t) = d_w(t)(\omega_w(t)r_w - v_v(t)) \tag{3.16}$$

where

$$d_w(t) = \frac{c_{F_\kappa,0}\eta_m m(t) g}{v_v(t)} \quad = \quad d_w(m(t), v_v(t)) \tag{3.17}$$

which can be regarded as a mass- and velocity-dependent damper.

Defining the state $\boldsymbol{x}_d(t) = (\delta\phi(t),\, \omega_{eq}(t),\, \omega_w(t),\, v_v(t))^T$ and the input $u(t) = T_e(t)$ yields the model $\mathcal{P}_{d,nl}$ for the driveline and the vehicle body:

$$\mathcal{P}_{d,nl} : \begin{cases} \dot{\boldsymbol{x}}_d(t) & = & \begin{pmatrix} \omega_{eq}(t) - i_f\omega_w(t) \\ J_{eq}^{-1}(i_g)\left(\eta_g i_g(t)u(t) - T_{ps}(t) - d_e\omega_{eq}\right) \\ J_w^{-1}\left(i_f\eta_f T_{ps}(t) - r_w F_x(t) - r_w F_{rl}(t)\right) \\ m^{-1}(t)\left(F_x(t) - F_{rs}(t)\right) \end{pmatrix} \\ y(t) & = & \omega_{eq}(t) \end{cases} \tag{3.18}$$

The model $\mathcal{P}_{d,nl}$ is nonlinear and variable in time due to the dependency on the mass $m(t)$, the gear ratio $i_g(t)$, the actual vehicle velocity $v_v(t)$, and the road inclination $\alpha(t)$.

3.3.5 LPV modeling

Consider the model $\mathcal{P}_{d,nl}$ (3.18), describing the dynamics of the driveline and the vehicle body. The road inclination $\alpha(t)$ is, in this case, regarded as a constant disturbance $\alpha(t) = \alpha \in A_O$, where A_O is the corresponding operating range. For the mass $m(t)$, or, more specifically, the loading of an HDV, a large operating range has to be taken into account. For a typical HDV, the operating range of the vehicle mass including the loading is given by $M_O = [m_{min}, m_{max}]$, where $m_{min} = 7000$ kg and $m_{max} = 40000$ kg. While driving, the loading will not change or only slowly in specific cases, e.g., for a garbage or gritting truck. Hence, it is assumed that the mass is constant, yielding $m(t) = m \in M_O$.

Furthermore, cruise control is only enabled when the clutch is closed, in which case the gear ratio $i_g(t)$ is constant. Moreover, the gear ratio comprises a limited number of fixed values, which are defined by the discrete gear numbers $G \in G_O = \{1, 2, \ldots, 12\}$. In practice, $G = 8$ is the lowest gear in which cruise control is enabled. Hence, the number of gears is limited, and switching between gears is accompanied by opening of the driveline via the clutch, which means that the cruise control functionality is switched on and off. Especially variations in the mass and the gear ratio have large influence on the dynamics, due to the large operating ranges.

The range of variation in the vehicle velocity $v_v(t)$ is coupled to the gear. Per gear, a limited range $v_v(t) \in V_O(G)$ has to be taken into account. Typically, this is a range of about 3.0 to 10.0 m/s. First, the velocity influences the aerodynamic drag force (3.9), which is, for simplicity, regarded as an low-frequent disturbance. Second, the velocity influences the nonlinear damper $d_w(m(t), v_v(t))$ (3.17), representing the tire-road contact. For the corresponding operating ranges, the effect of variations in $v_v(t)$ is, in general, small compared to variations in $m(t)$. Hence, for simplicity, the influence of variations in the vehicle velocity is not explicitly taken into account in the modeling and the corresponding controller synthesis.

Consequently, per gear G, a separate controller is synthesized, explicitly taking into account variations in the measured vehicle mass $m(t)$ via a GS or LPV controller synthesis. Variations in the vehicle velocity and the road inclination are considered as disturbances. Correspondingly, the operating point $\mathcal{R}_o$ is defined:

$$\mathcal{R}_o = (\alpha, G, v) \in A_O \times G_O \times V_O(G) \tag{3.19}$$

Per gear G, a mass-dependent LPV model is derived, where it is assumed that the mass is constant during operation. Hence, the resistance forces (3.9) through (3.11) can be regarded as constant disturbance forces. For the nominal controller design, they are assumed to be zero. Hence, special attention has to be paid to the steady-state error of closed-loop validation experiments, when the disturbance forces are not equal to zero.

Linearizing the model $\mathcal{P}_{d,nl}$ (3.18) at the operating point $\mathcal{R}_o$ (3.19), i.e., at a constant road inclination, a constant velocity, and for a specific gear, yields a 4^{th}-order mass-dependent LPV model $\mathcal{P}_d(m) : T_e(t) \mapsto \omega_{eq}(t)$ for the driveline and the vehicle body

$$\mathcal{P}_d(m) : \left\{ \begin{array}{rcl} \dot{\boldsymbol{x}}_d(t) & = & \mathbf{A}_d(m)\boldsymbol{x}_d(t) + \mathbf{B}_d u(t) \\ \omega_{eq}(t) & = & \mathbf{C}_d \boldsymbol{x}_d(t) \end{array} \right. \tag{3.20}$$

where $u(t) = T_e(t)$, and $\mathbf{A}_d(m)$ is linearly dependent on m^{-1}.

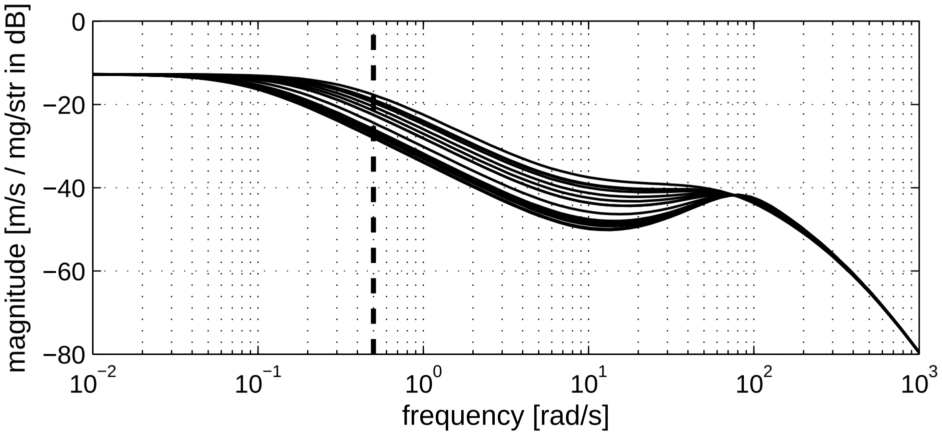


Figure 3.7: Bode magnitude plot of the open-loop system $\mathcal{P}_{\mathrm{HDV}}(\delta)$ for $\delta \in \Delta$ (3.21), and $\mathcal{R}_o = (0.0,\ 8,\ 10.0)$. The dashed line indicates the desired closed-loop bandwidth (see Table 3.1).

Define the (constant) scheduling parameter δ and the operating range Δ:

$$\delta = \frac{m - m_{min}}{m_{max} - m_{min}} \in \Delta = [0.0,\ 1.0] \tag{3.21}$$

and, correspondingly

$$m = (1 - \delta)m_{min} + \delta m_{max} \tag{3.22}$$

Substituting $m = m(\delta)$ in (3.20), and combining the resulting model $\mathcal{P}_d(\delta)$, the engine model $\mathcal{P}_e$ (3.5), and the tachograph model $\mathcal{P}_t$ (3.6) yields a linear, 7^{th}-order parameter-dependent model $\mathcal{P}_{\mathrm{HDV}}(\delta) : f_{e,d}(t) \mapsto v(t)$ that can be used for subsequent controller synthesis

$$\mathcal{P}_{\mathrm{HDV}}(\delta) : \begin{cases} \dot{\boldsymbol{x}}(t) &= \mathbf{A}_{\mathrm{HDV}}(\delta)\boldsymbol{x}(t) + \mathbf{B}_{\mathrm{HDV}} f_{e,d}(t) \\ v(t) &= \mathbf{C}_{\mathrm{HDV}}\boldsymbol{x}(t) \end{cases} \tag{3.23}$$

with $\boldsymbol{x}(t) = (x_e(t), \boldsymbol{x}_d(t), \boldsymbol{x}_t(t))^T$ the state vector, where $x_e(t)$, $\boldsymbol{x}_d(t)$, and $\boldsymbol{x}_t(t)$ are the state vectors of the engine model, the driveline model, and the tachograph model including the communication delay, respectively (see Section 3.3.3).

In Figure 3.7, a Bode magnitude plot of the model $\mathcal{P}_{\mathrm{HDV}}(\delta)$ is shown for a vehicle driving on a flat road in the lowest gear in which cruise control functionality is enabled, i.e., $\mathcal{R}_o = (0.0,\ 8,\ 10.0)$. The model is validated using measurement results with a real HDV, following the approach presented in Zalm et al. (2008). Targeting a closed-loop bandwidth of 0.5 rad/s (see Table 3.1, pg. 53), this figure shows the relevance of explicitly taking into account the mass dependency of the model in the controller synthesis, possibly reducing conservatism with respect to a corresponding robust design approach. For high frequencies, i.e., above the eigenfrequency of the driveline, the loading is virtually decoupled from the engine. For the lowest frequencies, the dynamics are dominated by the engine drag, which is represented by the damper d_e. Consequently, only for a specific frequency-band, the dynamics are actually influenced by variations in the loading, which is shown in Figure 3.7.

3.3.6 Generalized plant model

The mixed-sensitivity control problem formulation that is adopted in the controller synthesis Methods B to D (see Section 3.2.3) requires the definition of a generalized plant model $\mathcal{P}(\delta)$ (3.2). The generalized plant model combines the 7^{th}-order model of the HDV system (3.23) and the 1^{th}-order, the 2^{nd}-order, and the 0^{th}-order weighting filters $W_e(s)$, $W_y(s)$, and $W_u(s)$ (3.4), respectively. The model matrices of the resulting 10^{th}-order generalized plant model $\mathcal{P}(\delta)$ (3.2) (see Figure 3.3) are defined by:

$$
\begin{aligned}
\mathbf{A} &= \begin{pmatrix} \mathbf{A}_{\text{HDV}}(\delta) & 0 & 0 & 0 \\ -\mathbf{B}_{W_e}\mathbf{C}_{\text{HDV}} & \mathbf{A}_{W_e} & 0 & 0 \\ 0 & 0 & \mathbf{A}_{W_u} & 0 \\ \mathbf{B}_{W_y}\mathbf{C}_{\text{HDV}} & 0 & 0 & \mathbf{A}_{W_y} \end{pmatrix}, \quad \mathbf{B}_w = \begin{pmatrix} 0 \\ \mathbf{B}_{W_e} \\ 0 \\ 0 \end{pmatrix} \\
\mathbf{C}_z &= \begin{pmatrix} -\mathbf{D}_{W_e}\mathbf{C}_{\text{HDV}} & \mathbf{C}_{W_e} & 0 & 0 \\ 0 & 0 & \mathbf{C}_{W_u} & 0 \\ \mathbf{D}_{W_y}\mathbf{C}_{\text{HDV}} & 0 & 0 & \mathbf{C}_{W_y} \end{pmatrix}, \quad \mathbf{B}_u = \begin{pmatrix} \mathbf{B}_{\text{HDV}} \\ 0 \\ \mathbf{B}_{W_u} \\ 0 \end{pmatrix} \\
\mathbf{C}_y &= \begin{pmatrix} -\mathbf{C}_{\text{HDV}} \\ 0 \\ 0 \\ 0 \end{pmatrix}^T, \quad \mathbf{D}_{zw} = \begin{pmatrix} \mathbf{D}_{W_e} \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{D}_{zu} = \begin{pmatrix} 0 \\ \mathbf{D}_{W_u} \\ 0 \end{pmatrix} \\
\mathbf{D}_{yw} &= 1, \qquad \mathbf{D}_{yu} = 0
\end{aligned} \tag{3.24}
$$

where $\mathbf{A}_{W_i}$, $\mathbf{B}_{W_i}$, $\mathbf{C}_{W_i}$ and $\mathbf{D}_{W_i}$, $i = \{e,\ u,\ y\}$ are the matrices corresponding to a state-space representation of the corresponding weighting filters $W_i(s)$ (3.4).

3.4 Controller synthesis

Consider the cruise control problem defined in Section 3.3. Given a specific operating point $\mathcal{R}_o$ (3.19), and assuming that the mass is variable but constant during operation, the controller synthesis problem comes down to synthesizing a mass-dependent controller $\mathcal{K}(\delta)$ for the model $\mathcal{P}_{\text{HDV}}(\delta)$ (3.23) with operating range $\delta \in \Delta$ (3.21), and the performance requirements listed in Table 3.1 (pg. 53). In this section, application of the different controller synthesis methods that are listed in Section 3.2.3 is discussed. The results are summarized in Table 3.2 (pg. 75). The operating point $\mathcal{R}_o = \{0.0,\, 8,\, 10.0\}$ is used for the simulations presented in this section. Experimental results on a real vehicle and for different operating points are presented in the next section.

3.4.1 Method A: classical gain scheduling using manual loop shaping

Using manual loop-shaping techniques, a set of discretely-parameterized continuous-time controllers $\mathcal{K}_{\text{LS},o}(\delta)$ is synthesized for a grid $\delta \in \Delta_O \subset \Delta$, a subset of the operating range Δ (3.21):

$$\Delta_O = \{k/(N_O - 1)\,|\, k = 0,\, 1, \ldots, (N_O - 1)\} \subset \Delta = [0.0,\, 1.0], \qquad N_O > 1 \quad (3.25)$$

Given a physical model of the system, both the number of grid points N_O and the distribution of these points over the operating range Δ can be related to the influence of the scheduling parameter on of the relevant dynamics (see (3.18) and (3.20)) (Diepen, 2009). However, general guidelines for the definition of a grid are lacking. Moreover, in practice, a physical model is often lacking and controller design is based on measurement data and black-box modeling. Hence, in practice, a grid is often defined based on insight and experience. The main limitation in this design step is the lack of an explicit performance measure that indicates a-priori the influence of the chosen grid on the final closed-loop performance. To demonstrate the advantages and disadvantages of this non-automated design step, in this case, a linear grid Δ_O (3.25) with $N_O = 6$ operating points is used.

Accordingly, 4^{th}-order controllers $\mathcal{K}_{\text{LS},o}(\delta)$ are designed, including low-frequent integral action to achieve the demand on the steady-state error, a band-stop filter to achieve the desired bandwidth while preserving both the required modulus margin and enough phase margin to fulfill the demand on the maximum overshoot, and a low-pass filter to attain the desired high-frequency roll-off (see Table 3.1, pg. 53). Bode magnitude plots of the resulting set of controllers are shown in Figure 3.8.

Targeting online implementation, the set of controllers $\mathcal{K}_{\text{LS},o}(\delta)$, $\delta \in \Delta_O$ is discretized using a standard bilinear (Tustin) approximation scheme. To obtain a continuously parameterized family of controllers $\mathcal{K}_{\text{LS}}(\delta)$, $\delta \in \Delta$, the set of controllers $\mathcal{K}_{\text{LS},o}(\delta)$ has to be interpolated. This is, again, a non-automated design step for which general guidelines are missing. In this case, state-space interpolation is adopted, which is discussed in more detail in the next section.

As a result of the non-automated design steps, closed-loop stability and performance can be assessed only afterwards. In this case, piecewise affine Lyapunov stability analysis is adopted to analyze closed-loop stability over the entire operating range, taking into account the fact that the scheduling parameter is constant during operation. Closed-loop performance is assessed via simulations. For example, in Figure 3.9, step response

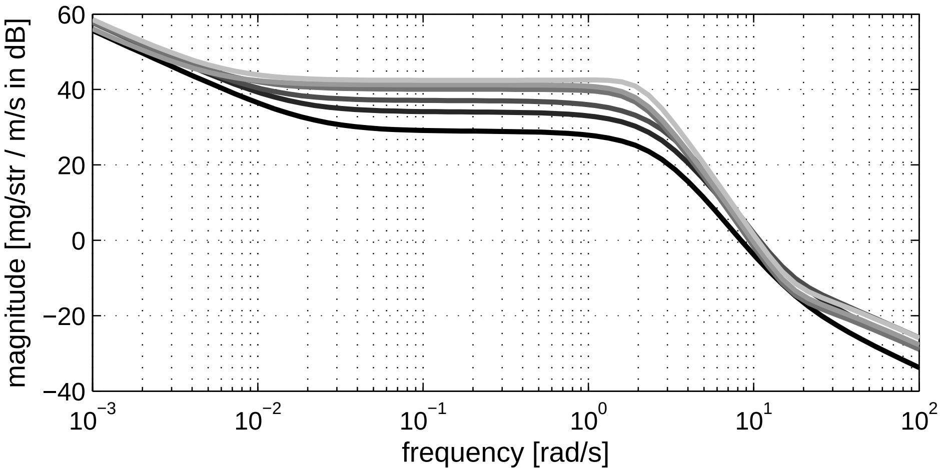


Figure 3.8: Bode magnitude plot of $\mathcal{K}_{\mathrm{LS},o}$, for $\delta \in \Delta_O$, $N_O = 6$ (3.25), where the plots are decreasing in darkness for increasing δ.

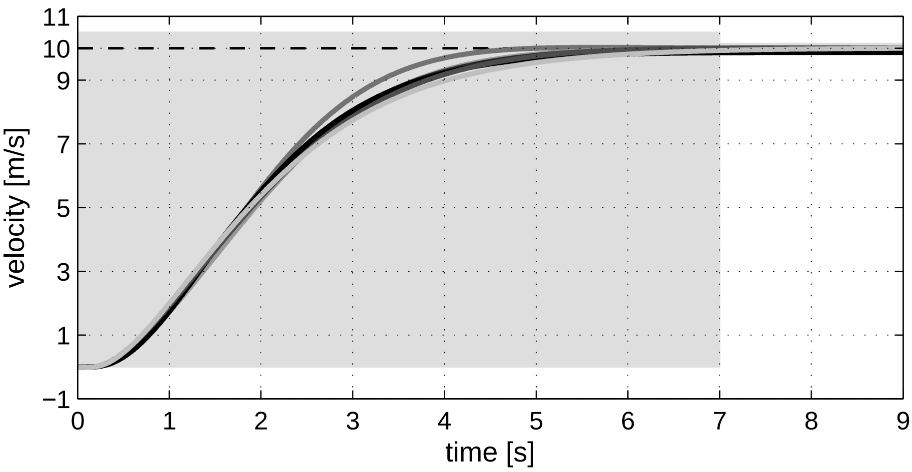


Figure 3.9: Step response results for the closed-loop system with $\mathcal{K}_{\mathrm{LS}}(\delta)$, for $\delta \in \Delta_O$, $N_O = 6$ (3.25), where the plots are decreasing in darkness for increasing δ, and the desired set velocity is indicated in dashed black. The grey patches indicate the desired settling time, the maximum overshoot, and the maximum steady-state error (see Table 3.1, pg. 53).

simulations with the resulting closed-loop system are shown for $\delta \in \Delta_O$, indicating that the performance requirements are achieved.

Implementation of the controller $\mathcal{K}_{\mathrm{LS}}(\delta)$, $\delta \in \Delta$ is straightforward. The use of the manual loop-shaping controller synthesis provides insight in the controller structure, which, in practice, is desirable. For example, some online fine-tuning or the addition of an anti-windup scheme are often applied only afterwards, requiring insight in the controller structure. However, the manual loop-shaping controller synthesis also implies that fine-tuning or re-tuning of the controller is relatively time-consuming. Per gear, a set of controllers $\mathcal{K}_{\mathrm{LS},o}(\delta)$, $\delta \in \Delta_O$ has to be synthesized, which took, in this case, about 1 h.

3.4.2 Method B: classical gain scheduling using $\mathcal{H}_\infty$ controller synthesis

Consider again the linear grid Δ_O (3.25) with $N_O = 6$ operating points $\delta \in \Delta_O$. Substituting δ in (3.2) and (3.24) yields a set of plant models $\mathcal{P}_o(\delta)$, $\delta \in \Delta_O$. Based on these models, a set of discretely parameterized $\mathcal{H}_\infty$ controllers $\mathcal{K}_{\mathcal{H}_\infty,o}(\delta)$, $\delta \in \Delta_O$ is synthesized with $\gamma \in [1.2023,\ 1.2030]$ using the Robust control toolbox of MATLAB (The MathWorks$^{\mathrm{TM}}$, 2010). Given the mixed-sensitivity performance weights $W_e(s)$, $W_y(s)$, and $W_u(s)$ (see Section 3.3.2), the controllers are an automatic result of the synthesis, as opposed to the manual loop-shaping approach that is adopted in Method A (see Section 3.4.1). Moreover, γ is an indicator for the performance of the designs, indicating, in this case, that the desired performance is reasonably well met. In Figure 3.10, the resulting closed-loop transfer functions are compared to the desired performance requirements, confirming

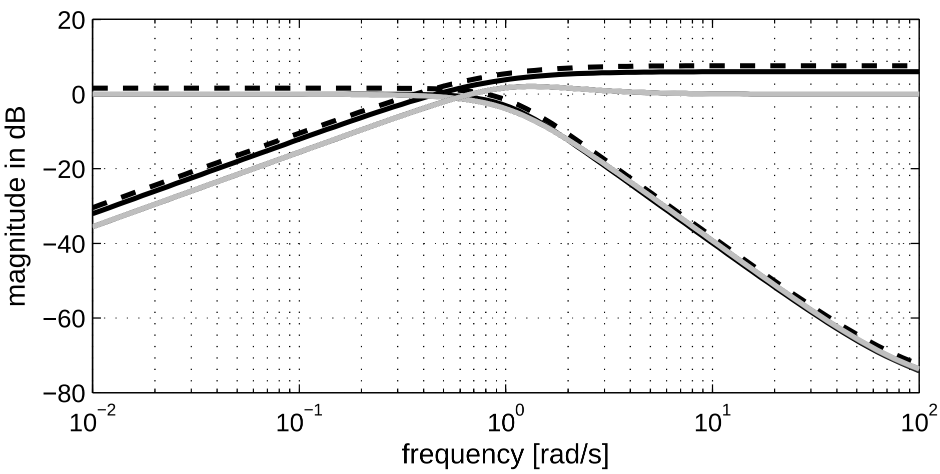


Figure 3.10: Sensitivity and complementary sensitivity plots for the closed-loop system with $\mathcal{K}_{\mathcal{H}_\infty,o}(\delta)$, for $\delta \in \Delta_O$, $N_O = 6$ (3.25) (solid grey), compared to the weights $W_e(s)^{-1}$ and $W_y(s)^{-1}$ (solid black), and $\gamma W_e(s)^{-1}$, $\gamma W_y(s)^{-1}$ (dashed black), with $\gamma = 1.2030$ the corresponding worst-case γ-value.

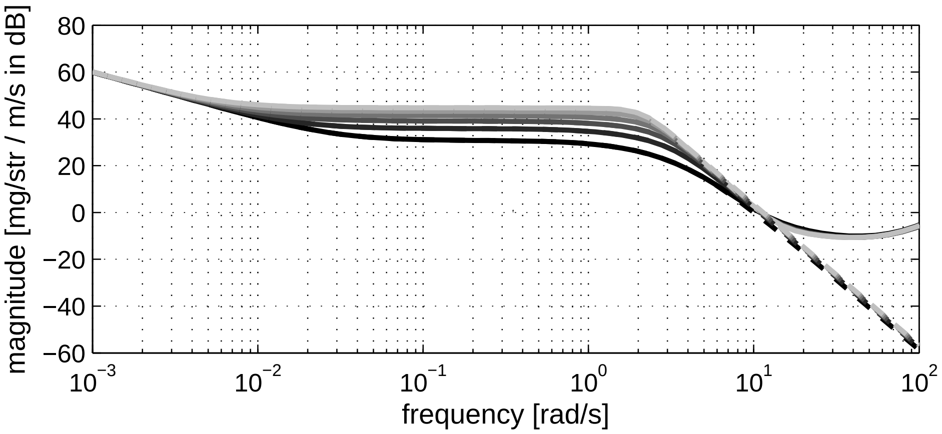


Figure 3.11: Bode magnitude plot of $\mathcal{K}_{\mathcal{H}_\infty,o}(\delta)$ (solid), and $\mathcal{K}_{\mathcal{H}_\infty,r,o}(\delta)$ (dashed), for $\delta \in \Delta_O$, $N_O = 6$ (3.25), where the plots are decreasing in darkness for increasing δ.

this result.

The next step is interpolation of the set of controllers $\mathcal{K}_{\mathcal{H}_\infty,o}(\delta)$, $\delta \in \Delta_O$, which, in this case, is based on insight and experience. To increase insight in the controller structure, order reduction is applied. The controller poles and zeros influencing the controller characteristics only above the desired closed-loop bandwidth of 0.5 rad/s are removed. This is a non-automated design step, based on insight in the system dynamics. The result is a set of 4^{th}-order controllers $\mathcal{K}_{\mathcal{H}_\infty,r,o}(\delta)$, $\delta \in \Delta_O$, which are compared to the original 10^{th}-order controllers in Figure 3.11. The result in Figure 3.11 indicates that the effect of this controller reduction is small in the relevant frequency range, which is confirmed by analysis of the closed-loop stability and performance characteristics.

Targeting online implementation, the set of controllers $\mathcal{K}_{\mathcal{H}_\infty,r,o}(\delta)$, $\delta \in \Delta_O$ is discretized using a standard bilinear (Tustin) approximation scheme. The resulting set of controllers is written in an observable canonical state-space form. In Figure 3.12, the black dots indicate the values of the corresponding state-space matrices for $\delta \in \Delta_O$. The observable canonical state-space form ensures for this specific example that most of the elements of the state-space matrices are constant over the operating range Δ, being either 0 or 1 (see Figure 3.12). Hence, assuming individual interpolation of all elements of the state-space matrices, complexity of the interpolation is reduced.

A physical model of the system dynamics can provide a guideline for the interpolation of the other elements (see (3.18) and (3.20)). In practice, however, black-box models and least-squares interpolation techniques are often applied. Especially in the latter case, the main problem is a lack of an explicit a-priori performance definition for the interpolation. Evaluation of the resulting continuously parameterized family of controllers $\mathcal{K}_{\mathcal{H}_\infty,r}(\delta)$, $\delta \in \Delta$ is only possible afterwards. The closed-loop performance at the grid points enables

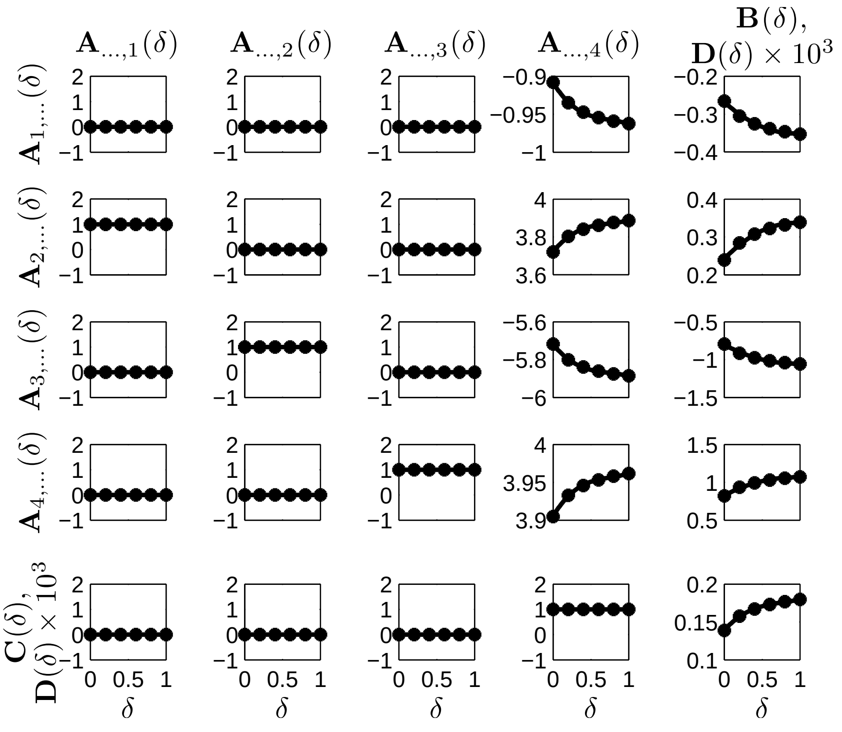


Figure 3.12: Interpolation result of the state-space matrices $\mathbf{A}(\delta)$, $\mathbf{B}(\delta)$, $\mathbf{C}(\delta)$, and $\mathbf{D}(\delta)$ of $\mathcal{K}_{\mathcal{H}_\infty,r,o}(\delta)$, $\delta \in \Delta_O$, $N_O = 6$ (3.25), with a (maximum) interpolation order $N_I = 2$.

to evaluate the approximation accuracy of the interpolation. Evaluation of the closed-loop performance at inter-grid points provides insight in both the interpolation and the corresponding grid. If the closed-loop performance at inter-grid points deviates more from the closed-loop performance than is expected on the basis of the set of operating-point dependent controllers $\mathcal{K}_{\mathcal{H}_\infty,r,o}(\delta)$, $\delta \in \Delta_O$, either the interpolation or the corresponding grid has to be changed.

A tradeoff is made between the complexity of the interpolation N_I and the approximation accuracy of the interpolation. For a set of $N_O = 6$ controllers, in general, only a 5^{th}-order interpolation ensures exact intersection with the original controllers. Often, however, the number of grid points N_O that is used, is larger than the required order of the interpolation, facilitating the use of least-squares-like interpolation techniques. Moreover, complexity of the resulting controller is directly related to the complexity of the interpolation. Hence, in practice, a low-order interpolation is desirable.

To demonstrate this, a state-space interpolation is adopted, using a 2^{nd}-order rational δ-parameterization, which is illustrated in Figure 3.12. In Figure 3.13, closed-loop step response simulation results with the controller $\mathcal{K}_{\mathcal{H}_\infty,r}(\delta)$ are shown for $\delta \in \Delta_O$, with $N_O = 11$. As the closed-loop performance requirements are achieved, these results indi-

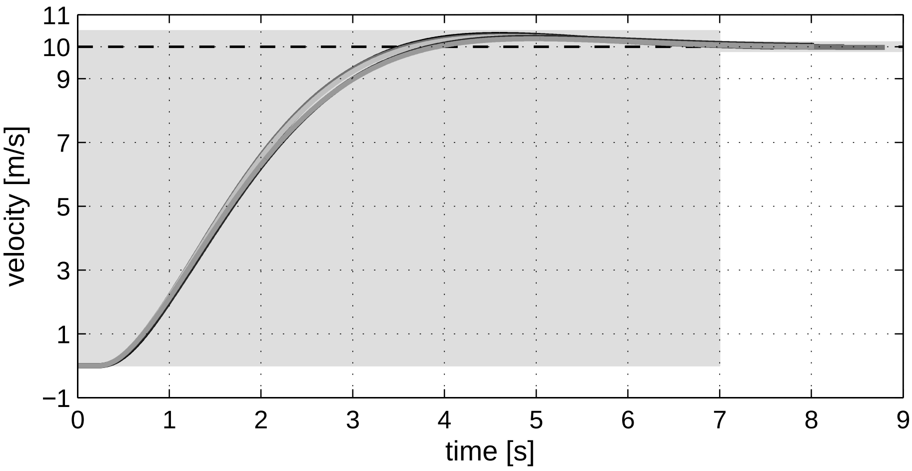


Figure 3.13: Step response results for the closed-loop system with $\mathcal{K}_{\mathcal{H}_\infty,r}(\delta)$, for $\delta \in \Delta_O$, $N_O = 11$ (3.25), where the plots are decreasing in darkness for increasing δ, and the desired set velocity is indicated in dashed black. The grey patches indicate the desired settling time, the maximum overshoot, and the maximum steady-state error (see Table 3.1, pg. 53).

cate that the interpolation approximation is accurate enough and that the 2$^{\text{nd}}$-order interpolation accurately describes the influence of the scheduling parameter on the dynamics for the specified operating range $\delta \in \Delta$.

Analogously, closed-loop stability can be assessed only afterwards. Based on an LFT representation of the closed-loop system (see Figure 3.1(b)), and re-scaling of $\delta \in [0.0,\ 1.0] \Leftrightarrow \delta' \in [-1.0,\ 1.0]$, standard μ robust stability analysis is adopted to analyze stability (Skogestad and Postlethwaite, 2005). The result indicates that the closed-loop system is robustly stable for $\delta \in [-0.21,\ 1.21] \supset \Delta$. Hence, closed-loop stability is guaranteed. However, note that this holds for the model dynamics rather than the actual system dynamics.

Implementation of the controller $\mathcal{K}_{\mathcal{H}_\infty,r}(\delta)$ is comparable to the implementation of $\mathcal{K}_{\text{LS}}(\delta)$ in Method A (see Section 3.4.1). In this case, the order reduction provides insight in the resulting controller structure. However, due to the automated $\mathcal{H}_\infty$ controller synthesis method, re-synthesizing of the controller is less time-consuming than in Method A, taking about 0.1 h, depending on the operating system (OS) and the available memory (see Table 3.2).

3.4.3 Method C: LPV controller synthesis using an extended $\mathcal{H}_\infty$ problem definition and gridding

Consider an LPV controller synthesis method using an extended $\mathcal{H}_\infty$ problem definition and gridding. Based on the results of Method B (see Section 3.4.2), a rational δ-parameterization is used to define a common quadratic Lyapunov function $\mathbf{X}(\delta)$ and a controller $\mathcal{K}_{\text{LPV}}(\delta)$:

$$\mathbf{X}(\delta) = \mathbf{X}_0 + \delta\mathbf{X}_1 + \ldots + \delta^{N_I}\mathbf{X}_{N_I} \tag{3.26}$$

$$\mathcal{K}_{\text{LPV}}(\delta) = \mathbf{K}_0 + \delta\mathbf{K}_1 + \ldots + \delta^{N_I}\mathbf{K}_{N_I} \tag{3.27}$$

where $\delta \in \Delta$, $\mathbf{X}_i = \mathbf{X}_i^T \in \mathbb{R}^{n_\xi \times n_\xi}$, $\mathbf{K}_i \in \mathbb{R}^{(n_\xi+n_y)\times(n_\xi+n_u)}$ for $i = 0, 1, \ldots, N_I$, and N_I is the interpolation order. For $N_I = 2$, the parameterization equals the interpolation of the set of $\mathcal{H}_\infty$ controllers $\mathcal{K}_{\mathcal{H}_\infty,r,o}(\delta)$ of Method B. A general drawback of LPV controller synthesis methods is the conservatism that is introduced by this finite dimensional parameterization. The amount of conservatism is, in general, difficult to measure. In this case, however, this can be related to the performance of the $\mathcal{H}_\infty$ point designs of Method B, which can be regarded as optimal.

Based on the plant model (3.2), (3.24), the Lyapunov function (3.26) and the controller (3.27), the controller synthesis problem can be defined as an extended $\mathcal{H}_\infty$ problem, yielding a continuously parameterized family of infinitely many, nonlinear constraints (Scherer et al., 1997). To arrive at a family of linear matrix inequalities (LMIs), the nonlinear change of variables $(\mathbf{X}(\delta), \mathcal{K}_{\text{LPV}}(\delta)) \mapsto (\mathbf{X}'(\delta), \mathcal{K}'_{\text{LPV}}(\delta))$ (3.3) is applied. Adopting the grid $\delta \in \Delta_O$, $N_O = 6$ (3.25), the upper-bound $\gamma = 1.2030$ and the interpolation order $N_I = 2$ that are used in Method B, the problem is reduced to a discretely parameterized family of finitely many feasibility constraints, which can be solved with standard algorithms. As the problem is feasible for the worst-case γ-value corresponding to the $\mathcal{H}_\infty$ point designs, it can be assumed that the amount of conservatism is negligible in this case. Hence, given the δ-parameterization (3.26) and (3.27), the interpolation is automatically included in the controller synthesis, yielding an LPV controller $\mathcal{K}_{\text{LPV}}(\delta)$, $\delta \in \Delta$.

Closed-loop stability and performance are guaranteed for the operating points $\delta \in \Delta_O$. However, the guarantees are also limited to these operating points. To assess stability and performance for the intermediate points $\delta \in \Delta \setminus \Delta_O$, commonly, the feasibility problem constraints are evaluated for a denser validation grid $\delta \in \Delta_v$ with grid points $N_v \gg N_O$, using the previously computed Lyapunov function and LPV controller (Wu et al., 1995; Gianelli and Primbs, 2000). If the problem is feasible for this denser grid, it is assumed that the problem is feasible for the total operating range $\delta \in \Delta$. In this case, a validation grid Δ_v with $N_v = 100$ is used, guaranteeing closed-loop stability and a closed-loop performance with $\gamma = 1.2030$ for the total operating range $\delta \in \Delta$. In Figure 3.14, step response results for the closed-loop system with $\mathcal{K}_{\text{LPV}}(\delta)$, $\delta \in \Delta_O$ are shown,

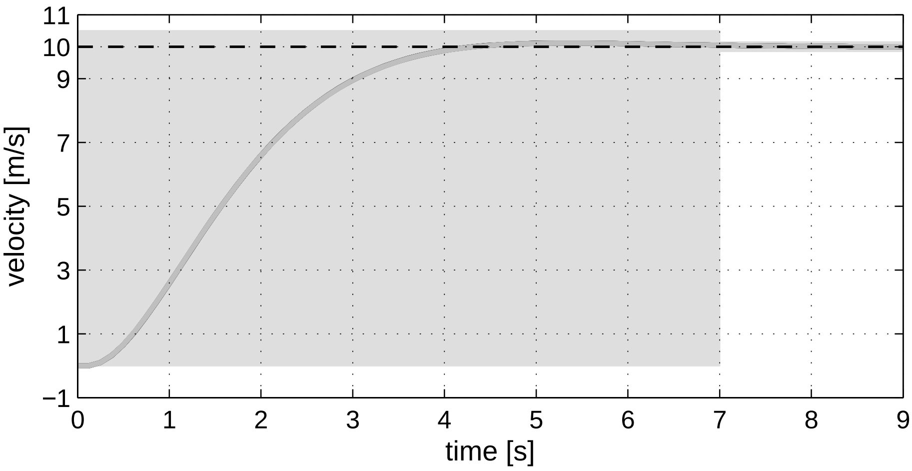


Figure 3.14: Step response results for the closed-loop system with $\mathcal{K}_{\text{LPV}}(\delta)$, for $\delta \in \Delta_O$, $N_O = 6$ (3.25) (solid grey), and the desired set velocity is indicated in dashed black. The grey patches indicate the desired settling time, the maximum overshoot, and the maximum steady-state error (see Table 3.1, pg. 53).

indicating that the performance requirements are indeed achieved (see Section 3.3.2).

Implementation of the controller is difficult in comparison with the Methods A and B. First, a singular value decomposition (SVD) is required to invert the nonlinear change of variables (3.3) and to obtain the actual controller $\mathcal{K}_{\text{LPV}}(\delta)$. Second, the relatively high, 10^{th}-order of the controller may yield implementation problems if (large) modeling errors are present. Third, solving the feasibility problem takes relatively much time, depending on the complexity of the problem. This complexity depends directly on both the number of operating points N_O and on the interpolation order N_I. For $N_O = 6$ and $N_I = 2$, the resulting set of LMIs contain 693 decision variables. Solving the problem with the LMI control toolbox of MATLAB 7.9, takes in the order of 10 h, depending on the operating system (OS) and the available memory (see Table 3.2) (The MathWorks$^{\text{TM}}$, 2010). Hence, online fine-tuning of the controller or the weighting filters is limited by this time consuming controller synthesis.

3.4.4 Method D: LPV controller synthesis using an extended $\mathcal{H}_\infty$ problem definition and an extension of the KYP-lemma

To arrive at a controller synthesis with a-priori full stability and performance guarantees, an extension of the classical Kalman-Yakubovich-Popov (KYP) lemma is used. The KYP-lemma states that continuously-parameterized frequency-domain inequalities can be reformulated as parameter-independent conditions, see, e.g., (Rantzer, 1996). An extension of the KYP-lemma to constraints with rational parameter-dependencies δ is presented in Dinh et al. (2005) (see also Rossignol et al., 2003), rewriting parameter-dependent constraints into equivalent parameter-independent constraints, where δ is real-valued, time-invariant, and bounded by $\delta \in [0.0,\ 1.0]$.

Lemma 3.4.1 *(Dinh et al., 2005) Let $\mathbf{\Phi}(\delta)$ be a rational matrix function of the time-invariant scalar parameter δ, well-posed on $[0.0,\ 1.0]$, and defined by its LFT realization*

$$\mathbf{\Phi}(\delta) = \mathcal{F}\left(\delta \mathbf{I}, \begin{pmatrix} \mathbf{A_\Phi} & \mathbf{B_\Phi} \\ \mathbf{C_\Phi} & \mathbf{D_\Phi} \end{pmatrix}\right) \tag{3.28}$$

with $\mathcal{F}$ the lower fractional transformation, and $\mathbf{A_\Phi}$, $\mathbf{B_\Phi}$, $\mathbf{C_\Phi}$, $\mathbf{D_\Phi}$ system matrices. Let $\mathbf{M}$ be a matrix of appropriate size, then the infinite-dimensional condition

$$\mathbf{\Phi}(\delta)^T \mathbf{M} \mathbf{\Phi}(\delta) < \mathbf{0}, \qquad \forall \delta \in [0.0,\ 1.0] \tag{3.29}$$

holds if and only if there exists a symmetric positive definite matrix $\mathbf{S} = \mathbf{S}^T \geq 0$ and a skew-symmetric matrix $\mathbf{G} = -\mathbf{G}^T$ of compatible sizes, such that

$$\begin{pmatrix} \mathbf{C_\Phi^T} \\ \mathbf{D_\Phi^T} \end{pmatrix} \mathbf{M} \begin{pmatrix} \mathbf{C_\Phi} & \mathbf{D_\Phi} \end{pmatrix} + \begin{pmatrix} \mathbf{A_\Phi^T}(\mathbf{S}-\mathbf{G}) + (\mathbf{S}+\mathbf{G})\mathbf{A_\Phi} - 2\mathbf{S} & (\mathbf{S}+\mathbf{G})\mathbf{B_\Phi} \\ \mathbf{B_\Phi^T}(\mathbf{S}-\mathbf{G}) & \mathbf{0} \end{pmatrix} < \mathbf{0} \tag{3.30}$$

which is a parameter-independent, finite dimensional inequality.

Lemma 3.4.1 can be regarded as a specific version of the full-block S-procedure (Scherer, 1999), using the multiplier

$$\mathbf{P} = \begin{pmatrix} -2\mathbf{S} & \mathbf{S}+\mathbf{G} \\ (\mathbf{S}+\mathbf{G})^T & \mathbf{0} \end{pmatrix} \tag{3.31}$$

Hence, Lemma 3.4.1 states that for parameter-dependent systems with rational, time-invariant and real-valued parameter dependency $\delta \in [0.0,\ 1.0]$, the full-block S-procedure is exact, i.e., no conservatism is introduced, if the multiplier $\mathbf{P}$ is used (Rossignol et al., 2003). This approach is applied in Dinh et al. (2005) to equivalently recast the infinite-dimensional extended $\mathcal{H}_\infty$ problem with rational parameter-dependency $\delta \in \Delta$ that is considered in Method C, as a finite-dimensional problem.

Define a Lyapunov function $\mathbf{X}(\delta)$ and a controller $\mathcal{K}_{\text{KYP}}(\delta)$ (Dinh et al., 2005)

$$\mathbf{X}(\delta) = \frac{\sum_{i=0}^{N_I} \delta^i \mathbf{X}_i}{1 + \sum_{i=1}^{N_I} \delta^i k_i}, \quad \mathcal{K}_{\text{KYP}}(\delta) = \frac{\sum_{i=0}^{N_I} \delta^i \mathbf{K}_i}{1 + \sum_{i=1}^{N_I} \delta^i k_i} \tag{3.32}$$

where $\delta \in \Delta$, $\mathbf{X}_i = \mathbf{X}_i^T \in \mathbb{R}^{n_\xi \times n_\xi}$, $\mathbf{K}_i \in \mathbb{R}^{(n_\xi+n_y)\times(n_\xi+n_u)}$, $k_i \in \mathbb{R}$ for $i = 0, 1, \ldots, N_I$, and N_I is the interpolation order. The possible conservatism introduced by this finite

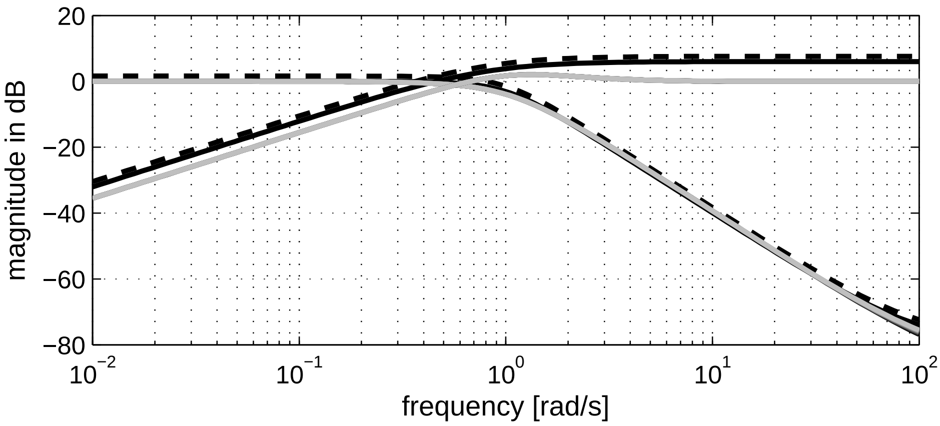


Figure 3.15: Sensitivity and complementary sensitivity plots for the closed-loop system with $\mathcal{K}_{\text{KYP}}(\delta)$, for $\delta \in \Delta_O$, $N_O = 6$ (3.25) (solid grey), compared to the weights $W_e(s)^{-1}$ and $W_y(s)^{-1}$ (solid black), and $\gamma W_e(s)^{-1}$, $\gamma W_y(s)^{-1}$ (dashed black), with $\gamma = 1.2030$.

parameterization is reduced by increasing the interpolation order N_I, at the cost of a controller of higher complexity (Dinh et al., 2005).

Based on the generalized plant model (3.2), (3.24), the Lyapunov function $\mathbf{X}(\delta)$ and the controller $\mathcal{K}_{\text{KYP}}(\delta)$ (3.32), the synthesis problem is defined as an extended $\mathcal{H}_\infty$ problem, which yields a continuously parameterized family of infinitely many, nonlinear constraints (Scherer et al., 1997). Analogous to Method C, the nonlinear change of variables (3.3) is applied to arrive at a family of LMIs. Next, instead of defining a grid Δ_O (3.25), the extended KYP lemma is applied to arrive at a finite-dimensional family of LMIs.

Given an upper-bound for $\gamma = 1.2030$ (see Section 3.4.2), the (finite) family of LMIs becomes a feasibility problem, which can be solved using standard algorithms. Although the extended KYP-lemma increases the theoretical complexity of the problem significantly, the main advantage is that the resulting controller automatically inherits full stability and performance guarantees for $\delta \in \Delta$. The resulting closed-loop sensitivity and complementary sensitivity frequency response functions are shown in Figure 3.15, confirming that the performance requirements are met reasonably well.

The problem size depends directly on the order of the interpolation N_I. For $N_I = 2$, the LMIs contain 4599 decision variables. Solving the problem with the LMI control toolbox of MATLAB 7.9 (The MathWorks$^{\text{TM}}$, 2010) using YALMIP 3 (Löfberg, 2004), takes in the order of 100 h, depending on the OS and the available memory (see Table 3.2). As the largest matrix size exceeds 1500 MB, a 64-bit OS in combination with a 64-bit version of MATLAB has been used. Implementation of $\mathcal{K}_{\text{KYP}}(\delta)$ is comparable to implementation of $\mathcal{K}_{\text{LPV}}(\delta)$ (see Section 3.4.3).

3.5 Experimental results

In the previous sections, simulation results for a specific gear $G = 8$ are presented (see Figure 3.9, 3.13, and 3.14). In this section, experimental measurement results are presented using different gears with corresponding velocities.

To validate the closed-loop behavior in practice, the controllers are implemented on a heavy duty vehicle (HDV), namely a DAF XF105. In Figure 3.16, a schematic representation of the experimental setup and a picture of the HDV are shown. A dSpace MicroAutoBox (MABX) rapid control prototyping platform is used to implement the controllers. Experiments with and without a fully loaded semi-trailer are performed, resulting in a vehicle mass of 39400 and 7400 kg, respectively. Via the controller area network (CAN) of the vehicle, an online estimate of the mass is obtained from the on-board electronic braking system (EBS) (WABCO, 2010).

The results are obtained with the controllers $\mathcal{K}_{\mathrm{LPV}}(\delta)$ (Method C) and $\mathcal{K}_{\mathcal{H}_\infty,r}(\delta)$ (Method B). The closed-loop characteristics using the controllers $\mathcal{K}_{\mathrm{LPV}}(\delta)$ and $\mathcal{K}_{\mathrm{KYP}}(\delta)$ are practically the same. Furthermore, implementation of $\mathcal{K}_{\mathrm{LS}}(\delta)$ and $\mathcal{K}_{\mathcal{H}_\infty,r}(\delta)$ is comparable, but differs from implementation of the other two controllers.

In Figure 3.17, step response results are shown for the HDV without the semi-trailer, driving in gear $G \in \{8, 10, 12\}$. The results show that the use of different gears and, correspondingly, different velocities does not influence the closed-loop performance. For all experiments, the closed-loop behavior satisfy the desired performance requirements (see Table 3.1, pg. 53). Furthermore, the experimental results resemble the simulation results presented in the previous section.

The results show that differences between $\mathcal{K}_{\mathrm{LPV}}(\delta)$ and $\mathcal{K}_{\mathcal{H}_\infty,r}(\delta)$ are negligible. This is confirmed by comparing the power spectral density (PSD) estimates of the measured velocity signals. For example, in Figure 3.18 the PSD estimates of the measurements at

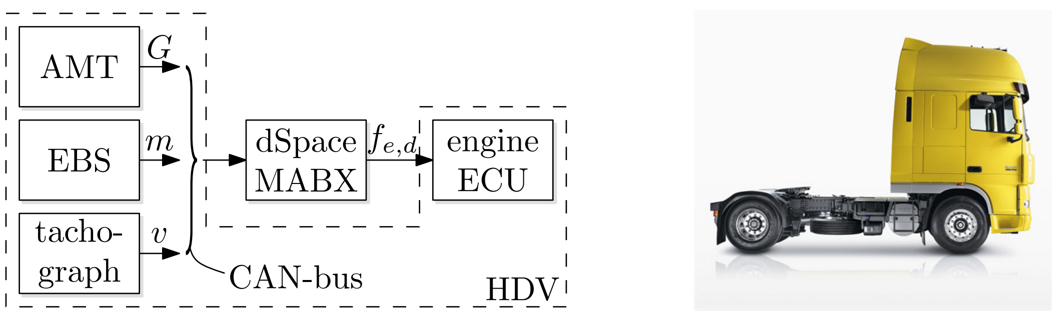


Figure 3.16: Schematic representation of the experimental setup and a picture of the DAF XF105 with which the experiments are performed. For the sake of clarity, time dependencies of the signals are omitted.

60 km/h are shown (corresponding to the middle plots in Figure 3.17). The controllers have been implemented at 100 Hz. However, the CAN-bus communication actually samples the measurements at 20 Hz, resulting in an effective Nyquist frequency of 10 Hz. Correspondingly, peaks can be observed at 20 and 40 Hz, which are present in every measurement, having constant power density. Accordingly, they can be related to the sampling by the CAN bus.

The peak at 15 Hz exactly matches the rotation frequency of the gearbox output shaft, while the peak at 12.5 Hz exactly matches half the rotation frequency of the engine drive shaft when the vehicle is driving at 60 km/h. Moreover, for measurements with other velocities, the peaks shift accordingly. Hence, these peaks most likely originate from this rotation. Furthermore, as a result of aliasing, the peaks are mirrored around the Nyquist frequency of 10 Hz, yielding corresponding peaks at 5 and 7.5 Hz. The same effect can be observed around 30 and 50 Hz, explaining the peaks at 25, 35 and 45 Hz, as well as at 27.5, 32.5 and 47.5 Hz.

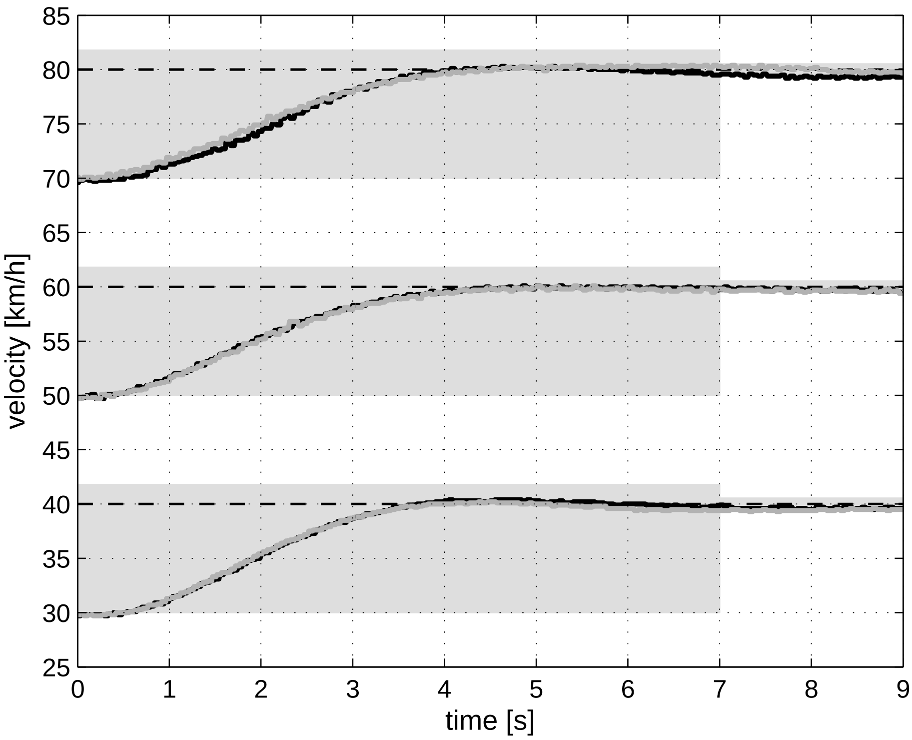


Figure 3.17: Step response results for the HDV with $\mathcal{K}_{\mathcal{H}_\infty,r}(\delta)$ (black), and $\mathcal{K}_{\text{LPV}}(\delta)$ (grey), for $m = 7400$ kg $\Leftrightarrow \delta = 0.01$, driving on a flat road in gear 8, 10, and 12 in the bottom, middle, and upper plots, respectively. The desired set velocities are indicated in dashed black. The grey patches indicate the desired settling time, the maximum overshoot, and the maximum steady-state error (see Table 3.1, pg. 53).

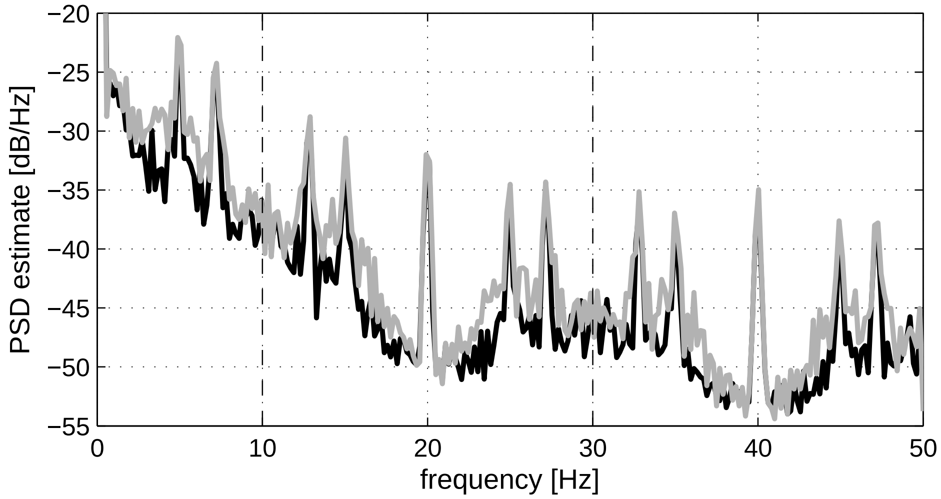


Figure 3.18: Power spectral density (PSD) estimate of the measured velocity signal for $m = 7400$ kg, driving on a flat road in gear 10 (corresponding to the middle plot in Figure 3.17), for $\mathcal{K}_{\mathcal{H}_\infty,r}(\delta)$ (black), and $\mathcal{K}_{\mathrm{LPV}}(\delta)$ (grey). The Nyquist frequency at 10 Hz (and 30 Hz) is indicated in dashed black.

In Figure 3.19, step response results are shown for the HDV with the fully-loaded semi-trailer, driving in gear $G = 10$. In this case, the desired engine torque is not achieved, which is illustrated in the bottom figure. The actual engine torque is limited by the engine hardware and software. As a result, the desired performance requirements are not achieved, which is shown in the upper figure. Experiments with other gears and velocities show the same closed-loop behavior. In this specific application, the effect of the integrator is relatively small. As a result, the overshoot is not noticeably increased by the saturation, although no anti-windup scheme is implemented.

Based on the experimental results, it is concluded that the controllers work appropriately in practice. If the engine torque is not saturated, the performance requirements are achieved, irrespective of the vehicle mass, the gear and the vehicle velocity.

3.6 Conclusions and recommendations

In this chapter, a comparison of different gain scheduling (GS) and linear parameter varying (LPV) controller synthesis methods for the design of a cruise control for heavy duty vehicles (HDVs) is made. Focus is on exposing the limitations of the classical GS methods that are often applied in practice and assessing the practical applicability of more recent LPV methods.

Adopting a GS or LPV controller synthesis method enables to account for the variable

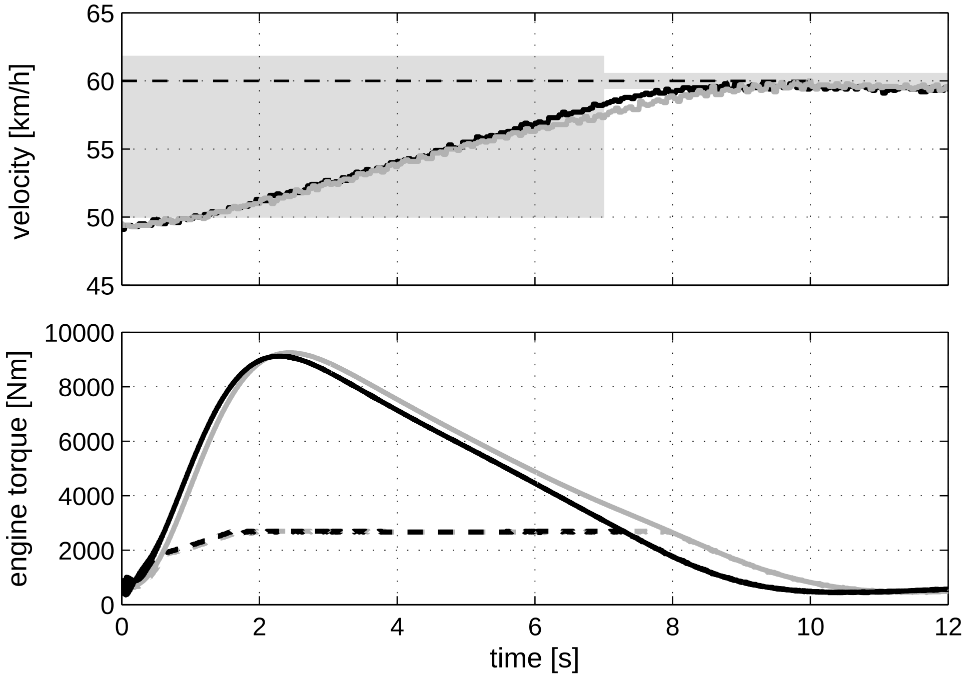


Figure 3.19: Upper figure: step response results for the HDV with $\mathcal{K}_{\mathcal{H}_\infty,r}(\delta)$ (black), and $\mathcal{K}_{\mathrm{LPV}}(\delta)$ (grey), for $m = 39400\,\mathrm{kg} \Leftrightarrow \delta = 0.98$, driving on a flat road in gear 10, the desired set velocity is indicated in dashed black. The grey patches indicate the desired settling time, the maximum overshoot, and the maximum steady-state error (see Table 3.1, pg. 53). Bottom figure: desired engine torque (solid) and actual engine torque (dashed), corresponding to the step response measurements of the upper figure.

loading of the vehicle in the cruise control functionality. The loading is constant during operation, but can vary in a, typically for HDVs, large range in between operations, influencing the relevant vehicle dynamics. The results in this chapter demonstrate that application of GS and LPV techniques enables to account for these variations in the cruise control design. Experimental results with a DAF XF105 demonstrate that this approach is beneficial for HDVs, facilitating constant performance over the operating range (see Section 3.5).

In Table 3.2, an overview of the different controller synthesis methods that are considered in this research is given (see also Section 3.2.3). First, going from Method A to D, the number of required non-automated design steps in the synthesis methods decreases. In Method A all design steps are non-automated: starting with gridding of the operating range, followed by synthesizing a set of discretely parameterized controllers, interpolation of this set of controllers, and evaluation of closed-loop stability and performance only

Table 3.2: Comparison of different GS and LPV controller synthesis methods.

characteristics	A: non-automated	B: semi-automated	C: automated, limited guarantees	D: automated, full guarantees
synthesis method	manual loop shaping	$\mathcal{H}_\infty$	extended $\mathcal{H}_\infty$	extended $\mathcal{H}_\infty$
stability guarantees	none	none	$\delta \in \Delta_O$	$\delta \in \Delta$
performance analysis	simulations	simulations	simulations, γ for $\delta \in \Delta_O$	γ for $\delta \in \Delta$
performance requirements	bandwidth, gain and phase margin	mixed sensitivity weights	mixed sensitivity weights	mixed sensitivity weights
gridding	yes	yes	yes	N/A
interpolation	manual	manual	automated*	N/A*
comp. time [h]**	~ 1.0	~ 0.1	~ 10	> 100

*A δ-parameterization is determined a-priori.

**Computations are performed on an Intel® Core Duo 3.00 GHz processor with 4.0 GB RAM, using 64-bit versions of WINDOWS and MATLAB (The MathWorks™, 2010).

afterwards. In Method D all steps are automated: after defining the controller synthesis problem, which is based on a robust performance specification, the LPV controller is an automatic result, inheriting full stability and performance guarantees. The number of non-automated design steps decreases gradually in the Methods B and C. Simultaneously, the number of guarantees that can be given increases.

However, it has to be remarked that these guarantees consider the model that is used, rather than the actual vehicle dynamics. Robustness models could be used to take into account differences between the model and the actual dynamics, however, differences will always be present. Consequently, absolute stability and performance guarantees for the actual vehicle dynamics cannot be given.

The main problem of the classical gain-scheduling techniques that are often adopted in practice is the lack of a measure of performance for the required non-automated design steps. It is not clear how the number of grid points, the interpolation method, or the interpolation order influences stability and performance of the resulting controller. Commonly, these non-automated design steps are based on insight and experience (see, e.g., Section 3.4.1 and 3.4.2). Furthermore, a-priori stability and performance guarantees lack, requiring analysis, simulations, and tests afterwards.

Second, going from Method A to D, the theoretical complexity of the controller syntheses, which is proportional to the computation time that is required to compute the controllers, increases significantly, with Method A being an exception. In Method A, manual loop-shaping techniques are adopted, requiring relatively much time, although theoretical complexity is low. Especially the time that is required to re-synthesize the controller when performance requirements change, is relatively large in comparison to Method B, adopting $\mathcal{H}_\infty$ techniques.

Considering implementation, the increasing theoretical complexity of the controller synthesis methods can form a problem. In practice, some online fine-tuning of the controller or the model that is adopted in the controller design is often desirable, which requires re-synthesizing the controller. However, insight in the resulting controller structure often decreases for increasing theoretical complexity, whereas the time required to re-synthesize a controller generally increases. Compare, for example, the computation effort that is required in the Methods A and D (see Table 3.2). In this research, the results of Method B are used in both the Methods C and D. Hence, the required on-line fine-tuning is based on the results of Method B, rather than using the more time-consuming Methods C and D, which are only used after the fine-tuning.

The differences between the classical GS controller synthesis methods that are often applied in practice and the available more recent LPV methods are large. The results in this chapter indicate that, in practice, a shift to the use of more automated methods can be made, reducing the amount of non-automated design steps and increasing a-priori stability and performance guarantees. Nevertheless, the value of these guarantees when the controller is implemented in practice has to be related to the effort it takes to achieve them. Furthermore, to facilitate practical application of the available LPV controller synthesis methods, research to more efficient and faster algorithms is desirable. Finally, implementation on an actual vehicle electronic control unit is not considered in this research. Instead, a rapid control prototyping platform is used to implement the controllers. Hence, future research could focus on the feasibility of implementation, considering, for example, the required online computing power and the required amount of memory.

CHAPTER 4

Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach[1]

Abstract - The combination of different characteristics and situation-dependent behavior cause the design of adaptive cruise control (ACC) systems to be time consuming. This chapter presents a systematic approach for the design of a parameterized ACC, based on explicit model predictive control. A unique feature of the synthesized ACC is its parameterization in terms of key characteristics, which, after the parameterization, makes it easy and intuitive to tune, even for the driver. The effectiveness of the design approach is demonstrated using simulations for relevant traffic scenarios, including stop-&-go. On-the-road experiments show the proper functioning of the synthesized ACC.

4.1 Introduction

Adaptive cruise control (ACC) is an extension of classic cruise control (CC), which is a widespread functionality in modern vehicles. Starting in the late 1990s with luxury passenger cars, ACC functionality is now available in numerous commercially available passenger cars as well as trucks. The objective of classic CC is to control the longitudinal vehicle velocity by tracking a desired velocity determined by the driver. Only the throttle

[1]This chapter is based on G. J. L. Naus, J. Ploeg, M. J. G. van de Molengraft, W. P. M. H. Heemels and M. Steinbuch (2010). Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach. *Control Eng. Pract.*, 18(8), p. 882-892.

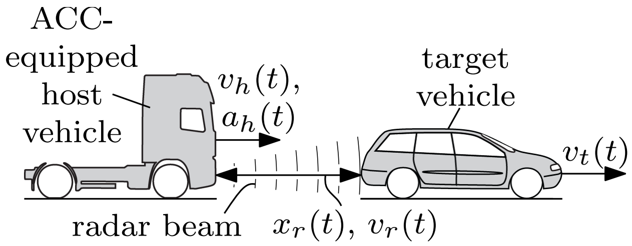


Figure 4.1: Example of the ACC working principle. The host vehicle, driving with velocity $v_h(t)$ and acceleration $a_h(t)$, is equipped with an ACC, which ensures automatic following of the preceding target vehicle, driving with velocity $v_t(t)$. A radar measures the distance $x_r(t)$ and the relative velocity $v_r(t) = v_t(t) - v_h(t)$ between the vehicles.

is used as an actuator. ACC extents classic CC functionality, by automatically adapting the velocity if there is a preceding vehicle, using the throttle as well as the brake system. Commonly, a radar, using radio waves, or a lidar, using laser light, is used to detect preceding vehicles, measuring the distance $x_r(t)$ and the relative velocity $v_r(t)$ between the vehicles. Hence, besides classic CC functionality, ACC enables also automatic following of a predecessor. In Figure 4.1, a schematic representation of the working principle of ACC is shown, where $v_h(t)$ and $a_h(t)$ are the velocity and the acceleration of the ACC-equipped host vehicle, respectively, and $v_t(t)$ is the velocity of the preceding target vehicle.

ACC systems typically consist of two parts: a vehicle-independent part and a vehicle-dependent part (Prestl et al., 2000; Moon et al., 2009). The vehicle-independent part determines a desired acceleration/deceleration profile for the vehicle. Subsequently, the vehicle-dependent part ensures tracking of this profile via actuation of the throttle and brake system. Hence, the latter part can be regarded as a controller for the longitudinal vehicle acceleration. The vehicle-independent part and the vehicle-dependent part form an outer and an inner control loop, respectively. In Figure 4.2, a schematic representation of these control loops is shown, where $v_{CC}(t)$ is the desired cruise control velocity, $a_{h,d}(t)$ is the desired acceleration, and $u_{th}(t)$ and $u_{br}(t)$ are the throttle and the brake system control signals, respectively. This chapter addresses the design of the vehicle-independent part of an ACC.

Focusing on the outer control loop, i.e., the vehicle-independent part, the primary control objective is to ensure following of a preceding vehicle. Considering the corresponding desired driving behavior, ACC systems are generally designed to have specific key characteristics, such as safety, comfort, fuel economy and traffic-flow efficiency (Vahidi and Eskandarian, 2003). In general, however, these characteristics typically impose conflicting objectives and introduce constraints, complicating the controller design. For instance, to ensure safe following, the system should be agile, requiring high acceleration and deceleration levels, which is not desirable concerning comfort or fuel economy (Moon

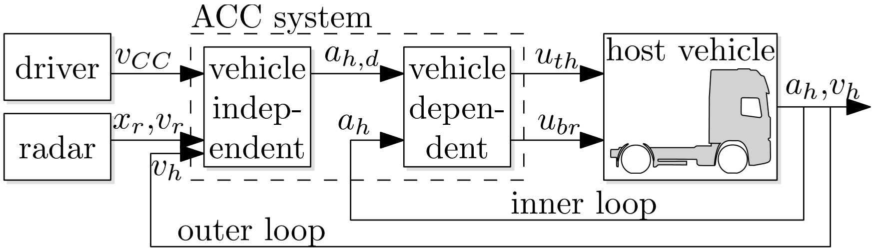


Figure 4.2: Schematic representation of the ACC control loop. For clarity, the time dependency of the signals is omitted. The ACC is divided into a vehicle-independent, outer control loop determining a desired acceleration $a_{h,d}(t)$ and a vehicle-dependent, inner control loop determining the throttle and brake system control signals $u_{th}(t)$ and $u_{br}(t)$, respectively. The distance $x_r(t)$ and relative velocity $v_r(t)$ with respect to the preceding vehicle are measured using a radar. The driver switches the ACC on and off, regulates characteristic system settings and determines a desired cruise control velocity $v_{CC}(t)$.

and Yi, 2008). To account for different characteristics, such as safety and comfort, a weighted optimization can be employed. For example, a model predictive control (MPC) approach may be adopted, which also facilitates taking into account constraints (Corona and De Schutter, 2008).

Besides specific key characteristics, driver acceptance of the system requires ACC behavior to mimic human driving behavior to some extent (Driel et al., 2007). Apart from the fact that human driving behavior is driver specific and variable in time, it is also situation dependent. Generally, situation-dependent behavior is incorporated in the ACC in an ad-hoc manner, by switching between different modes according to different situations. This switching is either based on logic rules, using a specific tuning for each mode (Moon et al. 2009, Widmann et al. 2000, Persson et al. 1999), or nonlinear filters are employed to combine all modes (e.g., Zhang and Ioannou, 2004; Yanakiev et al., 1998). Another, more crude method is to ignore specific traffic situations or consider them separately. For example, slow driving or standing still is only incorporated if so-called stop-&-go (SG) functionality is included (Venhovens et al., 2000).

The key characteristics and the desired situation dependency of the ACC design give rise to many tuning variables. This makes the design and tuning time consuming and error prone. In this chapter, a systematic procedure for the design and tuning of the vehicle-independent part of an ACC is presented. The contribution is the design of an ACC which is parameterized by the key characteristics, with at most one tuning variable for each characteristic. Hence, after the parameterization, the specific setting of the ACC can easily be changed, possibly even by the driver. Next to presenting this systematic design approach, the implementation of the ACC and the results of on-the-road experiments are discussed.

An explicit model predictive control (MPC) synthesis is adopted to design the ACC, following Corona and De Schutter (2008) and Möbus et al. (2003). One reason to use the MPC synthesis is that it enables to take into account conflicting controller requirements as well as possible constraints imposed by the key characteristics of the system. A second reason is that, when implemented in a receding horizon fashion, an optimization problem is solved in every time step. This enables the controller to adapt to actual working conditions, i.e. traffic situations, and, as such, the controller is situation dependent. For the implementation, it is desirable to solve the optimization problem offline in an explicit manner via a multi-parametric program, instead of direct online implementation of the controller. This yields an explicit, piecewise affine (PWA) control law (Bemporad et al., 2002b,a).

The organization of the chapter is as follows. The problem formulation is presented in Section 4.2. In Sections 4.3 and 4.4, the controller design and the corresponding tuning, including the parameterization of the controller are discussed. The implementation, experimental results and the working of the parameterization are presented in Section 4.5. Finally, conclusions are given and related work is introduced.

4.2 Problem formulation

4.2.1 Quantification measures

In this case, safety and comfort are chosen as the key characteristics of the desired behavior of an ACC. Considering safety, however, it has to be remarked that the ACC is not a safety system such as an emergency braking system or a collision avoidance system. ACC is primarily a comfort system that incorporates safety in the sense that appropriate driving actions within surrounding traffic are guaranteed. To enable quantification of the key characteristics, desirable properties of these characteristics, so-called quantification measures, have to be defined.

The safety of the driving behavior is typically related to the inter-vehicle distance and the relative velocity of the vehicles (Naus et al., 2008b). Typically, the safety of a traffic situation increases for an increasing inter-vehicle distance and a decreasing relative velocity. Furthermore, higher deceleration levels are beneficial, as a wider range of traffic situations can be handled in a safe manner. A high deceleration level, however, is clearly no quantification measure related to safety. It only enables to drive in a safe manner. Hence, regarding safety, the inter-vehicle distance and the relative velocity will be used as quantifications measures.

The comfort of a driving action is often related to the number, size and frequency of vibrations or oscillations in the longitudinal acceleration of the vehicle due to, for example, external disturbances, engine torque peaks, driveline characteristics, etc. (Dorey et al., 2001; International Organization for Standardization, 1997; Mo et al., 1996). Besides that, in practice, maximum deceleration values are often related to the comfort of ACC systems (Motor Presse Stuttgart GmbH & Co. KG, 2006). Furthermore, the (peak) jerk levels are often considered as a measure to reflect human's comfort (Martinez and Canudas-de-Wit, 2007). In designing trains and elevators for example, the jerk is typically limited to 2.0 m/s^3. Hence, regarding comfort, the (peak) acceleration and (peak) jerk levels will be used as quantification measures (Naus et al., 2008b).

4.2.2 Parameterization

This chapter presents the design of a parameterized ACC, with, at the end, only a few design parameters, i.e. tuning knobs, that are directly related to the key characteristics of the behavior of the ACC. The limited number of intuitive tuning variables enables quick and easy adaptation of the ACC to specific desirable driving behavior. Importantly, these variables can also be used by non-experts in (MPC) control, like the driver, to change the behavior of the ACC system. Enabling the driver to set these variables, really makes the ACC driver dependent.

An explicit MPC approach is used to design the parameterized ACC. The MPC synthesis accommodates constraints, an optimal situation-specific controller results when implemented in a receding horizon fashing, and the minimization of a cost criterion enables making trade-offs between conflicting characteristics. However, a disadvantage of the MPC synthesis is the large number of tuning parameters, which follow from the definition of the control objective, the constraints and the choice of the cost criterion.

To obtain an ACC with only a few, intuitive design parameters, the design parameters P_s and P_c are defined, indicating to what extent the driving behavior of an ACC-controlled vehicle is either safe or comfortable, with $P_s \in [0, 1]$ and $P_c \in [0, 1]$, where larger values for P_s and P_c indicate an increase in safety and comfort, respectively. The many tuning parameters of the MPC setup are used to map the quantification measures to these design parameters, which are directly related to the key characteristics of the ACC, in this case safety and comfort. Incorporating P_s and P_c in the controller design yields a parameterized ACC, i.e. ACC(P_s, P_c), with P_s and P_c as tuning variables directly related to the desired behavior of the ACC.

Hence, depending on the driver, the design parameters P_s and P_c can be chosen to accommodate a driver's desirable setting. For comparison, in most commercially available ACC systems, the desired distance is the only parameter a driver is able to vary, adjusting the behavior of the ACC. The parameterized ACC enables the driver to actually change the total behavior of the system with respect to the key characteristics. The systematic approach presented here, makes it possible to redesign the system relatively easily, for example for different key characteristics, and reduces the amount of time-consuming and error-prone trial-and-error techniques in the design. The approach is general and can be adopted for any characteristics, although focus lies on safety and comfort in this case.

4.3 MPC controller design

4.3.1 Modeling

The MPC synthesis requires a model of the relevant dynamics to use as a prediction model. Consider the control structure as presented in Figure 4.2. Focusing on the design of the vehicle-independent control part, the model should cover the longitudinal host vehicle dynamics, the vehicle-dependent control part and the longitudinal relative dynamics, which are measured by the radar. Assuming that the vehicle-dependent control part ensures perfect tracking of the desired acceleration $a_{h,d}(t)$, the internal vehicle dynamics and the vehicle-dependent control part together can be modeled by a single integrator, relating the host vehicle velocity $v_h(t)$ to the desired acceleration $a_{h,d}(t)$. This yields the following set of linear equations

$$\left\{ \begin{array}{lcl} x_r(t) & = & x_r(0) + \int_{t_0}^{t} v_r(t)dt \\ v_r(t) & = & v_r(0) + \int_{t_0}^{t} a_r(t)dt \\ v_h(t) & = & v_h(0) + \int_{t_0}^{t} a_h(t)dt \end{array} \right. \tag{4.1}$$

where $x_r(t)$ the relative position, $v_r(t) = v_t(t) - v_h(t)$ the relative velocity, $a_r(t) = a_t(t) - a_h(t)$ the relative acceleration, $v_h(t)$ the host vehicle velocity, and $a_h(t)$ the host vehicle acceleration at time $t \in \mathbb{R}^+$. The values of $x_r(t)$ and $v_r(t)$ are measured by the radar and measurements of $v_h(t)$ and $a_h(t)$ are standard available. The acceleration of the target vehicle $a_t(t)$ is not measured, nor is it predicted. The signal to noise ratio of the radar measurement is too small to use for derivation of the target-vehicle acceleration. Furthermore, wireless inter-vehicle communication or the use of, e.g., the global positioning system (GPS) in combination with a geographical information system (GIS) and map-matching algorithms to enable prediction of $a_t(t)$, are not considered in this research. Hence, $a_t(t)$ is, as a nominal case, assumed to be zero for the MPC prediction model, yielding $a_r(t) = -a_h(t)$. As a result, $a_t(t)$ will act as a disturbance on the system.

The MPC algorithm is commonly designed and implemented in the discrete-time domain. Hence, the linear, continuous-time equations (4.1) are converted into a discrete-time model using a zero-order hold assumption on $a_h(t)$ and an exact discretization method with sample time T_s. The signals are considered at the sampling times $t = k\,T_s$ where $k \in \mathbb{N}$ represents the discrete time steps:

$$\boldsymbol{x}(k+1) = \mathbf{A}\boldsymbol{x}(k) + \mathbf{B}a_h(k), \quad k \in \mathbb{N} \tag{4.2}$$

where $\boldsymbol{x}(k) = (x_r(k), v_r(k), v_h(k))^T$, and

$$\mathbf{A} = \begin{pmatrix} 1 & T_s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} -\frac{1}{2}T_s^2 \\ -T_s \\ T_s \end{pmatrix} \tag{4.3}$$

Considering the control structure as presented in Figure 4.2, and assuming perfect tracking of the desired acceleration $a_{h,d}(k)$, the host vehicle acceleration $a_h(k) = a_{h,d}(k)$ can be regarded as the control input $u(k)$. Furthermore, as all states of $\boldsymbol{x}(k)$ are measured, the output equation becomes $\boldsymbol{y}(k) = \boldsymbol{x}(k),\ k \in \mathbb{N}$. The total model then becomes

$$\mathcal{M} : \begin{cases} \boldsymbol{x}(k+1) &= \mathbf{A}\boldsymbol{x}(k) + \mathbf{B}u(k) \\ \boldsymbol{y}(k) &= \boldsymbol{x}(k) \end{cases}, \quad k \in \mathbb{N} \tag{4.4}$$

where $u(k) = a_h(k)$ and $\mathbf{A}$ and $\mathbf{B}$ as defined in (4.3).

4.3.2 Control objectives and constraints

The primary control objective of an ACC is to ensure automatic following of preceding traffic. Typically, this primary control objective amounts to following a target vehicle at a desired distance $x_{r,d}(k)$. Often, a so-called desired headway time h_d is used to define this desired distance, yielding

$$x_{r,d}(k) = x_{r,0} + v_h(k)\, h_d \tag{4.5}$$

with $x_{r,0}$ a constant representing the desired distance at standstill, and the desired headway time h_d a measure for the time it takes to reach the current position of the preceding vehicle if the host vehicle continues to drive with its current velocity, i.e. for constant $v_h(k)$. Correspondingly, the tracking error at discrete time $k \in \mathbb{N}$ is defined as $e(k) = x_{r,d}(k) - x_r(k)$. Hence, the primary control objective $\mathcal{O}_1$ comes down to minimizing the absolute tracking error $|e(k)|,\ k \in \mathbb{N}$.

Besides the primary control objective $\mathcal{O}_1$, several secondary objectives, related to the key characteristics, in this case safety and comfort, have to be included. These secondary objectives are based on the quantification measures discussed in Section 4.2.1. Regarding safety, the primary control objective $\mathcal{O}_1$ deals with the relative position. Besides control of the relative position, the relative velocity $|v_r(k)|$ should be made small. Regarding the comfort of a driving action, the peak values of the host vehicle acceleration $|a_h(k)|$ and jerk $|j_h(k)|$ should be kept small. Hence, next to $\mathcal{O}_1$, $|v_r(k)|$, $|a_h(k)|$ and $|j_h(k)|$ should all be small. Using the MPC setup, the objectives are incorporated in a weighted form in an optimization criterion, such that a tradeoff can be made between them.

Besides the control objectives, the key characteristics introduce several constraints, which have to be included in the MPC setup. For safety, the inter-vehicle distance should always be positive to avoid collisions. For comfort, the absolute value of the acceleration of the host vehicle $|a_h(k)|$ and the absolute value of the jerk $|j_h(k)|$ are constrained. The constraints on the jerk are given by $|j_h(k)| \leq j_{h,max}$, where $j_{h,max}$ is an appropriately chosen positive constant. The constraints on the acceleration are more involved. For comfort reasons, high accelerations at high velocities should be prohibited. At the same time, however, quickly driving off from standstill should be possible. Hence, the constraint on the maximum acceleration $a_{h,max}$ is chosen to depend affinely on the velocity of the host vehicle, i.e. $a_{h,max}(v_h(k)) = a_{h,0} - \alpha_0 v_h(k)$, where both $a_{h,0}$ and α_0 are appropriately chosen positive constants, such that $a_{h,max}$ decreases for increasing $v_h(k)$. To guarantee safe operation with respect to erroneously detected objects, the host vehicle minimum acceleration $a_{h,min}$ is, by legislation, confined to $a_{h,min} = -3.0$ m/s^2 (International Organization for Standardization, 2002).

To accommodate the constraint on $|j_h(k)|$, as well as to enforce integral action preventing steady-state errors in, for example, the following distance, the original input-output model $\mathcal{M}$ (4.4) is converted into an incremental input-output (IIO) model $\mathcal{M}_e$ (Maciejowski, 2002)

$$\mathcal{M}_e : \begin{cases} \boldsymbol{x}_e(k+1) &= \mathbf{A}_e \boldsymbol{x}_e(k) + \mathbf{B}_e \delta u(k) \\ \boldsymbol{y}_e(k) &= \boldsymbol{x}_e(k) \end{cases}, \quad k \in \mathbb{N} \tag{4.6a}$$

where $\boldsymbol{x}_e(k) = (\boldsymbol{x}^T(k), u(k-1))^T$ the new state vector, $\delta u(k) = u(k) - u(k-1)$ the new control input, and

$$\mathbf{A}_e = \begin{pmatrix} 1 & T_s & 0 & -\frac{1}{2}T_s^2 \\ 0 & 1 & 0 & -T_s \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{B}_e = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{4.6b}$$

the new model matrices. The variation in the control output $\delta u(k)$ is now used as a measure for the jerk $j_h(k)$. Correspondingly, the constraint on the jerk is transformed into $|\delta u(k)| \leq j_{h,max}$. Summarizing, the constraints are given by:

$$\mathcal{C} : \begin{cases} x_{r,min} < x_r(k) \\ a_{h,min} \leq u(k) \leq a_{h,max}(v_h(k)) \\ |\delta u(k)| \leq j_{h,max} \end{cases}, \quad k \in \mathbb{N} \tag{4.7}$$

where $u(k) = a_{h,d}(k) = a_h(k)$, and $x_{r,min} \leq 0$ the minimal inter-vehicle distance. The model $\mathcal{M}_e$ (4.6) is used as the MPC prediction model in the remainder of this chapter.

4.3.3 Control problem / cost criterion formulation

As MPC is used, a cost criterion J, which is minimized over a prediction horizon N_y, has to be defined. The future system states are predicted using the model $\mathcal{M}_e$ (4.6) and the current state $\boldsymbol{x}_e(k|k) := \boldsymbol{x}_e(k)$ at discrete time step k as initial condition. This yields the predicted states $\boldsymbol{x}_e(k+n|k)$ and the predicted tracking error $e(k+n|k)$, $n = 0, 1, \dots, N_y$ for a selected input sequence $\boldsymbol{\delta U}(k|k) = (\delta u(k|k), \dots, \delta u(k+N_y-1|k))^T$, starting at discrete time step k. Based on the prediction of the future system states, the minimization problem yields an optimal control sequence, subject to constraints (4.7) on the inputs and the outputs.

The cost criterion is typically formulated as a linear or as a quadratic criterion. To solve the resulting problem, the criterion is casted into a linear program (LP) or a quadratic program (QP). Finding the solution of an LP is less computationally demanding than the corresponding solution of a QP. However, the optimal solution of an LP is, generally, at the intersection of constraints, i.e., at a boundary of the feasible region. Hence, the solution is sensitive for changes in the constraints rather than tuning of the relative weights in the cost function. The optimal solution of a QP, on the other hand, can be moved intuitively within the feasible region, as this solution is sensitive for the tuning of the relative weights. The practical drawbacks in the tuning of linear formulations explains why MPC is often formulated using a quadratic criterion (Maciejowski, 2002; Rao and Rawlings, 2000). Consequently, a quadratic criterion is used, which is defined by

$$\begin{aligned} J(\boldsymbol{\delta U}(k|k), \boldsymbol{x}_e(k)) = \sum\nolimits_{n=1}^{N_y} \left(\boldsymbol{\xi}^T(k+n|k)\,\mathbf{Q}\,\boldsymbol{\xi}(k+n|k)\right) + \dots \\ \dots \sum\nolimits_{n=0}^{N_u-1} \left(\delta u^T(k+n)\,R\,\delta u(k+n)\right) \end{aligned} \tag{4.8}$$

with $\boldsymbol{\xi}(k+n|k) \triangleq (e(k+n|k), v_r(k+n|k), a_h(k+n|k))^T$ a column vector incorporating the control objectives, with $a_h(k+n|k) = u(k+n|k)$, and $\mathbf{Q} = \mathrm{diag}(Q_e, Q_{v_r}, Q_{a_h})$ and $R = Q_{j_h}$ the weights on the tracking error and the secondary control objectives. Furthermore, N_y and N_u denote the output and the control horizon, respectively, where $N_u \le N_y$. Moreover, for $N_u \le n < N_y$ the control signal is kept constant, i.e. $\delta u(k+n|k) = 0$ for $N_u \le n < N_y$. Finally, $u(k+n) = u(k+n-1|k) + \delta u(k+n|k)$, for $n \ge 0$.

Given a full measurement of the state $\boldsymbol{x}_e(k)$ of the model $\mathcal{M}_e$ (4.6) at the current time k, the MPC optimization problem at time k is formulated as

$$\begin{aligned} \underset{\boldsymbol{\delta U}(k|k)}{\text{minimize}} \quad & J(\boldsymbol{\delta U}(k|k), \boldsymbol{x}_e(k)) \\ \text{subject to} \quad & \text{the dynamics } \mathcal{M}_e \text{ (4.6)} \\ & \text{the constraints } \mathcal{C} \text{ (4.7)} \end{aligned} \tag{4.9}$$

The controller will be implemented in a receding horizon manner meaning that at every time step k, an optimal future input sequence $\boldsymbol{\delta U}^*(k|k) = (\delta u^*(k|k), \dots, \delta u^*(k+N_y-1|k))^T$ is computed in the sense of the minimization problem (4.9). The first component of this vector, $\delta u^*(k|k)$, is used to compute the new optimal control output $u^*(k) = u(k-1) + \delta u^*(k|k)$. This $u^*(k)$ is applied to the system, after which the optimization (4.9) is performed again for the updated measured state $\boldsymbol{x}_e(k+1) = (\boldsymbol{x}^T(k+1), u(k))^T$.

4.3.4 Explicit MPC

For the implementation, it is desirable to have an explicit MPC control law $\delta u^*(k) = \mathcal{K}(\boldsymbol{x}_e(k))$, instead of an implicit one obtained through online solving of the optimization problem (4.9) at each time step. Solving (4.9) as a multi-parametric quadratic program (mpQP) with parameter vector $\boldsymbol{x}_e$ enables an explicit form of the solution by offline optimization. As a result, the online computation time can be reduced, and it is easier to define an upper bound on the required computation time. Furthermore, the explicit controller is of low complexity, yielding easily verifiable code, which justifies the implementation in embedded, safety-critical systems. The resulting explicit controller inherits all stability and performance properties of the original implicit controller and has the form of a piecewise affine (PWA) state feedback law (Bemporad et al., 2002a,b). A disadvantage of the offline optimization is that it prohibits online tuning of the controller. The controller has to be tuned offline after which a new explicit solution has to be computed, which can be implemented online.

Solving the mpQP provides a set $\mathcal{X}_f \subseteq \mathbb{R}^{n_x}$, with n_x the dimension of $\boldsymbol{x}_e$, of states for which the constrained optimization problem (4.9) is feasible. Since the control law is given by a PWA state feedback law, the feasible set $\mathcal{X}_f$ is partitioned into n_R polyhedral regions $\mathcal{R}_i$, $i = 1, \ldots, n_R$, such that

$$\mathcal{X}_f = \bigcup_{i=1}^{n_R} \mathcal{R}_i \tag{4.10}$$

where $\mathrm{int}\mathcal{R}_i \cap \mathrm{int}\mathcal{R}_j = \emptyset$, for $i = 1, \ldots, n_R$, $j = 1, \ldots, n_R$ and $i \neq j$. At time step k, the optimal input $\delta u^*(k|k)$ is then given by

$$\delta u^*(k|k) = \mathbf{F}_{0,i}\boldsymbol{x}_e(k) + f_{0,i}, \text{ for } \boldsymbol{x}_e(k) \in \mathcal{R}_i, \; i = 1, \ldots, n_R \tag{4.11}$$

where $\mathbf{F}_{0,i}$ a matrix, and $f_{0,i}$ a constant. Hence, to compute the control input at discrete time step $k \in \mathbb{N}$, (4.11) has to be evaluated, in which the most time-consuming part is the determination of the region $\mathcal{R}_i$ that contains $\boldsymbol{x}_e(k)$. Implementation of the implicit controller requires solving an optimization in every time step, which is computationally often more demanding.

The state space, which is explored when solving the mpQP, is limited by imposing a polytopic constraint $\mathcal{C}_i$ on the initial state $\boldsymbol{x}_e(k|k)$. This polytope is defined by

$$\mathcal{C}_i : \begin{cases} 0 < x_r(k|k) \leq x_{rr} \\ 0 \leq v_h(k|k) \leq v_{h,max} \\ 0 \leq v_t(k|k) \leq v_{t,max} \\ u_{min} \leq u(k-1|k) \leq u_{max}(v_h(k-1|k)) \end{cases} \tag{4.12}$$

where x_{rr} the radar range, $v_{h,max}$ the maximum host vehicle velocity and $v_{t,max}$ the maximum target vehicle velocity. As the relative velocity is defined as $v_r(k) = v_t(k) - v_h(k)$,

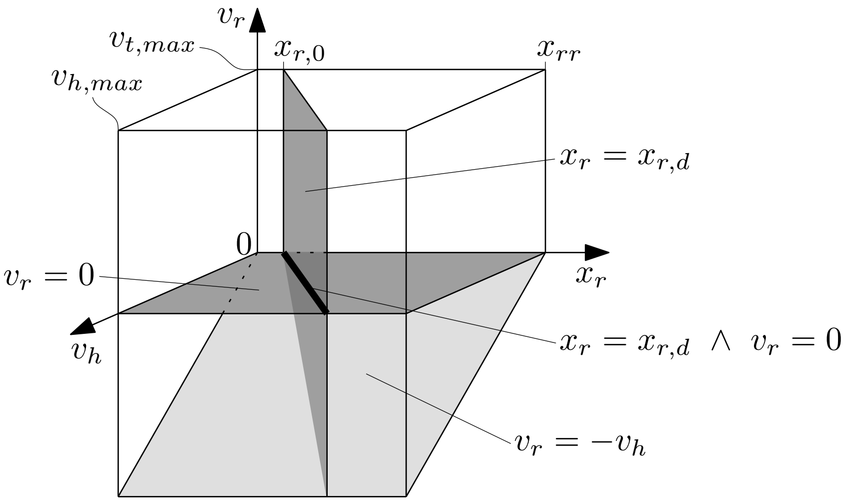


Figure 4.3: Visualization of a 3D crosscut of the state space $\boldsymbol{x}_e$ for constant $x_{e,4} = u$, including the constraint $\mathcal{C}_i$ on the initial state (4.12).

combination of the constraints on $v_h(k|k)$ and $v_t(k|k)$ yields a constraint on the initial state of the relative velocity $-v_h(k|k) \leq v_r(k|k) \leq v_{t,max}$. In Figure 4.3, a crosscut at constant $x_{e,4} = u$ of the state space including the constraint $\mathcal{C}_i$ on the initial state (4.12), is shown.

4.4 Controller parameterization

4.4.1 Approach

The MPC controller design incorporates a significant number of MPC tuning parameters. These tuning parameters are given by the desired headway time h_d, the constraints on the acceleration and the jerk, $a_{h,min}$, $a_{h,max}$ and $j_{h,max}$, respectively, the weights $\mathbf{Q} = \mathrm{diag}(Q_e, Q_{v_r}, Q_{a_h})$ and $R = Q_{j_h}$, and the control and prediction horizons N_u and N_y. Correspondingly, define the tuple $\mathcal{T}_{\mathrm{MPC}}$, containing the MPC tuning parameters

$$\mathcal{T}_{\mathrm{MPC}} = (h_d, a_{h,min}, a_{h,max}, j_{h,max}, \mathbf{Q}, R, N_u, N_y) \tag{4.13}$$

The goal is to relate these tuning parameters explicitly to the essential design parameters P_s and P_c, which is based on a mapping of the corresponding quantification measures, yielding $\mathcal{T}_{\mathrm{MPC}}(P_s, P_c)$.

The design parameters P_s and P_c are directly related to the characteristics of the driving behavior, indicating to what extent the driving behavior is either safe or comfortable (see Section 4.2.2). Furthermore, a combination of the operating ranges of the quantification measures can be regarded as a representation of the total operating range of the ACC system. Consequently, using the MPC tuning parameters, i.e., the elements of $\mathcal{T}_{MPC}$ to map the operating ranges of the quantification measures to those of the design parameters, being just $P_s \in [0,1]$, $P_c \in [0,1]$, enables intuitive tuning of the ACC system, directly related to the desired key characteristics. The design of these relations is, however, not trivial.

First, the operating ranges of the quantification measures are not clearly defined in all cases. For safety, a maximum deceleration of -3.0 m/s^2 is defined by legislation and a minimal inter-vehicle distance $x_r > x_{r,min}$ is included in the constraints $\mathcal{C}$ (4.7). However, for comfortable driving behavior, the operating ranges of the corresponding quantification measures are less clearly defined. For example, the maximum allowable acceleration or jerk for comfortable driving behavior in different situations are not specified unambiguously.

Although it is not the focus of this research, in this case benchmark measurements are used to determine reasonable operating ranges. The benchmark measurements involve on-the-road testing of various traffic scenarios by a preferably large panel of test drivers. In this case, experiments with a commercially available ACC SG system are conducted. Nevertheless, the experiments are conducted with only a limited number of drivers. Hence, the resulting tuning will probably not be representative of general human driver behavior. The exact tuning values will therefore not be discussed in detail.

Second, the mapping between the quantification measures and the design parameters P_s and P_c is not straightforwardly related to the MPC tuning parameters that are contained in $\mathcal{T}_{\text{MPC}}$. For example, the setting of one tuning parameter may influence the mapping of several quantification measures, and the mapping of a single quantification measure is, in general, influenced by the settings of several tuning parameters.

Hence, the design of the relationships between the MPC tuning parameters, i.e., the elements of $\mathcal{T}_{MPC}$ on the one hand, and the design parameters P_s and P_c on the other hand, mapping the operating ranges of the quantification measures to those of the corresponding design parameters, can be regarded as a tuning step. This step requires engineering work, and has to be done manually. Hence, this step can be regarded as a non-systematic step in the design of the parameterized adaptive cruise control. However, the design has to be done only once, fixing the MPC tuning parameters that are contained in $\mathcal{T}_{\text{MPC}}$ as a function of the essential design parameters P_s and P_c.

The design of the relations between the MPC tuning parameters and the design parameters is discussed in detail next. For simplicity, in this case affine relationships are used between the elements of $\mathcal{T}_{\text{MPC}}$ on the one hand, and P_s and P_c on the other hand. Furthermore, the control and prediction horizons are taken constant and equal, i.e., $N_y = N_u = c$.

4.4.2 Application

Safety

The quantification measures related to safety are the distance and the relative velocity (see Section 4.2.1). The desired distance is translated into a desired headway time h_d, which is typically varied between 1.0 and about 2.0 s (see, e.g., Prestl et al., 2000). Human driving behavior shows a somewhat wider range, in between 0.5 and about 2.5 s (Moon and Yi, 2008). The larger the headway time, the more time the controller has to react to a certain traffic situation. Besides that, if the controller cannot handle a specific situation appropriately, the driver has more time to intervene. Hence, the larger the headway time, the safer the driving will be. A corresponding relationship between h_d and P_s is $h_d = 0.5 + 2\,P_s$, yielding $h_d \in [0.5, 2.5]$, which is a sufficiently large range.

Furthermore, the weight Q_e, which is the weight on the error $e(k)$ between the desired and the actual distance, has to be considered. The larger Q_e, the smaller the time to reach a steady-state situation, i.e., $e(k) = 0$, which is desirable regarding safety. A corresponding relation between Q_e and P_s is $Q_e = q_{e,0}\,P_s$ with $q_{e,0} > 0$ a positive constant. Although the focus is on safety, it has to be remarked that for increasing Q_e, the acceleration and deceleration peaks will increase as well, which indicates less comfortable driving behavior.

Finally, the relative velocity $v_r(k)$ should be minimized as fast as possible, which is influenced by the weight Q_{v_r}. In Figure 4.4, simulation results are shown for variable Q_{v_r}, corresponding to of approach of a vehicle driving at constant velocity. At 22.3 s, a preceding vehicle, which is driving slower than the ACC-equipped host vehicle, enters the radar range and is detected. As the results show, the time at which the controller starts decreasing $v_h(k)$ and, accordingly, $v_r(k)$, decreases for increasing Q_{v_r}. This is desirable regarding safety. Counter-intuitively, however, the time it takes to reach a steady-state situation, in which $v_r(k) = 0$, decreases simultaneously, which is not desirable regarding safety. This is a result of the fact that the error between the desired distance and the actual distance is also minimized, which is only possible for $v_r(k) \neq 0$. Hence, whether increasing or decreasing Q_{v_r} increases or decreases the safety of the driving behavior, depends on the situation. Consequently, a constant value $Q_{v_r} = q_{v_r,0} > 0$ is adopted, ensuring on average desirable behavior.

Comfort

The quantification measures related to comfort are the peak acceleration and jerk levels (see Section 4.2.1). The sizes of the weights Q_{a_h} and Q_{j_h} are naturally related to the sizes of the resulting acceleration and jerk peak values and, hence, to the amount of comfort. The higher Q_{a_h} and Q_{j_h}, the lower the corresponding acceleration and jerk peak

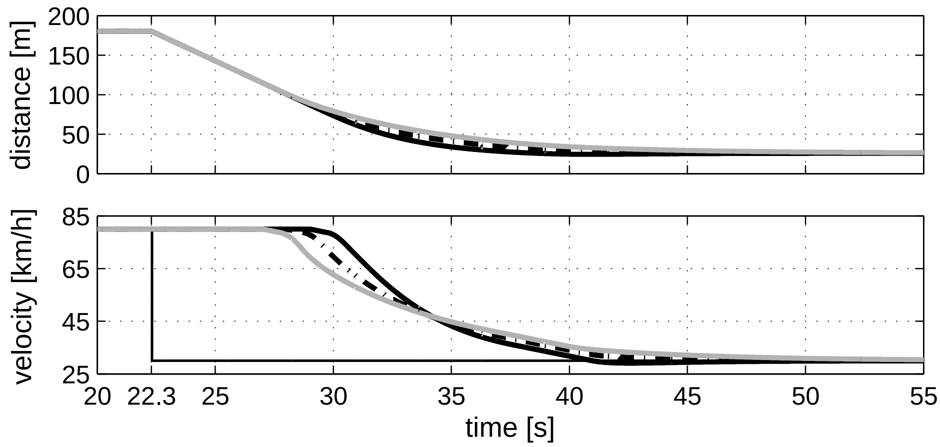


Figure 4.4: The distance x_r and the host vehicle velocity v_h, corresponding to *the approach of a vehicle driving with constant velocity*. The solid black, the dash-dotted black and the solid grey lines represent the results for increasing Q_{v_r}, respectively. The narrow black line in the lower figure represents the measured target vehicle velocity v_t.

values are and, consequently, the more comfortable the driving behavior is. This yields $Q_{a_h} = q_{a_h,0}P_c$ and $Q_{j_h} = q_{j_h,0}P_c$, with $q_{a_h,0} > 0$ and $q_{j_h,0} > 0$ positive constants. Also, the sizes of the constraint parameters $a_{h,max}(v(k))$, $a_{h,min}$ and $j_{h,max}$ are related to the amount of comfort. The smaller $a_{h,max}(v(k))$, $|a_{h,min}|$ and $j_{h,max}$, the smaller the maximum acceleration, deceleration and jerk values will be, and, hence, the more comfortable the driving behavior will be.

Following legislation, the maximum deceleration is limited to $a_{h,min} = -3.0$ m/s^2 (International Organization for Standardization, 2002; Driel et al., 2007). Analogously, in this research, the maximum acceleration is limited to $a_{h,max} = (3.0 - P_c)(1 - v_h(k)/v_{h,max})$, with $v_{h,max}$ the maximum vehicle velocity. This implies approximately full acceleration possibilities at low velocity, i.e. $a_{h,max} \in [2.0,\ 3.0]$ m/s^2, whereas this decreases linearly to $a_{h,max} = 0.0$ m/s^2 for $v_h(k) = v_{h,max}$. As the benchmark measurements did not provide distinctive limits for $j_{h,max}$, a constant $j_{h,max} = 3.0$ m/s^3 is adopted. Although the focus lies on comfort, it has to be remarked that tighter constraints on the maximum acceleration, deceleration and especially the jerk values implies that the reaction of the controller will be more sluggish, which will result in decreased safety.

4.4.3 Parameterization

Using affine relations, the MPC tuning parameters that are contained in $\mathcal{T}_{\text{MPC}}$ (4.13) are explicitly related to the key characteristics safety and comfort via the corresponding design parameters P_s and P_c (see Section 4.4.2). Consequently, the resulting tuning of the

ACC depends only on these two design parameters, as desired. Moreover, in this specific case, considering comfort and safety as key characteristics, it can be assumed that the key characteristics are complementary: the design of the relations shows that for increasing safety, comfort of the driving decreases, and vice versa. For example, small acceleration and jerk peak values, indicating a high level of comfort, induce a long time to steady state, which is not desirable regarding safety. Consequently, in this case of two key characteristics, a single parameter P results:

$$P = P_c, \quad P_c + P_s = 1, \quad P \in [0, 1] \tag{4.14}$$

If different or more than two characteristics would be considered in the design, typically more design parameters would remain in the end.

Parameterization of the ACC with safety and comfort amounts to incorporating the relations discussed in Section 4.4.2, accounting for (4.14), in the original optimization problem (4.9), which yields

$$\begin{aligned} \underset{\boldsymbol{\delta U}(k|k)}{\text{minimize}} \quad & J(P, \boldsymbol{\delta U}(k|k), \boldsymbol{x}_e(k)) \\ \text{subject to} \quad & \text{the dynamics } \mathcal{M}_e \ (4.6) \\ & \text{the constraints } \mathcal{C}(P) \ (4.7) \end{aligned} \tag{4.15}$$

Changing the behavior of the ACC system comes down to adjusting P. Allowing the driver to change $P \in [0, 1]$, enables individual drivers to influence the behavior of the controller, focusing on either comfortable or safe driving.

4.4.4 Explicit solution

The controller design is implemented via the multi parametric toolbox (MPT) (Kvasnica et al., 2006). As an explicit solution is desirable, an explicit controller is calculated offline by casting the problem as a multi-parametric program as discussed in Section 4.3.4. The result is a feedback control law as in (4.11), which is dependent on the state vector $\boldsymbol{x}_e(k) \in \mathbb{R}^{n_x}$ and the parameter P (4.14). To set the ACC to desirable behavior according to the driver's wishes using an implicit solution, setting of P can be done online. Using an explicit solution, however, the controller has to be recomputed offline. In this case, one might store various explicit controllers for a finite number of values $P = n/N_P$ for $n = \{0, 1, 2, \ldots, N_P\}$. Adopting this approach, the number of regions in the explicit ACC laws for the multi-parametric program as discussed in Section 4.3.4 ranges from 110 to 121. Computation of the piecewise affine maps with an Intel Pentium 2.13 GHz processor takes several seconds.

The amount of memory that is used for storing the explicit solution depends on the size of the look-up table and the number of points N_P that is used to discretize the continuous operating range of P. The size of the look-up table depends on the complexity of the

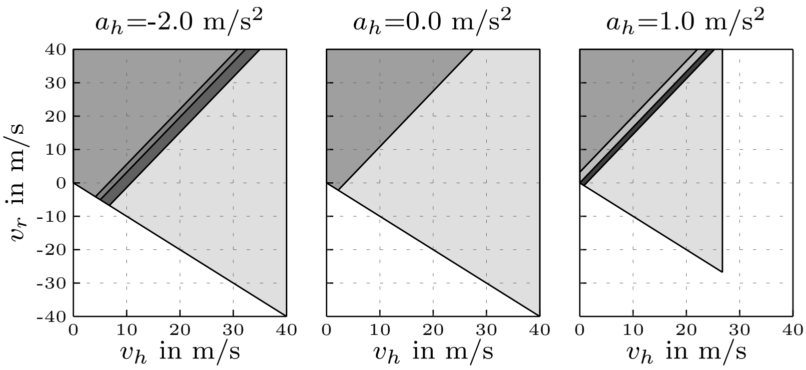


Figure 4.5: Three 2D crosscuts of the solution space at constant $x_r = 10.0$ m, for variable $x_{e,(4)} = a_h \in \{-2.0, 0.0, 1.0\}$ m/s^2 and $P = 0.1$.

problem, for which both the number of regions in the explicit ACC laws and the dimension of the solution space are indicators. For $N_P = 10$, a 4D solution space, and 110 to 121 regions per explicit ACC law, the amount of memory that is required for storing the explicit solutions comes down to approximately 6500 floats. The size and complexity of the piecewise affine maps are sufficiently small for fast online evaluation.

For illustration, three 2D crosscuts of the intersection $\mathcal{C}_i \cap \mathcal{X}_f$ are shown in Figure 4.5, where $x_{e,(1)} = x_r$ is constant, $x_{e,(4)} = u = a_h$ is variable, and $P = 0.1$. The grey areas represent different regions $\mathcal{R}_i$ with the same affine control law. The right plot in Figure 4.5 shows that the solution space $\mathcal{X}_f$ decreases as a result of $a_{h,max} = a_{h,max}(P, v_h(k), v_{h,max})$ (see Section 4.4.2).

4.5 Implementation and results

To enable actual implementation and corresponding evaluation of the ACC, additional functionality is required, being i) CC functionality, ii) the transition between CC functionality and ACC functionality, and iii) warning of the driver in a potentially dangerous situation. The design of this additional functionality is discussed first, after which the setup and results of simulations and on-the-road experiments are presented.

4.5.1 CC functionality

The overall ACC system combines both ACC and classic CC functionality. For classic CC functionality, tracking of a desired CC velocity v_{CC} is desired. Furthermore, driving in CC mode, the ACC system should switch automatically to ACC mode in case of a preceding vehicle driving slower than this desired CC velocity.

In this research, focus lies on the controller design for the ACC mode. To get CC func-

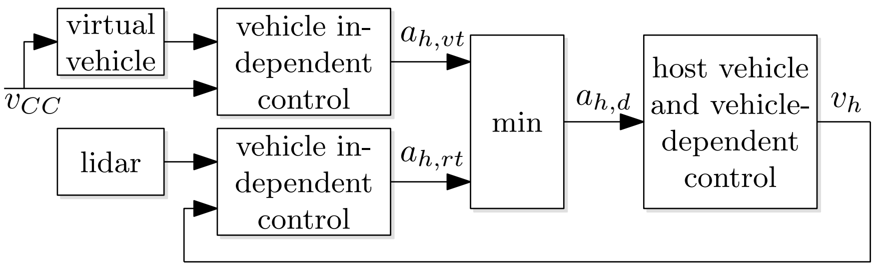


Figure 4.6: Implementation of the CC-ACC transition. A virtual vehicle, driving virtually at a velocity v_{CC}, mimics radar data. The corresponding control output $a_{h,vt}$ is compared to the control output corresponding to the real radar data, $a_{h,rt}$. Based on the minimum of both control outputs, the system switches between the virtual vehicle, which represents classic CC functionality, or the real radar data, i.e., ACC functionality.

tionality from this ACC design, a 'virtual target vehicle' is created, which virtually drives with a velocity equal to the desired CC velocity v_{CC} at the corresponding desired distance $x_r(k) = x_{r,d}(k, v_h(k))|_{v_h(k)=v_{CC}}$ with respect to the host vehicle. Using the virtual radar output corresponding to the position and velocity of this 'virtual target vehicle', instead of the actual radar output corresponding to a real target vehicle, the same explicit MPC solution can be used for both ACC and classic CC functionality. As a result, in CC mode, the same driving behavior is achieved as in ACC mode.

4.5.2 CC-ACC transition

For switching from ACC to classic CC functionality and vice versa, the common approach proposed in literature employs logic rules. Either the functionality yielding the lowest acceleration, i.e. the control input $u(k)$, is employed (Persson et al., 1999), or ACC functionality is employed if braking is required and classic CC otherwise (Zhang and Ioannou, 2004). To prohibit chattering, often, a boundary layer comprising hysteresis or a delay is assigned to the switching rules (Widmann et al., 2000).

The solution proposed here uses switching based on the lowest acceleration. As the acceleration is the control input, this ensures smooth transitions. The desired acceleration following from the preceding real target vehicle, which is used for ACC functionality, and the desired acceleration following from the virtual target vehicle, which is used for classic CC functionality, are compared. The lowest acceleration is used as the input, which is schematically shown in Figure 4.6.

Table 4.1: Envelope of traffic scenarios.

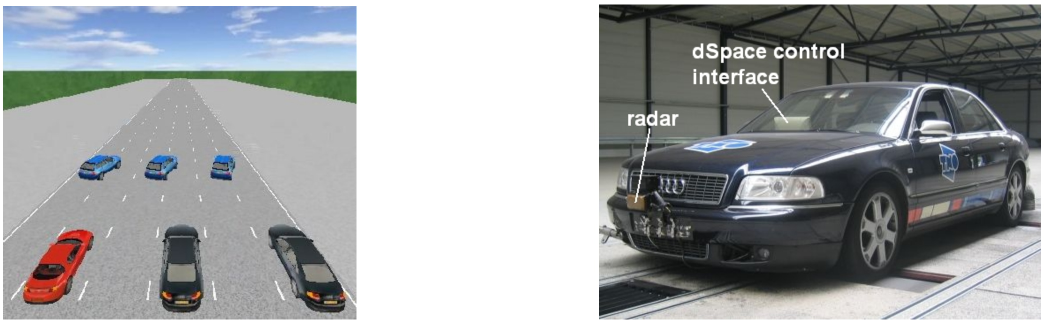
nr.	description
1.	Steady following of a target vehicle with a varying velocity.
2.	Approach of a vehicle at standstill or a vehicle driving with a constant velocity, yielding a CC to ACC switch.
3.	A cut in, which involves a sudden step in x_r such that $x_r < x_{r,d}$. For $v_r < 0$ and $v_r > 0$, this is called a negative and a positive cut in, respectively.
4.	A cut out, which involves a sudden step in x_r, yielding an ACC to CC switch.
5.	Following of a decelerating vehicle to standstill.
6.	Following of an accelerating vehicle, driving off at a traffic light, yielding an ACC to CC switch.
7.	Accelerating and decelerating in the CC mode due to changes in the CC velocity v_{CC}.

4.5.3 Positively invariant subset

Consider the intersection of the space defined by the constraints $\mathcal{C}_i$ on the initial state and the feasible state space $\mathcal{X}_f$ (4.10), i.e. $\mathcal{C}_i \cap \mathcal{X}_f$. All states $\boldsymbol{x}_e(k) \in \mathcal{C}_i \cap \mathcal{X}_f$ are feasible, meaning that all constraints $\mathcal{C}$ (4.7) are fulfilled. However, only for a positively invariant subset $\mathcal{F}$ inside $\mathcal{C}_i \cap \mathcal{X}_f$, where a set $\mathcal{F}$ is called positively invariant for a system $\boldsymbol{x}(k+1) = \boldsymbol{g}(\boldsymbol{x}(k))$ if for all $\boldsymbol{x}(0) \in \mathcal{F}$ it holds that the corresponding solution to $\boldsymbol{x}(k+1) = \boldsymbol{g}(\boldsymbol{x}(k))$ satisfies $\boldsymbol{x}(k) \in \mathcal{F}$ for $k \in \mathbb{N}$, it can be guaranteed that the constraints $\mathcal{C}$ (4.7) are fulfilled for all times, in case the solution stays inside $\mathcal{C}_i$ and the target vehicle acceleration equals $a_t(k) = 0$ for $k \in \mathbb{N}$.

This is an important aspect when implementing an ACC. For example, consider a cut-in scenario in which a vehicle, driving with a lower velocity than the host vehicle, cuts in at a small distance in front of the host vehicle. This is a feasible state. To prevent violation of the constraint on the relative position, i.e. to prevent a collision, significant braking is required. As this might be prohibited by the constraint on the maximum deceleration, one of the constraints might be violated as a result.

Consequently, take $\mathcal{F}_\infty$, the largest positively invariant subset inside the intersection $\mathcal{C}_i \cap \mathcal{X}_f$ (Raković et al., 2004; Kolmanovsky and Gilbert, 1997). For states $\boldsymbol{x}_e(k) \in \mathcal{F}_\infty$, it is guaranteed that all constraints will be fulfilled for all times in case the target vehicle acceleration equals $a_t(k) = 0$ for $k \in \mathbb{N}$. This means that the ACC system can handle the present traffic scenario in an appropriate manner. However, for states $\boldsymbol{x}_e(k) \notin \mathcal{F}_\infty$, this cannot be guaranteed. At this point, the ACC system can warn the driver to take over control of the ACC system. Warning the driver in case of such a potentially dangerous situation, indicating to take over control, follows naturally from the theoretical MPC set up. The advantage of this setup is that it can be predicted if future constraint violation might occur, by detection of $\boldsymbol{x}_e(k) \notin \mathcal{F}_\infty$, and, accordingly, warn the driver in time.



(a) (b)

Figure 4.7: (a) Screen shot of a PreScan simulation environment. Three ACC SG-equipped host vehicles and three target vehicles causing corresponding cut-in situations are shown. The ACC SG systems of the host vehicles are tuned distinctively for comparison. (b) The Audi S8 in which the ACC SG is implemented. The functionality of the controller was first tested in the TNO VEHIL test facility (TNO Automotive, 2009) before the tests in actual traffic have been performed.

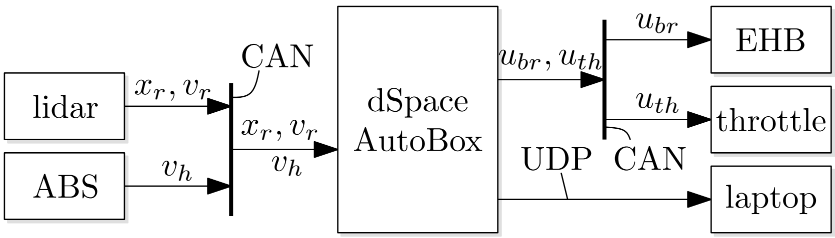


Figure 4.8: Schematic overview of the instrumentation of the vehicle. The main communication channels and corresponding signals are indicated, where u_{th} and u_{br} are the throttle and brake system control signals, respectively, v_h is the host vehicle velocity, and x_r and v_r are the relative position and velocity, respectively. For clarity, time dependency of the signals is omitted.

4.5.4 Simulations and on-the-road experiments

A set of seven distinct scenarios, encompassing the total envelope of working conditions, is determined to evaluate the functionality of the controller, see Table 4.1. Based on this set of scenarios, a test program is set up. Simulations are performed using the numerical tool PreScan, see Figure 4.7(a) for an impression (TNO Automotive, 2009). To validate the simulation results and to enable performance evaluation, the controller has been implemented on an Audi S8 (see Figure 4.7(b)).

A schematic overview of the instrumentation of the Audi S8 is shown in Figure 4.8. The velocity of the vehicle $v_h(t)$ is available on the CAN-bus via the built-in anti-lock braking system (ABS). The acceleration $a_h(t)$ is derived from this velocity signal. The vehicle

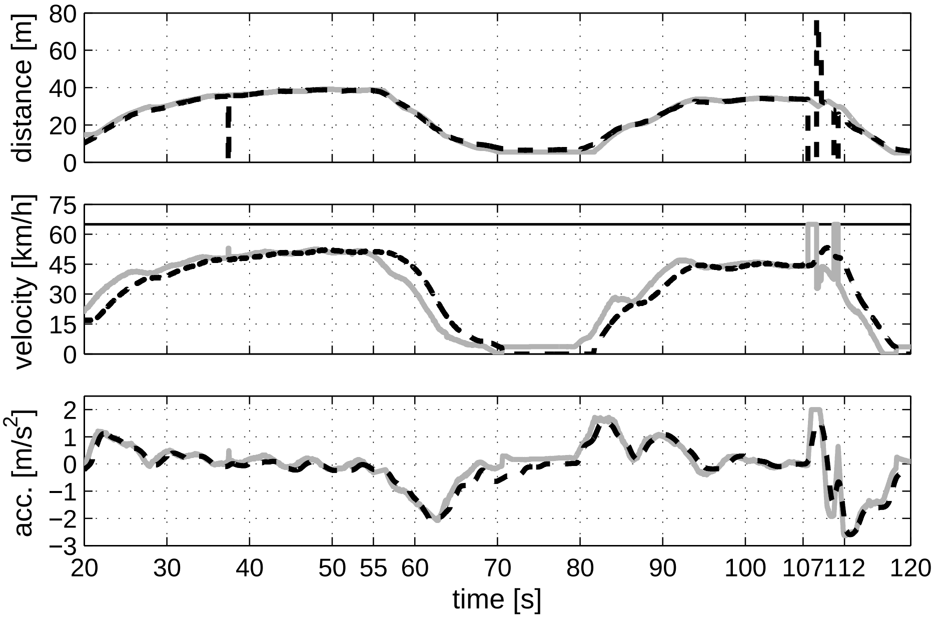


Figure 4.9: Experimental results corresponding to driving in city traffic. The dashed black lines represent x_r, v_h and a_h. The solid grey lines represent $x_{r,d}$, a combination of v_t and the desired CC velocity v_{CC}, and the controller output $a_{h,d}$. The thin solid black line in the middle plot indicates v_{CC}.

is equipped with an electro-hydraulic braking (EHB) system, facilitating brake-by-wire control. An OMRON laser radar, i.e. a lidar, with 150 m range is built-in. Using rapid control prototyping, the ACC system is implemented on a dSpace AutoBox, with a sample rate of 100 Hz. The ACC system includes a controller for the longitudinal dynamics of the vehicle, i.e. the inner control loop in Figure 4.2. A laptop is used to monitor all signals and log the data.

If not specified otherwise, a setting $P = 0.5$ is used for both the simulations and the experiments. In Figure 4.9, the results of on-the-road experiments with the Audi S8 are shown. The results correspond to driving in city traffic, showing, subsequently, steady following of a preceding target vehicle (scenario 1 of Table 4.1), with at about 37 s a momentarily loss of the fix of the radar on the target vehicle (the default radar output is $x_r = 0$ m), which has negligible influence on the driving behavior in this case; in between 55 and 100 s, the approach and following of a vehicle decelerating to standstill for a traffic light (scenario 5 of Table 4.1), which, subsequently, drives off (scenario 6 of Table 4.1); at 107 s, and several seconds later again, a cut out of the preceding vehicle, inducing a switch from ACC mode to CC mode (scenario 4 of Table 4.1); immediately following the cut-out situations, vehicles cut in, both times driving with a lower velocity than the host vehicle (scenario 3 of Table 4.1); and, finally, at 112 s, the results show again the approach and following of a vehicle decelerating to standstill (scenario 5 of Table 4.1).

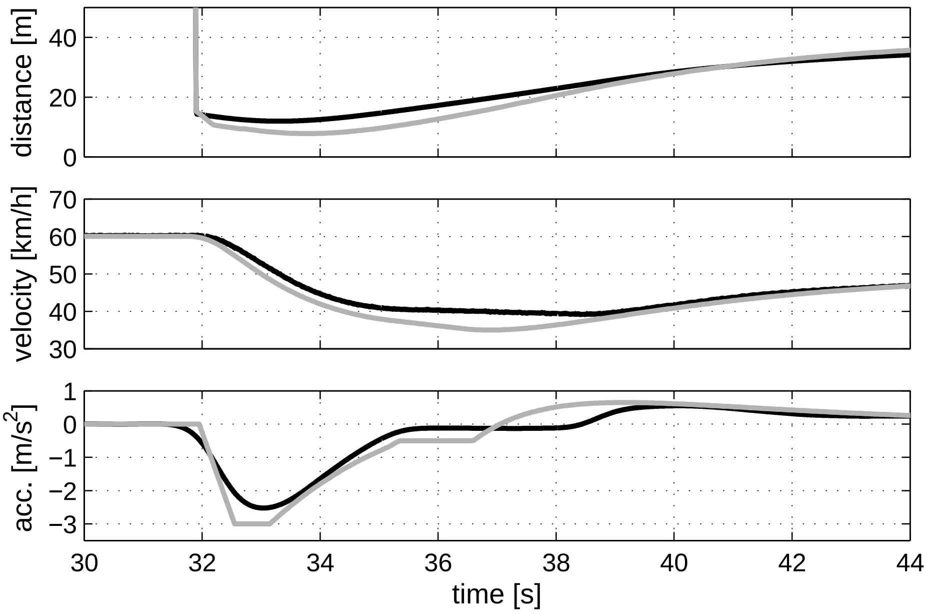


Figure 4.10: The distance x_r, host vehicle velocity v_h and acceleration a_h, corresponding to a *negative cut in* (scenario 3 of Table 4.1). The solid black and grey lines represent the results of on-the-road experiments and simulation results, respectively.

The sensitivity of the ACC for model uncertainties and measurement noise is not specifically investigated in this research. Nevertheless, the measurement results show that the ACC is, at least to some extent, in practice, robust for model uncertainties, and that the sensitivity for measurement noise is small.

Simulations vs experiments

In Figure 4.10, results of an on-the-road experiment and simulation results are compared. Values of the jerk $j_h(t)$ are not shown as these are difficult to obtain in practice. For the simulations, $a_h(k) = a_{h,d}(k)$ holds, as, for simplicity, vehicle models are not taken into account. Although good tracking properties are normally guaranteed by the vehicle-dependent control part, exact tracking is, of course, not the case in practice. This is the main cause of the differences between the simulation and experimental results. Taking into account appropriate vehicle models in the MPC synthesis as well as the simulations would increase the resemblance between the two responses significantly.

Nevertheless, the same characteristics can be seen in both the simulation and the experimental results. The time constants and peak values correspond fairly well. This means that the simulations can be used for the purpose of evaluation of the ACC characteristics. As the reproducibility of simulated traffic situations is better than that of real traffic situ-

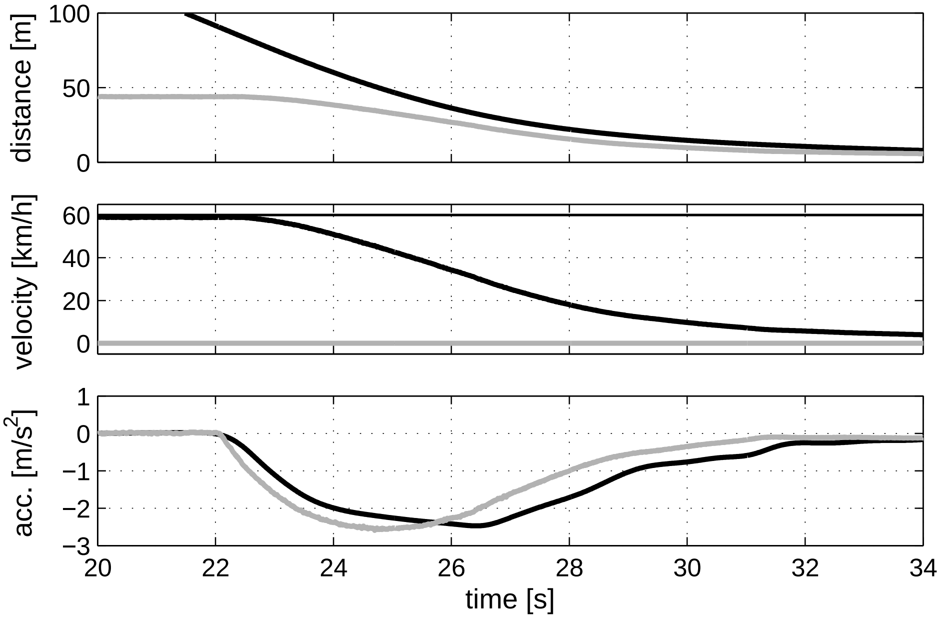


Figure 4.11: Experimental results corresponding to *the approach of a standstill vehicle* (scenario 2 of Table 4.1). The solid black lines represent x_r, v_h and a_h. The solid grey lines represent $x_{r,d}$, v_t and the controller output $a_{h,d}$. The thin black line in the middle plot represents v_{CC}.

ations, simulations are very useful for the comparison of the results of various settings. From this point of view, the resemblance between experimental and simulation results is satisfactory.

Additional functionality

The experimental results shown in Figure 4.11 show the working of the classic CC functionality and the switching between CC and ACC functionality, see Section 4.5.1 and 4.5.2, respectively. Experimental results corresponding to the approach of a standstill vehicle (scenario 2 of Table 4.1) are shown. Initially, the vehicle drives at the desired CC velocity of 60 km/h. With decreasing distance x_r, a desirable switch from CC to ACC functionality takes place at 22 s. The system switches from classic CC functionality to automatic following, i.e., ACC functionality, as the target vehicle is driving at a velocity which is lower than the desired CC velocity.

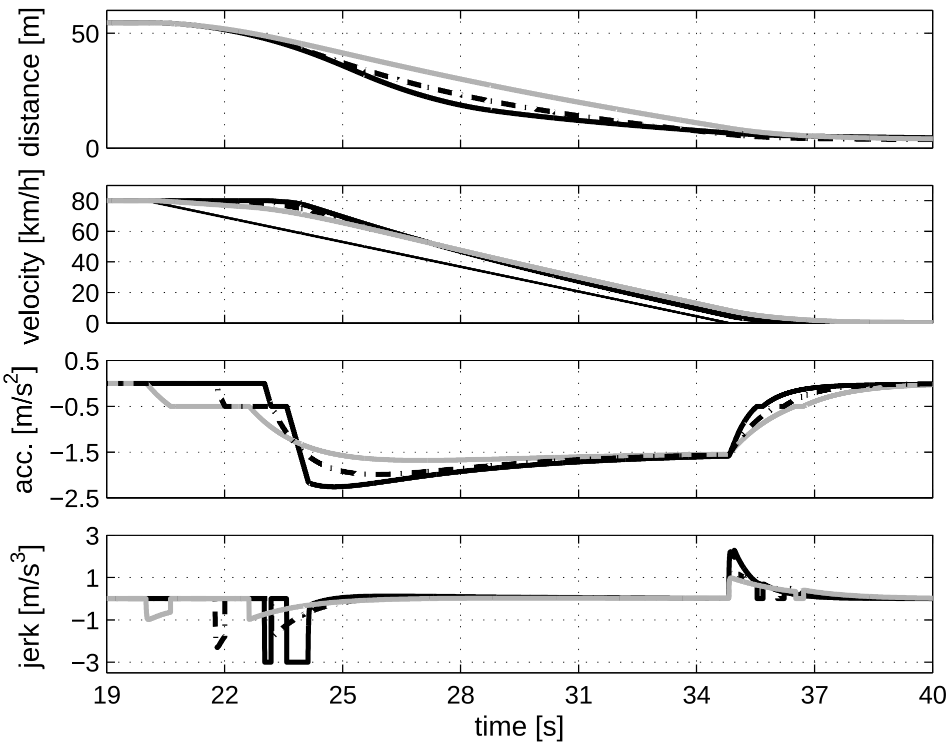


Figure 4.12: The distance x_r, the host vehicle velocity v_h, the acceleration a_h and the jerk j_h, corresponding to *following of a decelerating vehicle* (scenario 5 of Table 4.1). The solid black, dash-dotted and solid grey lines show the results for increasing $P \in \{0.2, 0.5, 0.8\}$.

Variable ACC behavior

To illustrate the influence of variable $P \in [0, 1]$, several scenarios of Table 4.1 are simulated for different settings $P \in \{0.2, 0.5, 0.8\}$. For the sake of reproducibility, simulation results instead of experimental results are shown. The results are presented in the Figures 4.12 to 4.15. For increasing $P \in \{0.2, 0.5, 0.8\}$, the results are indicated in solid black, dash-dotted black and solid grey, respectively.

Figure 4.12 and 4.13 show the results corresponding to following of a decelerating vehicle (scenario 5 of Table 4.1) and the approach of a standstill vehicle (scenario 2 of Table 4.1), respectively. The results in both Figures 4.12 and 4.13 clearly indicate more comfortable behavior for increasing P. The larger P, the smaller the resulting absolute acceleration and jerk peak values are.

In Figure 4.14, the results of a negative-cut-in scenario are shown (scenario 3 of Table 4.1). From a safety point of view, direct reaction and substantial braking are required in this case, disregarding the setting of P. At 20 s, a target vehicle shows up 20 m in front of the host vehicle with a velocity of 65 km/h, while the host vehicle is driving in

CC mode at 80 km/h. As a result, the host vehicle starts to brake immediately, indeed disregarding comfort-related measures such as the peak acceleration level. This indicates that safe behavior is achieved for any P. Furthermore, the results in Figure 4.14 show that for decreasing P the desired distance increases, which is desirable regarding safety.

Finally, the results for a so-called cut out (scenario 4 of Table 4.1) are shown in Figure 4.15. A preceding target vehicle changes lane, which yields an ACC to CC switch and subsequently acceleration to the desired CC velocity v_{CC}. As no other preceding traffic is present, the radar output becomes maximal, in this case 180 m. The constraint on the maximum acceleration is dependent on $P \in \{0.2, 0.5, 0.8\}$ as well as on the host vehicle velocity $v_h(k)$, which is clearly visible.

The results presented in Figures 4.12 to 4.15 show the proper working of the parameterization. By changing the setting of the design parameter $P \in [0, 1]$, the behavior of the ACC system changes, with respect to the comfort and the safety of the resulting driving action. The behavior of the system is thus translated into one essential parameter P, which is directly related to the desired characteristics of the driving behavior.

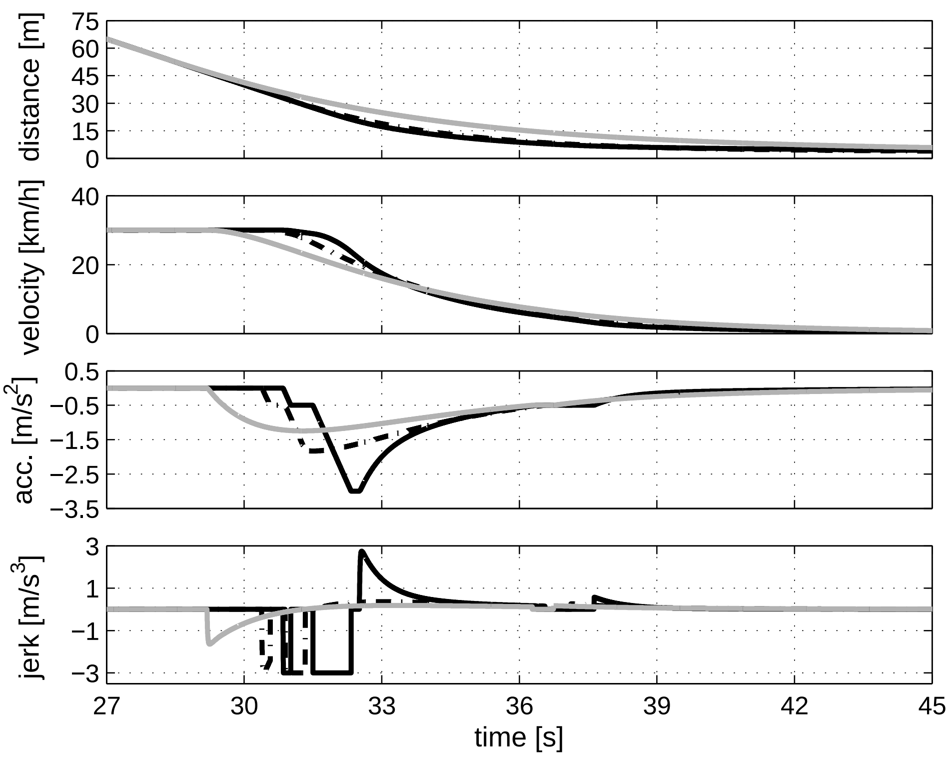


Figure 4.13: The distance x_r, the host vehicle velocity v_h, the acceleration a_h and the jerk j_h, corresponding to *approach of a standstill vehicle* (scenario 2 of Table 4.1). The solid black, dash-dotted and solid grey lines show the results for increasing $P \in \{0.2, 0.5, 0.8\}$.

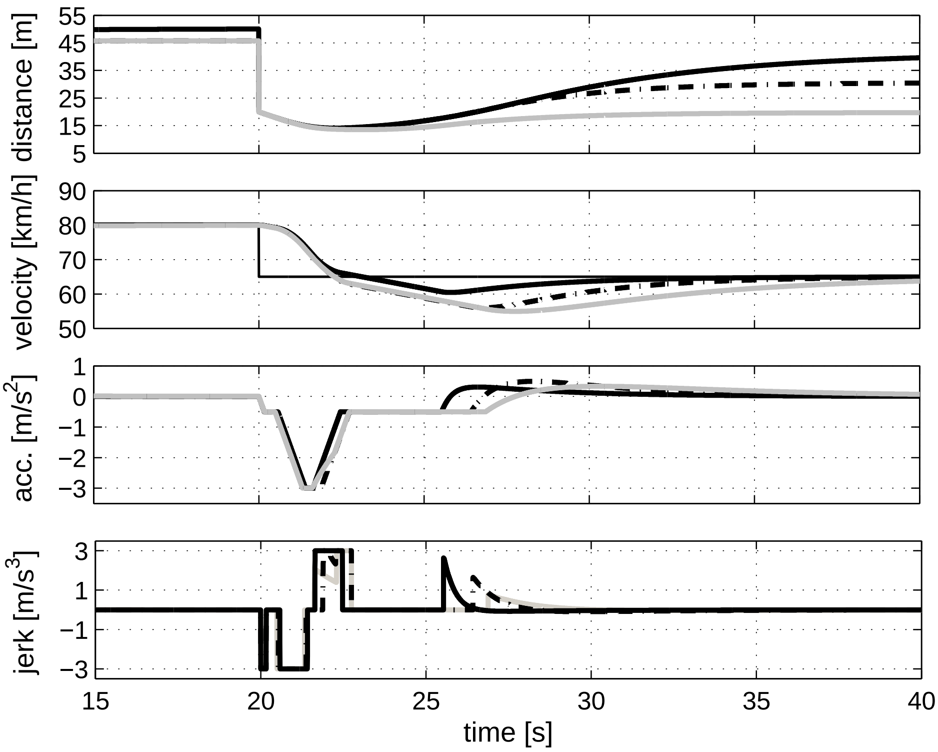


Figure 4.14: The distance x_r, the host vehicle velocity v_h, the acceleration a_h and the jerk j_h, corresponding to a *negative cut in* (scenario 3 of Table 4.1). The solid black, dash-dotted black and solid grey lines represent the results for increasing $P \in \{0.2, 0.5, 0.8\}$, respectively. The narrow black line in the middle figure represents the target vehicle velocity v_t.

4.6 Conclusions and recommendations

This chapter presents a systematic procedure to design an ACC, which is directly parameterized by the key characteristics of the ACC behavior. The goal of the parameterization of the ACC is to reduce the time it takes to tune the system and to enable the tuning for the driver. The latter requires that the tuning should be simple and intuitive with only a few design parameters, i.e. tuning knobs, that are directly related to the key characteristics of the ACC, such as safety, comfort, fuel economy and traffic flow efficiency. In this case, focus lies on safety and comfort, defining the design parameters P_s and P_c, respectively. To indicate the desired properties of the ACC system, quantification measures are defined corresponding to the key characteristics. The parameterized ACC is obtained by mapping the operating ranges of the quantification measures to the operating ranges of the design parameters P_s and P_c, being just $P_s \in [0, 1]$ and $P_c \in [0, 1]$.

The approach is based on (explicit) MPC. MPC can handle constraints and can easily in-

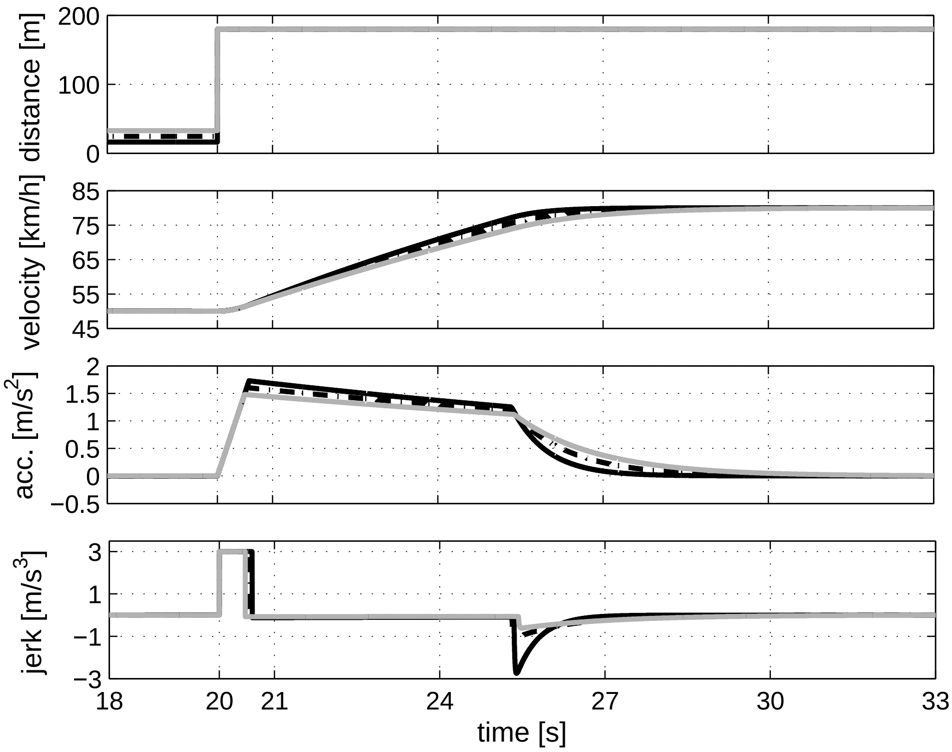


Figure 4.15: The distance x_r, the host vehicle velocity v_h, the acceleration a_h and the jerk j_h, corresponding to a *cut out, yielding an ACC to CC switch* (scenario 4 of Table 4.1). The solid black, dash-dotted black and solid grey lines represent the results for increasing $P \in \{0.2, 0.5, 0.8\}$, respectively.

clude tradeoffs between different key characteristics by appropriately selecting the cost function. In addition, MPC is suitable because its receding horizon implementation renders the ACC situation specific. This enables mimicking of human driving behavior, which is necessary for driver acceptance of the system. The many tuning parameters of the MPC setup are used to perform the mapping of the operating ranges of the quantification measures to the design parameters P_s and P_c. This requires engineering work. However, the tuning has to be done only once, fixing the MPC tuning parameters as a function of the essential design parameters P_s and P_c. Furthermore, a systematic procedure forms the basis for the mapping. In this specific case, the design parameters P_s and P_c could be united in one design parameter P. Consequently, after this parameterization, the ACC is easy and intuitive to tune by means of a single parameter P, which is directly related to the key characteristics safety and comfort.

Simulations as well as on-the-road experiments have shown the proper functioning of the parameterized ACC for a complete envelope of working conditions: i) the simulation results resemble the experimental results satisfactorily, ii) additional functionality includes

CC functionality, provides switching between CC and ACC functionality, and facilitates in time warning of the driver in case of a potentially dangerous situation, and iii) changing the behavior of the system by changing the setting of the design parameter P, has proven to work in a desired manner. Nevertheless, stability of the switching between CC and ACC functionality is still to be proven.

Due to the generality of the approach, other characteristics can be incorporated in the design using the same systematic design procedure. This is, for example, demonstrated in Naus et al. (2010e), considering, besides safety and comfort, also fuel economy as a key characteristic. In the next chapter, the two-vehicle model is extended to multiple vehicles, taking into account vehicle-to-vehicle communication. This allows for the design of so-called cooperative ACC systems. The wireless communication provides additional information concerning the surrounding traffic. As a result, it could be possible, for example, to consider traffic throughput as a key characteristic.

CHAPTER 5

String-stable CACC design and experimental validation, a frequency-domain approach[1]

Abstract - The design of a cooperative adaptive cruise control (CACC) system and its practical validation are presented. Focusing on the feasibility of implementation, a decentralized controller design with a limited communication structure is proposed, in this case a wireless communication link with the nearest preceding vehicle only. Accordingly, a necessary and sufficient frequency-domain condition for string-stability is derived, taking into account heterogeneous traffic, i.e., vehicles with possibly different characteristics. For a velocity-dependent inter-vehicle spacing policy, it is shown that the wireless communication link enables driving at small inter-vehicle distances, while string stability is guaranteed. For a constant, velocity-independent inter-vehicle spacing, string stability cannot be guaranteed. To validate the theoretical results, experiments are performed with two CACC-equipped vehicles. Implementation of the CACC system, the string-stability characteristics of the practical setup, and experimental results are discussed, indicating the advantages of the design over standard adaptive cruise control (ACC) functionality.

5.1 Introduction

Cooperative adaptive cruise control (CACC) is an extension of adaptive cruise control (ACC) functionality. Nowadays, ACC functionality is widespread and available in numerous commercially available vehicles. ACC automatically adapts the cruise control velocity

[1]This chapter is based on G. J. L. Naus, R. P. A. Vugts, J. Ploeg, M. J. G. van de Molengraft and M. Steinbuch (2010). String-stable CACC design and experimental validation, a frequency-domain approach. *IEEE Trans. Veh. Technol.* (accepted for publication).

of a vehicle if there is preceding traffic driving too close and at a lower velocity. Commonly, a radar is adopted to detect preceding traffic (see Chapter 4). As ACC is primarily intended as a comfort system, and, to a smaller degree, as a safety system, relatively large inter-vehicle distances are adopted in commercially available systems (International Organization for Standardization, 2002; Ioannou and Chien, 1993; Vahidi and Eskandarian, 2003). Decreasing the inter-vehicle distance to a small value of only a few meters is expected to yield an increase in traffic throughput. Moreover, specifically focusing on heavy duty vehicles, a significant reduction of the aerodynamic drag force is possible, thus decreasing fuel consumption (Arem et al., 2006; Shladover, 2005). Consequently, it is desirable to enable this for larger strings of vehicles, so-called platoons.

When commercially available ACC functionality is employed to achieve such small inter-vehicle distances, string-unstable driving behavior may result. The string stability of a platoon indicates whether oscillations are amplified upstream the traffic flow (Peppard, 1974). The longitudinal dynamics of the platoon are called string stable if sudden changes in the velocity of a vehicle at the front of a platoon are attenuated by the vehicles upstream the platoon. If changes in the velocity of a vehicle at the front of a platoon are amplified by the vehicles upstream the platoon, the longitudinal dynamics of the platoon are called string unstable. An example of string-unstable behavior is the forming of traffic jams that occur for no apparent reason. No accident or bottleneck needs to be present, just too much traffic or erratic driving behavior may cause a shockwave of continuously increased braking upstream a string of vehicles, until vehicles come to a halt and a traffic jam results (Sugiyama et al., 2008). Consequently, considering traffic throughput and fuel economy, as well as comfort and safety, string-unstable driving behavior is highly undesirable.

Extending standard ACC functionality with a wireless inter-vehicle communication link enables driving at small inter-vehicle distances while maintaining string stability (Rajamani and Zhu, 2002). The resulting functionality is called cooperative adaptive cruise control (CACC). The design of CACC functionality has been discussed extensively in literature, see, e.g., (Ioannou and Chien, 1993; Lu et al., 2004; Rajamani and Zhu, 2002; Sheikholeslam and Desoer, 1992; Shladover, 2005; Swaroop et al., 2001). However, a generic approach for the design of a CACC system does not (yet) exist. In most cases, a specific system setup and corresponding working conditions are considered, rather than true generalizations. Furthermore, focus is often on theoretical analysis of the system, rather than the possibilities for practical implementation. In this research, a CACC design specifically focusing on the feasibility of implementation is proposed.

The contribution of this research involves, first, the design of a CACC system focusing on the feasibility of implementation and the definition of a corresponding sufficient, frequency-domain condition for string stability of heterogeneous traffic. Second, a theoretical analysis and, in particular, experimental validation of the proposed CACC system are presented.

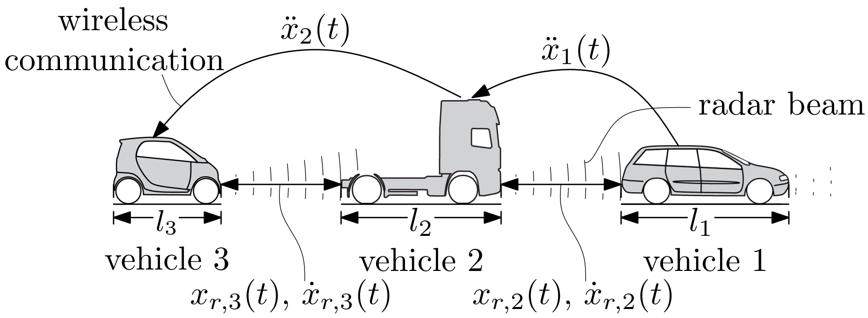


Figure 5.1: Schematic representation of a platoon of vehicles equipped with CACC functionality, where $x_{r,i}(t)$, $\dot{x}_{r,i}(t)$, $\ddot{x}_i(t)$, and l_i represent the relative position, the relative velocity, the acceleration, and the length of vehicle i, respectively.

In Sections 5.2 and 5.3, the problem formulation and the design of the CACC system are presented, respectively. The definition of string stability and both the benefit of the available wireless information and the choice for the inter-vehicle spacing policy are discussed in Sections 5.4 and 5.5, respectively. Finally, experimental validation of the proposed framework is presented using two Citroën C4's (Section 5.6). The chapter is closed with conclusions and recommendations.

5.2 Problem formulation

5.2.1 CACC system setup

In Fig. 5.1, a schematic representation of a platoon of vehicles equipped with CACC functionality is shown, where $x_{r,i}(t)$, $\dot{x}_{r,i}(t)$, $\ddot{x}_i(t)$, and l_i are the relative position, the relative velocity, the acceleration, and the length of vehicle i, respectively.

Focusing on the feasibility of implementation, a decentralized controller design is pursued, as opposed to a centralized design (Levine and Athans, 1966). Communication with the nearest preceding vehicle is adopted, as is schematically depicted in Fig. 5.1. Examples of other communication structures are a centralized controller design and communication between all vehicles in a platoon (Levine and Athans, 1966), bi-directional communication with the nearest vehicles (Peppard, 1974; Yanakiev et al., 1998), or communication with both the nearest vehicles and a designated platoon leader (Sheikholeslam and Desoer, 1993; Rajamani and Zhu, 2002). Communication with the directly preceding vehicle only is often called semi-autonomous ACC and facilitates easy implementation.

Similar to the communication structure, the variety in spacing-policies that are proposed in literature, i.e., the desired distance between the vehicles in a platoon, is large (Santhanakrishnan and Rajamani, 2003; Swaroop et al., 1994). For example, a constant spacing policy (Rajamani and Zhu, 2002), a velocity-dependent spacing policy (Barooah and Hespanha, 2005), or more complex, nonlinear spacing policies (Yanakiev et al., 1998) are considered. Focusing on the feasibility of implementation rather than the definition of a new spacing policy, the most common policy is adopted, including a constant part and, optionally, a velocity-dependent part (Rajamani and Zhu, 2002; Barooah and Hespanha, 2005).

The CACC design is based on a standard ACC system. Moreover, the additional data that is available through the wireless communication link is used in a feedforward setting. Hence, ACC functionality is still available if no communication is present (Rajamani and Zhu, 2002; Swaroop and Hedrick, 1999). Furthermore, heterogeneous traffic is considered, i.e., a platoon of vehicles with possibly different characteristics, as is the case in actual traffic. In literature, however, homogeneous traffic is often considered instead, i.e., vehicles with identical characteristics (e.g., Shaw and Hedrick, 2007a; Khatir and Davison, 2004). Finally, delay in the communication signal is taken into account. The effect of delay in the communicated data is often neglected (see, e.g, Rajamani and Zhu, 2002; Barooah and Hespanha, 2005), with some notable exceptions being Lu et al. (2004); Liu and Mahal (2001); Sheikholeslam and Desoer (1992).

5.2.2 String stability

Considering the definition of the string stability of a platoon of vehicles, much ambiguity is present in literature. The common part in the definitions is that they all consider amplification of oscillations upstream a string of vehicles, i.e., from vehicle $i = 1$ to vehicle $i > 1$ (see Fig. 5.1) (Peppard, 1974). However, oscillations in different signals are considered.

Focusing on preventing collisions, the errors $e_i(t)$ between the desired and the actual inter-vehicle distances are often considered. Correspondingly, the driving behavior of a platoon is denoted string stable if these errors do not amplify upstream a platoon (Swaroop et al., 1994; Seiler et al., 2004; Shladover et al., 1991; Sheikholeslam and Desoer, 1993; Warnick and Rodriguez, 1994; Yanakiev and Kanellakopoulos, 1998; Rajamani et al., 2000). A modification on this definition is presented by (Liang and Peng, 1999), considering amplification of oscillations in the inter-vehicle distances $x_{r,i}(t)$ instead.

For the case of heterogeneous traffic, amplification of oscillations in the absolute vehicle positions $x_i(t)$ or in the vehicle velocities $\dot{x}_i(t)$ is considered by (Huppe et al., 2003; Khatir and Davison, 2004; Shaw and Hedrick, 2007b; Peppard, 1974). In these papers, focus is on oscillations as discussed in the introduction, which, for example result in traffic jams. Finally, in some cases the error signals are considered for analysis, while the absolute vehicle positions and velocities are considered when evaluating the results (Santhanakrishnan and Rajamani, 2003; Rajamani and Zhu, 2002; Ioannou and Chien, 1993; Swaroop et al., 2001).

Furthermore, both time-domain and frequency-domain conditions are presented, e.g., (Lu et al., 2004; Shaw and Hedrick, 2007b). In this paper, a frequency-domain approach is adopted. The definition of string-stability is revised, focusing on the feasibility of implementation, i.e., for a decentralized controller design, communication with the directly preceding vehicle only and heterogeneous traffic. This yields a necessary and sufficient definition for string stability. Adopting a linear, frequency-domain approach enables easy and intuitive analysis, focusing on amplification of oscillations. The resulting condition for string stability is used to compare the characteristics of standard ACC functionality and the proposed CACC system, indicating the benefits of the wireless communication link.

5.3 Control structure

The CACC design is based on a standard ACC system and the spacing policy that is most commonly used. In this section, the ACC design, the spacing policy and the CACC structure are discussed.

5.3.1 ACC control structure

Consider a string of heterogeneous vehicles, as depicted in Figure 5.1, where $x_{r,i}(t)$, $\dot{x}_{r,i}(t)$, $\ddot{x}_i(t)$, and l_i are the relative position, the relative velocity, the acceleration, and the length of vehicle i, respectively. The primary control objective for each vehicle is to follow the corresponding preceding vehicle at a desired distance, i.e., a desired relative position $x_{r,d,i}(t)$. Using a radar, the relative position $x_{r,i}(t)$ and the relative velocity $\dot{x}_{r,i}(t)$ are measured. The relative position is defined as

$$x_{r,i}(t) = x_{i-1}(t) - x_i(t) \tag{5.1}$$

where the vehicle length l_i is not taken into account (see Figure 5.1).

In a standard ACC system, the radar output data are used in a feedback setting. A feedback controller controls the spacing error between the desired distance and the actual distance, which is defined as

$$e_i(t) = x_{r,i}(t) - x_{r,d,i}(t) \tag{5.2}$$

Defining $e_i(t)$ in this manner implies that positive control action, i.e., acceleration, is required when the inter-vehicle distance $x_{r,i}(t)$ is too large with respect to the desired distance $x_{r,d,i}(t)$. This makes the control action $u_i(t)$ intuitive. The resulting control setup is depicted schematically in Figure 5.2. For the purpose of analysis, the radar output data

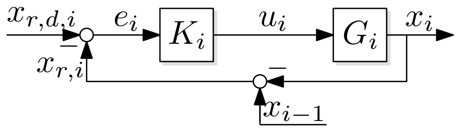


Figure 5.2: ACC control structure, where $G_i = G_i(s)$ represent the dynamics of the i^{th} vehicle, and $K_i = K_i(s)$ is the corresponding ACC feedback controller, for $i \geq 1$. For clarity, the time dependency of the signals is omitted.

are reconstructed, based on the positions of the vehicles i and $i-1$. The models $G_i(s)$ and $K_i(s)$ represent linear transfer function models, where s the Laplace operator. For clarity, this dependency is omitted in the figures.

It is assumed that the model $G_i(s)$ includes a (low level) control loop for the longitudinal vehicle dynamics. Consequently, the input $u_i(t)$ of $G_i(s)$ can be regarded as a desired acceleration $\ddot{x}_{d,i}(t)$. The low level control loop ensures tracking of this desired acceleration via actuation of the throttle and the brake system. It is assumed that the low level closed-loop longitudinal vehicle dynamics of vehicle i can be represented by

$$G_i(s) = \frac{k_{G,i}}{s^2(\tau_i s + 1)} e^{-\phi_i s}, \qquad \text{for } i \geq 1 \tag{5.3}$$

where $\tau_i^{-1} = \omega_{bw,l,i}$ is the low level closed-loop bandwidth, $k_{G,i}$ is the model gain, and ϕ_i represents the actuator and internal communication delay time.

Consider the closed-loop complementary sensitivity transfer function of the ACC control structure

$$T_i'(s) = \frac{G_i(s)K_i(s)}{1 + G_i(s)K_i(s)}, \qquad \text{for } i \geq 1 \tag{5.4}$$

Define the bandwidth $\omega_{bw,i}'$ of $T_i'(s)$ as the frequency at which the magnitude of the open-loop frequency response function $L_i'(j\omega) = G_i(j\omega)K_i(j\omega)$ crosses 0 dB in the downwards sense. Given the vehicle dynamics $G_i(s)$ (5.3) with reasonably small ϕ_i, a standard PD controller provides the freedom to choose a desired bandwidth $\omega_{bw,i}' = \omega_{bw,d,i}'$, with a desired phase margin $\alpha_{PM,i}' = \alpha_{PM,d,i}'$. Correspondingly, the ACC feedback controller $K_i(s)$ that is considered in this research, is given by

$$K_i(s) = k_{P,i} + k_{D,i}s, \qquad \text{for } i \geq 1 \tag{5.5}$$

The values for $k_{P,i}$ and $k_{D,i}$ can be computed straightforwardly, using $|L_i'(j\omega_{bw,d,i}')| = 1$ and $\angle L_i'(j\omega_{bw,d,i}') = \alpha_{PM,d,i}'$. In this research, we take, for simplicity, $k_{P,i} = k_{D,i}^2 = \omega_{K,i}^2$, which yields

$$K_i(s) = \omega_{K,i}\left(\omega_{K,i} + s\right), \qquad \text{for } i \geq 1 \tag{5.6}$$

The desired bandwidth $\omega'_{bw,i}(\omega_{K,i})$, which can be regarded as a performance measure for the tracking behavior of the CACC system, i.e., the tracking of $x_{r,d,i}(t)$ (5.8), can still be chosen freely. The phase margin, on the other hand, now is a result of the choice for the desired bandwidth, i.e., $\alpha'_{PM,i} = \angle L'_i(j\omega'_{bw,d,i})$.

5.3.2 Spacing policy

The desired relative position or distance $x_{r,d,i}(t)$, is determined by the so-called spacing policy. The most common policy includes a constant part, e.g., (Rajamani and Zhu, 2002), and, optionally, a velocity-dependent part, e.g., (Barooah and Hespanha, 2005), which is given by

$$x_{r,d,i}(t) = x_{r,0,i} + h_{d,i}\dot{x}_i(t), \qquad \text{for } i \geq 1 \tag{5.7}$$

where $x_{r,0,i}$ is the constant part, or the desired distance at standstill, $h_{d,i}$ is the so-called desired headway time, and $\dot{x}_i(t)$ is the velocity of vehicle i, for $i \geq 1$. Hence, the headway time $h_{d,i}$ represents the time it takes for vehicle i to bridge the distance in between the vehicles i and $i-1$ when continuing to drive with a constant velocity.

The desired distance at standstill, $x_{r,0,i}$, can be regarded as an extension of the vehicle length l_i, i.e., $l'_i = l_i + x_{r,0,i}$. This is depicted schematically in Figure 5.3. Accordingly, the desired distance at standstill is not considered in the rest of this research, yielding

$$x_{r,d,i}(t) = h_{d,i}\dot{x}_i(t), \qquad \text{for } i \geq 1 \tag{5.8}$$

Correspondingly, the relative position of the vehicles is redefined, with slight abuse of notation, via

$$x_{r,i}(t) \triangleq x_{r,i}(t) - x_{r,0,i} \tag{5.9}$$

For $h_{d,i} = 0.0$ s, a constant spacing policy results. For $h_{d,i} > 0.0$ s, the spacing policy (5.8) is called a constant headway time policy, targeting a constant inter-vehicle time gap

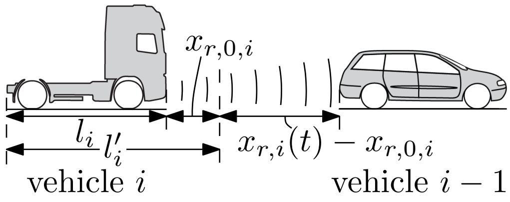


Figure 5.3: Schematic representation of two vehicles driving in a platoon, where l_i is the actual vehicle length of vehicle i, $x_{r,0,i}$ is the desired distance at standstill, $x_{r,i}(t)$ is the relative position, and $l'_i = l_i + x_{r,0,i}$ is the vehicle length considered for analysis.

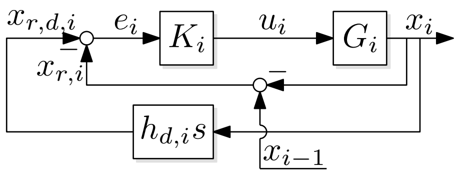


Figure 5.4: ACC control structure, where $G_i = G_i(s)$ represent the dynamics of the i^{th} vehicle, $K_i = K_i(s)$ is the corresponding ACC feedback controller, s is the Laplace operator, and $h_{d,i}$ is the headway time. For clarity, the time dependency of the signals is omitted.

of $h_{d,i}$ seconds. In the system analysis further on, both the constant spacing policy and the constant headway time policy are considered.

As a result of the velocity-dependent spacing policy, the control structure becomes cascaded. This is depicted schematically in Figure 5.4. The design of the outer loop of the cascaded controller comes down to the choice for the headway time $h_{d,i}$. The corresponding closed-loop transfer $T_i(s)$ equals

$$T_i(s) = \frac{G_i(s)K_i(s)}{1 + H_i(s)G_i(s)K_i(s)}, \qquad \text{for } i \geq 1 \tag{5.10}$$

where

$$H_i(s) = 1 + h_{d,i}s, \qquad \text{for } i \geq 1 \tag{5.11}$$

includes the spacing policy dynamics (5.8). Consider the corresponding open-loop transfer function $L_i(s) = H_i(s)G_i(s)K_i(s) = H_i(s)L_i'(s)$ and assume that $K_i(s)$ is designed such that the inner control loop $T_i'(s)$ (5.4) is stable. Comparing $L_i(s) = H_i(s)L_i'(s)$ to $L_i'(s)$, it follows directly that $H_i(s)$ (5.11) has positive influence on the phase margin for $h_{d,i} \geq 0.0$ s. The larger the headway time $h_{d,i}$, the larger the phase lead of the open-loop $L_i(j\omega) = H_i(j\omega)L_i'(j\omega)$ with respect to $L_i'(j\omega)$. Consequently, stability of the closed-loop system $T_i(s)$ is positively influenced for $h_{d,i} \geq 0.0$ s. Furthermore, assuming that the bandwidth $\omega_{bw,i}$ of $T_i(s)$ is of the same order or smaller than $\omega_{bw,i}'$, the performance of the system is not affected significantly.

5.3.3 CACC control structure

As discussed in Section 5.2, wireless communication with the nearest preceding vehicle is considered for the CACC system. Via this communication channel, the acceleration of the preceding vehicle $\ddot{x}_{i-1}(t)$ is available, see Figure 5.1. The wirelessly communicated data are used in a feedforward setting, extending the standard ACC feedback controller to CACC functionality. As a result, the system can degrade to a standard ACC system when no communication is present or if communication fails.

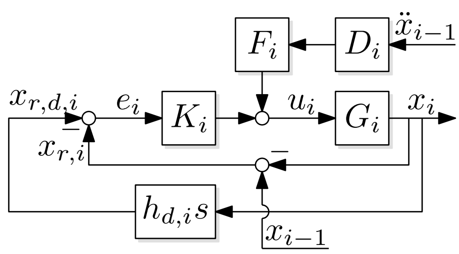


Figure 5.5: CACC control structure, where $G_i = G_i(s)$ represent the dynamics of the i^{th} vehicle, $K_i = K_i(s)$ is the corresponding ACC feedback controller, $F_i = F_i(s)$ is the feedforward filter, $D_i = D_i(s)$ is the communication delay, s is the Laplace operator, and $h_{d,i}$ the headway time. For clarity the time dependency of the signals is omitted.

The acceleration of the preceding vehicle is used as a feedforward control signal via a feedforward filter $F_i(s)$. The acceleration is obtained through wireless communication, which includes a communication delay $D_i(s)$. The resulting control structure is depicted schematically in Figure 5.5.

The delay is represented by a constant delay time θ_i, yielding

$$\mathcal{L}\left(\ddot{x}_{i-1}(t-\theta_i)\right) = D_i(s)s^2 X_{i-1}(s) \tag{5.12}$$

where

$$D_i(s) = e^{-\theta_i s}, \qquad \text{for } i > 1 \tag{5.13}$$

and $\mathcal{L}(\cdot)$ denotes the Laplace transformation. The design of the feedforward filter is based on a zero-error condition, where the Laplace transform of the error (5.2) is defined as

$$\begin{aligned} \mathcal{L}(e_i(t)) &= \mathcal{L}(x_{r,i}(t)) - \mathcal{L}(x_{r,d,i}(t)), \qquad \text{for } i > 1 \\ &= \mathcal{L}(x_{i-1}(t)) - H_i(s)\mathcal{L}(x_i(t)) \\ &= \frac{1 - H_i(s)G_i(s)F_i(s)D_i(s)s^2}{1 + H_i(s)G_i(s)K_i(s)}\mathcal{L}(x_{i-1}(t)) \end{aligned} \tag{5.14}$$

Considering $\mathcal{L}(e_i(t))$ (5.14), the communication delay $D_i(s)$ can only be compensated for using an estimator for the communicated acceleration signal. In this research, it is assumed that such an estimator is not available. However, in the subsequent system analysis, both a system with and a system without communication delay are considered. The latter is representative of the case that an appropriately designed estimator is available. Hence, demanding $\mathcal{L}(e_i(t)) = 0$ and taking into account a communication delay $D_i(s)$ (5.13) that is not compensated for by the feedforward filter, yields

$$F_i(s) = (H_i(s)G_i(s)s^2)^{-1}, \qquad \text{for } i > 1 \tag{5.15}$$

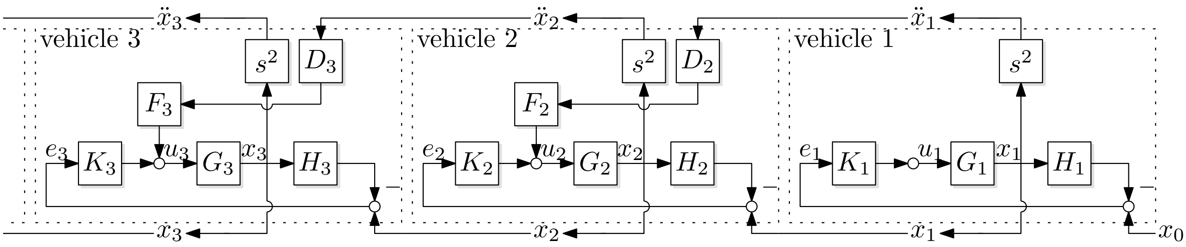


Figure 5.6: Control structure of a platoon of vehicles, where $G_i = G_i(s)$ represent the dynamics of the i^{th} vehicle, $K_i = K_i(s)$ is the corresponding ACC feedback controller, $F_i = F_i(s)$ is the feedforward controller, $D_i = D_i(s)$ is the communication delay and $H_i = H_i(s)$ represent the spacing policy dynamics, for $i \in \{1, 2, 3\}$. For clarity, the time dependency of the signals is omitted.

5.4 String stability, a frequency-domain approach

In this section, the definition of string stability is revised, following a frequency-domain approach (see Section 5.2). Focus is on the feasibility of application in practice. Hence, heterogeneous traffic is considered, a decentralized controller design is pursued, and communication with the directly preceding vehicle only is considered. It is shown that in that case, the frequency-domain system state $X_i(s)$ has to be considered to define string-stability of a platoon of vehicles, which is directly related to the absolute vehicle positions $x_i(t)$ or the vehicle velocities $\dot{x}_i(t)$.

Consider the CACC control structure shown in Figure 5.5. Coupling several of these control structures yields the control structure for a platoon of vehicles. This is shown in Figure 5.6, where the inner and the outer control loop (see Figure 5.5) are merged using the definition of $H_i(s)$ (5.11). The first vehicle in the platoon, $i = 1$, is assumed to follow a given time-varying reference position $x_0(t)$ using radar measurements. The other vehicles, $i > 1$, use both the radar and the wireless communication.

The transfer functions from the input $\mathcal{L}(x_0(t)) = X_0(s)$ to the Laplace transforms of $u_i(t)$, $x_i(t)$, and $e_i(t)$, i.e., $\mathcal{L}(u_i(t)) = U_i(s)$, $\mathcal{L}(x_i(t)) = X_i(s)$, and $\mathcal{L}(e_i(t)) = E_i(s)$, are given by

$$\frac{U_i(s)}{X_0(s)} = \begin{cases} S_1(s)K_1(s), & \text{for } i = 1 \\ \dfrac{U_1(s)}{X_0(s)} \displaystyle\prod_{k=2}^{i} S_k(s)\left(F_k(s)D_k(s)s^2 + K_k(s)\right)G_{k-1}(s), & \text{for } i > 1 \end{cases} \tag{5.16a}$$

$$\frac{X_i(s)}{X_0(s)} = G_i(s)\frac{U_i(s)}{X_0(s)}, \qquad \text{for } i > 1 \tag{5.16b}$$

$$\frac{E_i(s)}{X_0(s)} = \begin{cases} S_1(s), & \text{for } i = 1 \\ \dfrac{X_{i-1}(s)}{X_0(s)} S_i(s)(1 - H_i(s)G_i(s)F_i(s)D_i(s)s^2), & \text{for } i > 1 \end{cases} \tag{5.16c}$$

where

$$S_i(s) = (1 + H_i(s)G_i(s)K_i(s))^{-1} \tag{5.17}$$

is the closed-loop sensitivity transfer function of vehicle i. These relations follow directly from the control structure in Figure 5.6. Next, define the so-called string-stability transfer functions $\mathcal{G}'_{\Lambda,i'}(s)$, $\Lambda \in \{U, X, E\}$

$$\mathcal{G}'_{\Lambda,i'}(s) = \frac{\Lambda_{i'}(s)}{\Lambda_1(s)} = \frac{\Lambda_{i'}(s)}{X_0(s)} \left(\frac{\Lambda_1(s)}{X_0(s)}\right)^{-1}, \qquad \text{for } i' > 1 \tag{5.18}$$

where i' denotes the last vehicle in a platoon of vehicles. The magnitude of the string-stability transfer functions $\mathcal{G}'_{\Lambda,i'}(s)$ is a measure for the amplification of oscillations upstream a platoon. Hence, as string-stability is defined as damping of the magnitude of oscillations upstream a platoon, a necessary condition for string stability is (see, e.g., Sheikholeslam and Desoer, 1993; Liang and Peng, 2000)

$$\left\|\mathcal{G}'_{\Lambda,i'}(j\omega)\right\|_\infty \leq 1, \qquad \text{for } i' > 1 \tag{5.19}$$

in which $\|\cdot\|_\infty$ denotes the maximum amplitude for all ω. In the case that

$$\left\|\mathcal{G}'_{\Lambda,i'}(j\omega)\right\|_\infty = 1, \qquad \text{for } i' > 1,\ \omega > 0 \tag{5.20}$$

holds, this is called marginal string stability.

Targeting a decentralized controller design, the dynamics of all vehicles $k \in \{1, \ldots, i'-1\}$ have to be known to fulfill condition (5.19). For heterogeneous traffic, i.e., vehicles with possibly different characteristics and dynamics, this requires an extensive communication structure. As communication with the nearest preceding vehicle only is considered, more conservative string-stability transfer functions $\mathcal{G}_{\Lambda,i}(s)$, $\Lambda \in \{U, X, E\}$ are defined,

$$\mathcal{G}_{\Lambda,i}(s) = \frac{\Lambda_i(s)}{\Lambda_{i-1}(s)} = \frac{\Lambda_i(s)}{X_0(s)} \left(\frac{\Lambda_{i-1}(s)}{X_0(s)}\right)^{-1}, \qquad \text{for } i > 1 \tag{5.21}$$

yielding a more conservative, sufficient condition for string stability:

$$\|\mathcal{G}_{\Lambda,i}(j\omega)\|_\infty \leq 1, \qquad \text{for } i > 1 \tag{5.22}$$

Analogously, a sufficient condition for marginal string stability is

$$\|\mathcal{G}_{\Lambda,i}(j\omega)\|_\infty = 1, \qquad \text{for } i > 1,\ \omega > 0 \tag{5.23}$$

As it holds that

$$\mathcal{G}'_{\Lambda,i}(s) = \prod_{k=2}^{i} SS_{\Lambda,k}(s), \qquad \text{for } i > 1 \tag{5.24}$$

condition (5.19) is automatically satisfied if (5.22) is satisfied. In literature, condition (5.19) is referred to as weak string stability, whereas (5.22) is called strong string stability (Swaroop and Hedrick, 1999). The weak string-stability condition (5.19) considers the platoon as a whole: if compensated for somewhere else, local string-unstable behavior can be allowed in the platoon. The strong string-stability condition (5.22), on the other hand, imposes string-stable behavior at every position in the platoon. As the latter condition imposes more stringent conditions on the CACC design, this condition is referred to as strong string stability.

Focusing on the feasibility of implementation, i.e., for a platoon of heterogeneous vehicles with a limited communication structure and a decentralized control architecture, condition (5.22) is a useful condition to use in the controller design. Hence, in the rest of this research, the sufficient condition (5.22) is considered as a necessary condition for string stability.

The string-stability transfer functions $\mathcal{G}_{U,i}(s)$, $\mathcal{G}_{X,i}(s)$, and $\mathcal{G}_{E,i}(s)$ (5.21) follow directly from (5.16a,b,c), yielding

$$\mathcal{G}_{U,i}(s) = \frac{U_i(s)}{U_{i-1}(s)} = S_i(s)(F_i(s)D_i(s)s^2 + K_i(s))G_{i-1}(s), \qquad \text{for } i > 1 \tag{5.25a}$$

$$\mathcal{G}_{X,i}(s) = \frac{X_i(s)}{X_{i-1}(s)} = S_i(s)(F_i(s)D_i(s)s^2 + K_i(s))G_i(s), \qquad \text{for } i > 1 \tag{5.25b}$$

$$\mathcal{G}_{E,i}(s) = \frac{E_i(s)}{E_{i-1}(s)} = \begin{cases} S_2(s)(1-\Xi_2(s))G_1(s)K_1(s), & \text{for } i = 2 \\ \dfrac{S_i(s)(1-\Xi_i(s))}{S_{i-1}(s)(1-\Xi_{i-1}(s))}\dfrac{X_{i-1}(s)}{X_{i-2}(s)}, & \text{for } i > 2 \end{cases} \tag{5.25c}$$

with

$$\Xi_i(s) = H_i(s)G_i(s)F_i(s)D_i(s)s^2, \qquad \text{for } i > 1 \tag{5.26}$$

The transfer functions (5.25a,b,c) are referred to as the input, the output and the error string stability transfer functions, respectively. Strikingly, for homogeneous traffic, where $G_i(s) = G(s)$, $K_i(s) = K(s)$, etc. for $i > 1$, the string-stability transfer functions (5.25a,b,c) are equal, i.e., $\mathcal{G}_{U,i}(s) = \mathcal{G}_{X,i}(s) = \mathcal{G}_{E,i}|_{i>2}(s)$. In literature, often, homogeneous traffic is considered, while the different string-stability transfer functions (5.25a,b,c) are mixed up, see, e.g., (Rajamani and Zhu, 2002; Swaroop and Hedrick, 1999). Focusing on heterogeneous traffic, where $G_i(s) \neq G_j(s)$, $K_i(s) \neq K_j(s)$, etc. for $i, j > 1$, $i \neq j$, the string-stability transfer functions (5.25a,b,c) are clearly different.

Consider a platoon of heterogeneous vehicles for which the system parameter values are given in Table 5.1. Apart from the headway time, the parameters are chosen different for each vehicle. Substituting $G_i(s)$ (5.3), $K_i(s)$ (5.6), $H_i(s)$ (5.11), $D_i(s)$ (5.13) and $F_i(s)$ (5.15) in the string-stability transfer function (5.25) and using the system specifications of Table

Table 5.1: System parameters for a platoon of heterogeneous vehicles.

i vehicle	$k_{G,i}$ [-]	τ_i [s/rad]	ϕ_i [s]	$\omega_{K,i}$ [rad/s]	θ_i [s]	ϵ_i [-]	$h_{d,i}$ [s]
1	0.7	0.1	0.0	3.0	0.1	1.0	1.0
2	1.0	0.5	0.1	0.3	0.3	0.8	1.0
3	1.3	0.4	0.3	1.0	0.0	0.5	1.0
4	0.9	1.0	0.1	0.3	0.2	0.9	1.0

5.1 yields the Bode magnitude plots shown in Figure 5.7. As these plots show, the low-frequent asymptotic values of $|\mathcal{G}_{U,i}(j\omega)|$ and $|\mathcal{G}_{E,i}(j\omega)|$ differ per vehicle, whereas this is not the case for $|\mathcal{G}_{X,i}(j\omega)|$. Computing these values gives

$$\lim_{\omega\to 0} |\mathcal{G}_{U,i}(j\omega)| = \frac{k_{G_{i-1}}}{k_{G_i}}, \qquad \text{for } i > 1 \tag{5.27a}$$

$$\lim_{\omega\to 0} |\mathcal{G}_{X,i}(j\omega)| = 1, \qquad \text{for } i > 1 \tag{5.27b}$$

$$\lim_{\omega\to 0} |\mathcal{G}_{E,i}(j\omega)| = \begin{cases} \dfrac{\epsilon_2 k_{G_1}\omega_{K_1^2}}{k_{G_2}\omega_{K_2}^2}, & \text{for } i = 2 \\ \dfrac{\epsilon_i k_{G_{i-1}}\omega_{K_{i-1}^2}}{\epsilon_{i-1} k_{G_i}\omega_{K_i^2}}, & \text{for } i > 2 \end{cases} \tag{5.27c}$$

where

$$\epsilon_i = \lim_{\omega\to 0} |1 - \Xi_i(j\omega)| = 1 - \frac{k_{G_i}}{\hat{k}_{G_i}}, \qquad \text{for } i > 1 \tag{5.28}$$

with $\hat{k}_{G_i}$ an estimate for $k_{G,i}$, which is used in the feedforward filter $F_i(s)$. In practice, $\hat{k}_{G_i} \neq k_{G_i}$ holds, yielding $\epsilon_i \neq 0$, for $i > 1$.

As (5.27a) and (5.27c) show, the conditions for input as well as for error string-stable behavior of vehicle i, for $i > 1$, depend on the characteristics of the dynamics of vehicle $i-1$. Without knowledge of these characteristics, no theoretical guarantees regarding input or error string stability can be given. As these characteristics are not known a priori, the design of a string-stable CACC system is complicated, and, probably conservative performance requirements have to be adopted. Moreover, depending on the characteristics of the dynamics of vehicle $i-1$ with respect to the dynamics of vehicle i, it may be infeasible to guarantee input or error string-stable behavior for vehicle i. For the output string-stability transfer function $\mathcal{G}_{X,i}(s)$, string stability for vehicle i, $i > 1$, can be guaranteed regardless of the dynamics of vehicle $i-1$. Based on this result, it is concluded that for the CACC setup as presented in Section 5.3 and heterogeneous traffic, output string stability has to be considered.

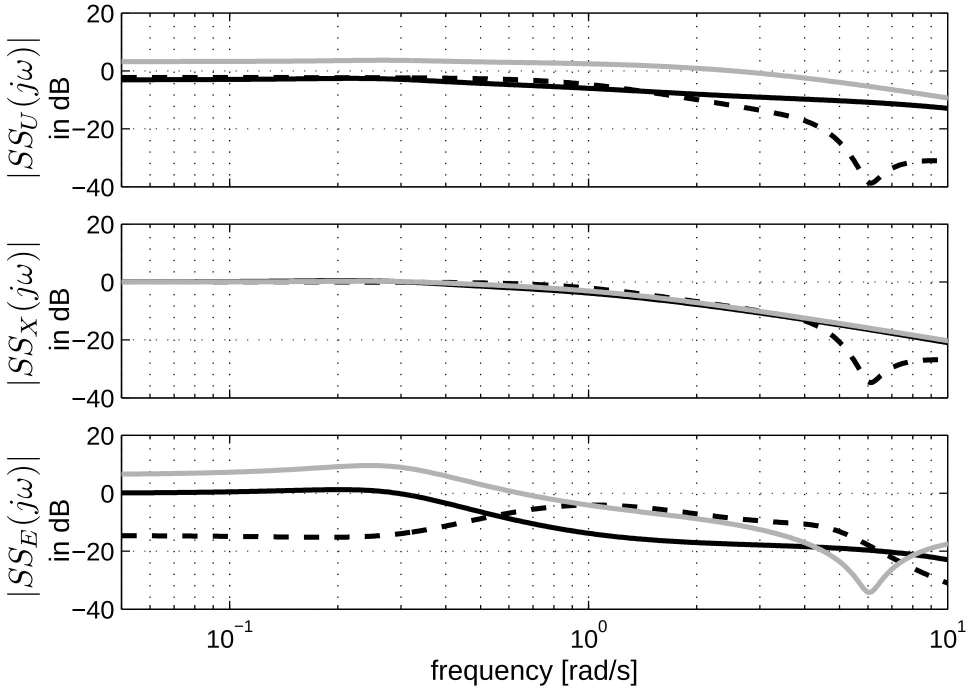


Figure 5.7: Bode magnitude plots of the string-stability transfer functions (5.25a,b,c), corresponding to three heterogeneous CACC-equipped vehicles $i \in \{2, 3, 4\}$ (solid black, dashed black, and solid grey, respectively), driving in a platoon. The system parameters for each vehicle are listed in Table 5.1.

Correspondingly, the sufficient condition for string stability (5.22), with $\Lambda = X$, is adopted as a necessary condition for string stability (see, e.g., Huppe et al., 2003), yielding

$$\|\mathcal{G}_{X,i}(j\omega)\|_\infty = \left\| \frac{X_i(j\omega)}{X_{i-1}(j\omega)} \right\|_\infty \leq 1, \qquad \text{for } i > 1 \tag{5.29}$$

where $\mathcal{G}_{X,i}(s)$ as defined in (5.25b), and $\|\mathcal{G}_{X,i}(j\omega)\|_\infty = 1$, for $i > 1$, $\omega > 0$, is denoted as marginal string stability.

It has to be remarked that this frequency-domain condition for string stability considers amplification of signals, i.e., condition (5.29) does not guarantee the absence of overshoot, considering the desired inter-vehicle distance, in the time domain. A corresponding time-domain condition for string stability, however, does guarantee that overshoot is avoided (Lu et al., 2004)

$$\frac{\|x_i(t)\|_\infty}{\|x_{i-1}(t)\|_\infty} \leq 1, \qquad \text{for } i > 1 \tag{5.30}$$

where $\|\cdot\|_\infty$ denotes the maximum amplitude for all time, i.e.,

$$\|x_i(t)\|_\infty = \sup_t |x_i(t)| \tag{5.31}$$

Following Lu et al. (2004), a necessary and sufficient condition to guarantee this time-domain condition (5.30) is given by

$$\|g_{x,i}(t)\|_1 \leq 1, \qquad \text{for } i > 1 \tag{5.32}$$

where $g_{x,i}(t) = \mathcal{L}^{-1}(\mathcal{G}_{X,i}(s))$ denotes the impulse response of $\mathcal{G}_{X,i}(s)$, and $\|\cdot\|_1$ denotes the 1-norm over all t. Linear system theory yields (Lu et al., 2004; Khatir and Davison, 2004)

$$\|\mathcal{G}_{X,i}(s)\|_\infty \leq \|g_{x,i}(t)\|_1, \qquad \text{for } i > 1 \tag{5.33}$$

Hence, condition (5.32) is a stronger condition, in the sense that condition (5.29) is a necessary, but not sufficient condition to guarantee the time-domain condition (5.30). However, condition (5.29) provides a suitable basis for controller design, while overshoot is limited. Hence, further analysis of the time-domain condition (5.30) is out of the scope of this research and condition (5.29) is considered as a necessary and sufficient condition for string-stability.

5.5 System analysis focusing on string stability

Consider the CACC system setup as presented in Section 5.3, with the feedback controller $K_i(s)$ (5.6), the feedforward filter $F_i(s)$ (5.15), and the spacing policy dynamics $H_i(s)$ (5.11). Furthermore, assume that the low level longitudinal vehicle dynamics (5.3) are ideal, yielding $G_i(s) = s^{-2}$. Hence, the design variables are the feedback controller breakpoint frequency $\omega_{K,i}$, the feedforward filter $F_i(s)$, and the headway time $h_{d,i}$.

In this section, the influence of these design variables on string stability of the proposed CACC systems setup is analyzed for these idealized vehicle dynamics. Experimental validation with real, non-ideal vehicle dynamics is discussed in the next section. Note that the string-stability transfer function (5.25b), and, hence, the string-stability condition (5.29) are independent of the dynamics of other vehicles in the platoon. Consequently, although idealized vehicle dynamics are considered, the analysis holds for homogeneous as well as heterogeneous traffic.

5.5.1 Constant, velocity-independent, inter-vehicle spacing

Lemma 5.5.1 *Consider the control setup as presented in Section 5.3, ideal vehicle dynamics $G_i(s) = s^{-2}$ and assume that a controller $K_i(s)$ with $\omega_{K,i} > 0$ is designed that renders the corresponding closed-loop $T_i'(s)$ (5.4) stable, where the corresponding open-loop $L_i'(s) = K_i(s)G_i(s)$ is a proper transfer function. Given the string-stability condition (5.29), only marginal string stability can be guaranteed for a constant, velocity-independent inter-vehicle spacing, i.e., $h_{d,i} = 0.0$ s.*

Proof To start with, consider the case of no feedforward, i.e., $F_i(s) = 0$. Without feedforward, an ACC system instead of a CACC system results. A constant, velocity-independent, inter-vehicle spacing implies $h_{d,i} = 0.0$ s, yielding $H_i(s) = 1$. Correspondingly, the output string-stability transfer function (5.25b) reduces to

$$\mathcal{G}_{X,i}(s) = \frac{G_i(s)K_i(s)}{1 + G_i(s)K_i(s)} = T_i'(s), \qquad \text{for } i > 1 \tag{5.34}$$

where $T_i'(s)$ is the complementary sensitivity of the inner control loop (see Section 5.3.1). For $T_i'(s)$ a stable system, with the corresponding open-loop $L_i'(s) = K_i(s)G_i(s)$ a proper transfer function, it holds that, if the magnitude of $T_i'(s)$ is smaller than 1 over some frequency range, it will always be larger than 1 in another frequency range, which is due to the well-known Bode-sensitivity-integral constraint (Seron et al., 1997; Bode, 1945). Hence, only marginal string stability can be guaranteed $\|T_i'(j\omega)\|_\infty = |T_i'(j\omega)| = 1$, for $i > 1, \omega > 0$.

Next, consider the case with the feedforward filter $F_i(s)$ as defined in (5.15). If the communication delay equals $\theta_i = 0.0$ s, yielding $D_i(s) = 1$, the string-stability transfer function (5.25b) becomes

$$\mathcal{G}_{X,i}(s) = \frac{1 + G_i(s)K_i(s)}{1 + G_i(s)K_i(s)} = 1, \qquad \text{for } i > 1 \tag{5.35}$$

Hence, in that case, only marginal string stability $\|\mathcal{G}_{X,i}(j\omega)\|_\infty = 1$, for $i > 1$, $\omega > 0$, can be guaranteed. If a communication delay $\theta_i > 0.0$ s is present, the string stability condition (5.29) becomes

$$\left| \frac{D_i(j\omega) + G_i(j\omega)K_i(j\omega)}{1 + G_i(j\omega)K_i(j\omega)} \right| \leq 1, \qquad \text{for } i > 1,\ \forall\omega \tag{5.36}$$

Substituting $D_i(j\omega)$ (5.13) and, without loss of generality, $G_i(j\omega)K_i(j\omega) = f_i(\omega) + j\, g_i(\omega)$, gives

$$\left| e^{-\theta_i j\omega} + f_i(\omega) + j\, g_i(\omega) \right| \leq \left| 1 + f_i(\omega) + j\, g_i(\omega) \right|, \qquad \text{for } i > 1,\ \forall\omega \tag{5.37}$$

which yields

$$\begin{cases} \dfrac{g_i(\omega)}{f_i(\omega)} \sin(-\theta_i\omega) & \leq & 1 - \cos(-\theta_i\omega), & \text{for } i > 1,\ \forall\omega,\ f_i(\omega) > 0 \\ 2g_i(\omega)\sin(-\theta_i\omega) & \leq & 0, & \text{for } i > 1,\ \forall\omega,\ f_i(\omega) = 0 \\ \dfrac{g_i(\omega)}{f_i(\omega)} \sin(-\theta_i\omega) & \geq & 1 - \cos(-\theta_i\omega), & \text{for } i > 1,\ \forall\omega,\ f_i(\omega) < 0 \end{cases} \tag{5.38}$$

Standard goniometry shows that (5.38) is only true for $f_i(\omega) \geq 0$ and $g_i(\omega) = 0$. Moreover, in that case, only marginal string stability can be guaranteed. In all other cases, string-stability cannot be guaranteed.

■

5.5.2 Velocity-dependent inter-vehicle spacing, ACC case

Lemma 5.5.2 *Consider the ACC control setup as presented in Section 5.3.2, and ideal vehicle dynamics $G_i(s) = s^{-2}$. Given the string-stability condition (5.29), string stability can be guaranteed for $h_{d,i} \geq h_{d,i,min}(\omega_{K,i})$.*

Proof In the case of an ACC system, no feedforward is present, i.e., $F_i(s) = 0$. For $h_{d,i} > 0.0$ s, the output string-stability transfer function (5.25b) equals

$$\mathcal{G}_{X,i}(s) = \frac{G_i(s)K_i(s)}{1 + H_i(s)G_i(s)K_i(s)}, \qquad \text{for } i > 1 \tag{5.39}$$

Substituting $G_i(s) = s^{-2}$, $K_i(s)$ (5.6), and $H_i(s)$ (5.11), in (5.39), the string-stability condition (5.29) yields

$$\frac{\omega_{K,i}^2(2 - \omega_{K,i}^2 h_{d,i}^2)}{(1 + \omega_{K,i}h_{d,i})^2} \leq \omega^2, \qquad \text{for } i > 1,\ \forall \omega \tag{5.40}$$

As $\omega \in \mathbb{R}^+$ holds, this implies $\min\{\omega^2\} = 0$, leading to

$$\frac{\omega_{K,i}^2(2 - \omega_{K,i}^2 h_{d,i}^2)}{(1 + \omega_{K,i}h_{d,i})^2} \leq 0, \qquad \text{for } i > 1 \tag{5.41}$$

From (5.41) it follows directly that string stability can be guaranteed for

$$h_{d,i} \geq h_{d,i,min} = \sqrt{2}\omega_{K,i}^{-1}, \qquad \text{for } i > 1 \tag{5.42}$$

■

The result (5.42) clearly depends on the design of $K_i(s)$. Moreover, the result only holds for ideal vehicle dynamics and the spacing policy dynamics as defined in (5.8). A different design of $K_i(s)$, for non-ideal vehicle dynamics $G_i(s)$, or a different choice for $x_{r,d,i}(t)$ would yield a different result. Hence, it has to be noted that the minimal headway time as defined in (5.42) does not reflect the absolute minimum that is feasible with the proposed ACC setup.

Example Take $\omega_{K,i} = 0.5$ rad/s. Consider ideal vehicle dynamics $G_i(s) = s^{-2}$, implying $k_{G,i} = 1.0$, $\tau_i = 0$ and $\phi_i = 0.0$ s. The bandwidth $\omega'_{bw,i}$ and the phase margin $\alpha'_{PM,i}$ corresponding to $T'_i(s)$ are then straightforwardly derived as $\omega'_{bw,i} \approx 0.6$ rad/s and $\alpha'_{PM,i} \approx 51^\circ$, respectively (see Section 5.3). Following (5.42), $h_{d,i} \geq h_{d,i,min} \approx 2.8$ s has to hold to ensure string stability.

This is illustrated in Figure 5.8, showing simulation results corresponding to a platoon of three ACC-equipped vehicles following a reference vehicle. For clarity, homogeneous traffic is considered. For $h_{d,i} = 3.0$ s (the upper figure), all vehicles in the platoon follow the reference vehicle while decreasing the amplitude of the velocity signal. For $h_{d,i} = 1.0$ s (the bottom figure), the amplitude of the velocity signal is amplified upstream the platoon, which represents string-unstable behavior. In Figure 5.9(a), the corresponding Bode magnitude plots of $\mathcal{G}_{X,i}(j\omega)$ are shown, confirming these results.

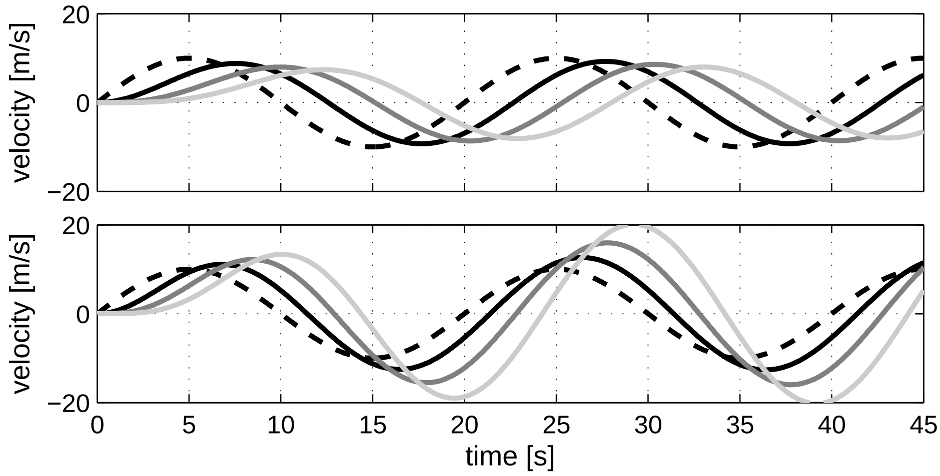


Figure 5.8: Simulation results of a platoon of three ACC-equipped vehicles $i = 1$ to 3 (solid dark to light grey, respectively), following a reference vehicle (dashed). The results in the upper and bottom figure correspond to $h_{d,i} = 3.0$ s and $h_{d,i} = 1.0$ s, for $i \in \{1, 2, 3\}$, respectively.

5.5.3 Velocity-dependent inter-vehicle spacing, CACC case

Lemma 5.5.3 *Consider the control setup as presented in Section 5.3.3, ideal vehicle dynamics $G_i(s) = s^{-2}$, and a specific operating range $(\omega_{K,i}, \theta_i) \in \Omega_{K,O} \times \Theta_O = [0.1,\ 2.0]$ rad/s $\times$ $[0.0,\ 0.5]$ s, $i > 1$. Given the string-stability condition (5.29), string stability can be guaranteed for $h_{d,i} \geq h_{d,i,min}(\omega_{K,i}, \theta_i)$.*

Proof Substituting the feedforward filter $F_i(s)$ (5.15) in (5.25b) yields

$$\mathcal{G}_{X,i}(s) = \frac{D_i(s) + H_i(s)G_i(s)K_i(s)}{H_i(s)(1 + H_i(s)G_i(s)K_i(s))}, \qquad \text{for } i > 1 \tag{5.43}$$

In the case of no communication delay, i.e. $\theta_i = 0.0$ s, yielding $D_i(s) = 1$, (5.43) reduces to

$$\mathcal{G}_{X,i}(s) = \frac{1}{H_i(s)}, \qquad \text{for } i > 1 \tag{5.44}$$

Hence, for a velocity-dependent inter-vehicle spacing with $h_{d,i} > 0.0$ s, the string-stability condition (5.29) is directly fulfilled.

If a communication delay $\theta_i > 0.0$ s is taken into account, the analytical derivation of the minimal required headway time $h_{d,i,min}$ becomes rather complex and does not provide additional insight. Consequently, this derivation is not discussed here. Instead, in Figure 5.10, the results of a numerical approximation of $h_{d,i,min} = h_{d,i,min}(\omega_{K,i}, \theta_i)$ are shown for $(\omega_{K,i}, \theta_i) \in \Omega_{K,O} \times \Theta_O = [0.1,\ 2.0]$ rad/s $\times$ $[0.0,\ 0.5]$ s. The results in Figure 5.10 show

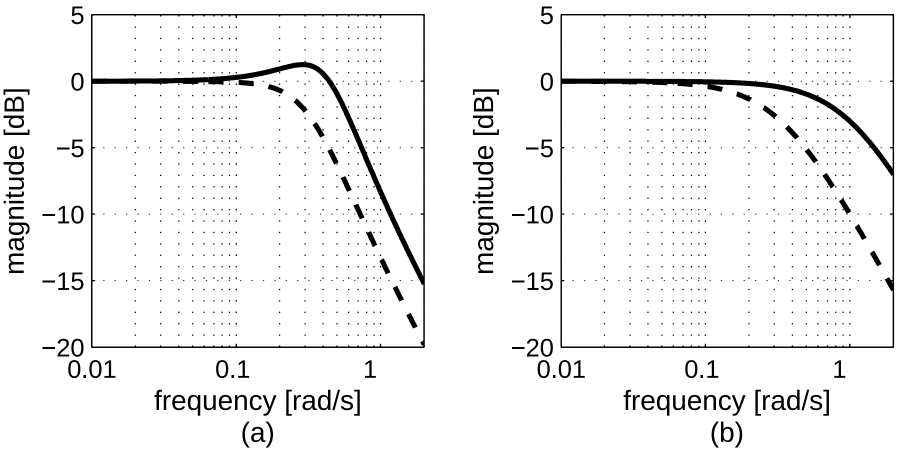


Figure 5.9: Bode magnitude plots of $\mathcal{G}_{X,i}(j\omega)$ (5.25b), corresponding to (a) the ACC case, and (b) the CACC case, for $h_{d,i} = 1.0$ s (solid black), and $h_{d,i} = 3.0$ s (dashed black).

that, depending on $\omega_{K,i}$ and on the size of the time delay in the wireless communication signal θ_i, string stability is guaranteed for $h_{d,i} > h_{d,i,min}(\omega_{K,i}, \theta_i)$, where, following (5.44), $\lim_{\theta_i \to 0} h_{d,i,min}(\omega_{K,i}, \theta_i) = 0.0$ s holds. ∎

If $\omega_{K,i}$ and θ_i are known, the minimum value $h_{d,i} = h_{d,i,min}(\omega_{K,i}, \theta_i)$ required to guarantee string stability in the case of a CACC system follows directly from Figure 5.10. The results in Figure 5.10 clearly depends on the design of $K_i(s)$. Moreover, the result only holds for ideal vehicle dynamics and the spacing policy dynamics as defined in (5.8). A different design of $K_i(s)$, for non-ideal vehicle dynamics $G_i(s)$, or a different choice for $x_{r,d,i}(t)$ would yield a different result. Hence, it has to be noted that the minimal headway time as presented in Figure 5.10 does not reflect the absolute minimum that is feasible with the proposed ACC setup. Nevertheless, comparing this to the minimal headway time that is required in the case of an ACC system (see (5.42)), for a reasonably-valued delay time θ_i, the minimal headway time required to guarantee string stability in case of a CACC system is significantly smaller.

Example Take $\theta = 200$ ms, and, analogous to the example in Section 5.5.2, $\omega_{K,i} = 0.5$ rad/s. Figure 5.10 indicates that string stability can be guaranteed for $h_{d,i} \gtrsim 0.8$ s. This value is clearly smaller than the minimum value required to achieve string stability in the case of an ACC system, which is $h_{d,i,min} \approx 2.8$ s (see Section 5.5.2).

In Figure 5.11, the simulation results of a platoon of CACC-equipped vehicles where $h_{d,i} = 1.0$ s is used, are shown. As these results show, all vehicles in the platoon follow the reference vehicle while decreasing the amplitude of the velocity of preceding vehicles in the platoon, as opposed to the corresponding results shown in Figure 5.8 in which an

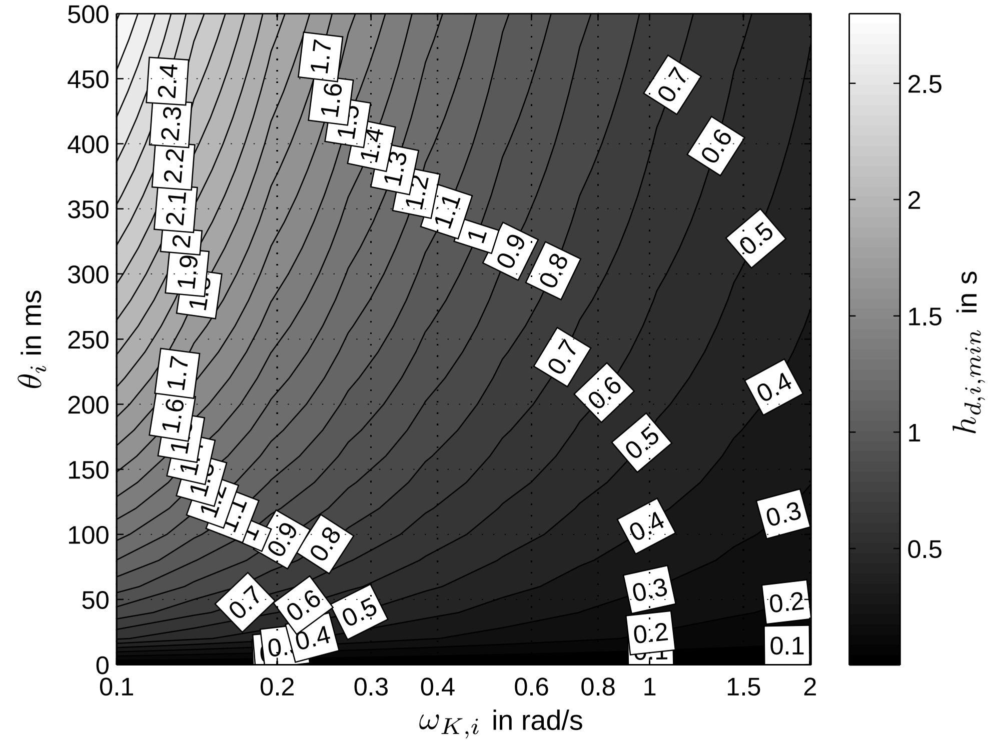


Figure 5.10: Contour plot of $\omega_{K,i}$ versus θ_i, indicating the corresponding minimal value for $h_{d,i,min}(\omega_{K,i}, \theta_i)$ for which string stability is ensured.

ACC system is employed. The corresponding Bode magnitude plot is shown in Figure 5.9(b), confirming this string stable driving behavior result.

With respect to the reference vehicle, however, string-unstable behavior is observed (see Figure 5.11). The reference vehicle, driving in front of the platoon, is assumed not to communicate. Consequently, it is followed using ACC only. For the $h_{d,i} = 1.0$ s used, this yields string-unstable behavior, as observed before.

5.6 Experimental validation

To validate the theory, experiments are performed using two vehicles. The models for the communication delay and the vehicle dynamics are identified using measurements. Based on these models, the designs of the feedback controller, the feedforward filter, and the spacing policy dynamics are discussed. Three different experiments are executed, including experiments with and without wireless communication, and experiments with different headway times (Naus et al., 2009b).

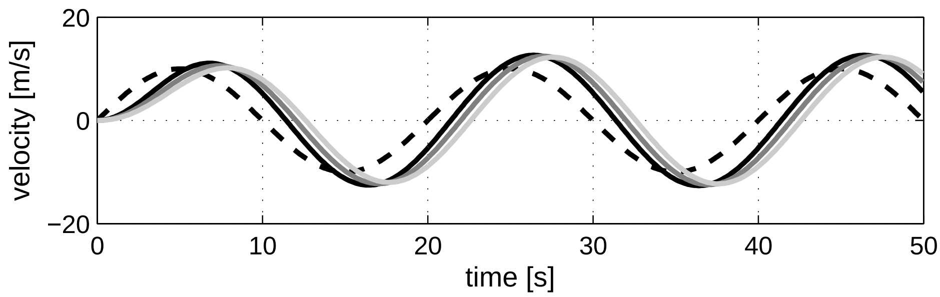


Figure 5.11: Simulation results of a platoon of three CACC-equipped vehicles $i = 1$ to 3 (solid dark to light grey, respectively), following a reference vehicle (dashed). The results correspond to $h_{d,i} = 1.0$ s, for $i \in \{1, 2, 3\}$.

5.6.1 Experimental setup

Two Citroën C4's are used as a testing platform, see Figure 5.12. For the wireless inter-vehicle communication, the standard Wi-Fi protocol IEEE 802.11g is used, with an update rate of 10 Hz (IEEE Computer Society, 2003). The acceleration of vehicle 1 is derived from the built-in anti-lock braking system (ABS), which is available on the CAN-bus, and communicated to vehicle 2. A zero-order-hold approach is adopted for the communicated signal, introducing a corresponding average delay of about 50 ms. To synchronize the measurements of the two vehicles, GPS time stamping is adopted. Correspondingly, an additional communication delay of about 10 ms is identified. Combination of these values yields $\theta_i \approx 60$ ms as a total delay for the model $D_i(s)$ (5.13).

Vehicle 2 is equipped with an electro-hydraulic braking (EHB) system, facilitating brake-by-wire control. Implementation of the controller for the longitudinal dynamics of the vehicle as well as actuation of the throttle and EHB system are covered by the TNO modular automotive control system (MACS) (TNO Automotive, 2009). An OMRON laser radar, i.e., a lidar, with 150 m range is built-in. Using rapid control prototyping, the CACC system is implemented on a dSpace AutoBox, with a sample rate of 100 Hz. Finally, a laptop is used to monitor all signals and log the data. A schematic overview of the instrumentation is shown in Figure 5.13.

5.6.2 Vehicle model identification

In the system analysis presented in Section 5.5, it is assumed that the low level closed-loop dynamics of the vehicle $G_i(s)$ (5.3) are ideal, i.e., $G_i(s) = s^{-2}$. In practice, however, this does not hold. Open-loop step-response measurements are performed to identify these dynamics. Measurements are performed for various acceleration levels and at different velocities. A least-squares minimization method is used to identify the parameters of the

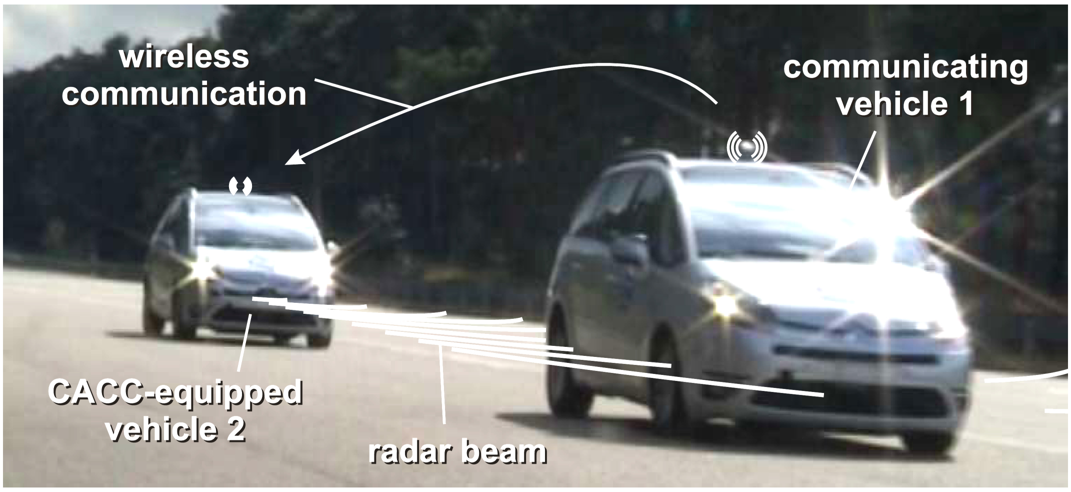


Figure 5.12: Experimental setup with two Citroën C4's (Naus et al., 2009b).

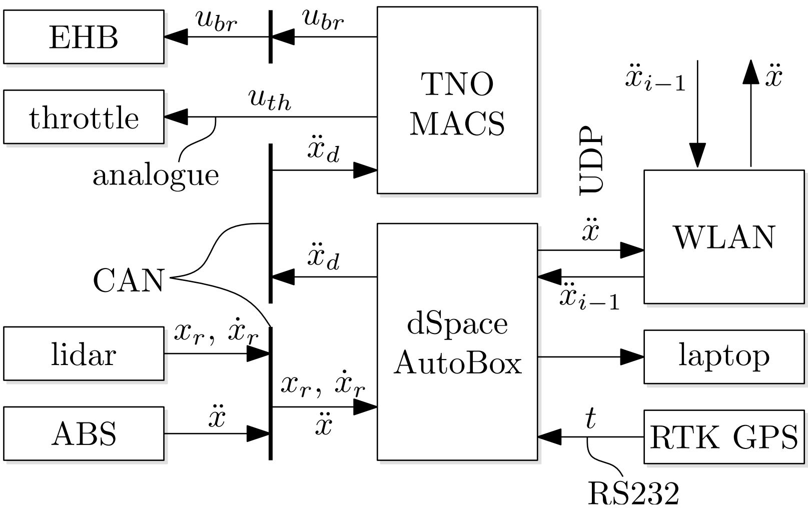


Figure 5.13: Schematic overview of the instrumentation of the vehicles. The main communication channels and corresponding signals are indicated, where $u_{\text{th}}(t)$ and $u_{br}(t)$ are the throttle and brake system control signals, respectively, $\ddot{x}_d(t)$ and $\ddot{x}(t)$ are the desired and actual acceleration, respectively, $x_r(t)$ and $\dot{x}_r(t)$ are the relative position and velocity, respectively, $\ddot{x}_{i-1}(t)$ is the communicated acceleration of the preceding vehicle, and t is the time stamping signal. For clarity, both the index i, indicating vehicle i, and the time dependency of the signals are omitted.

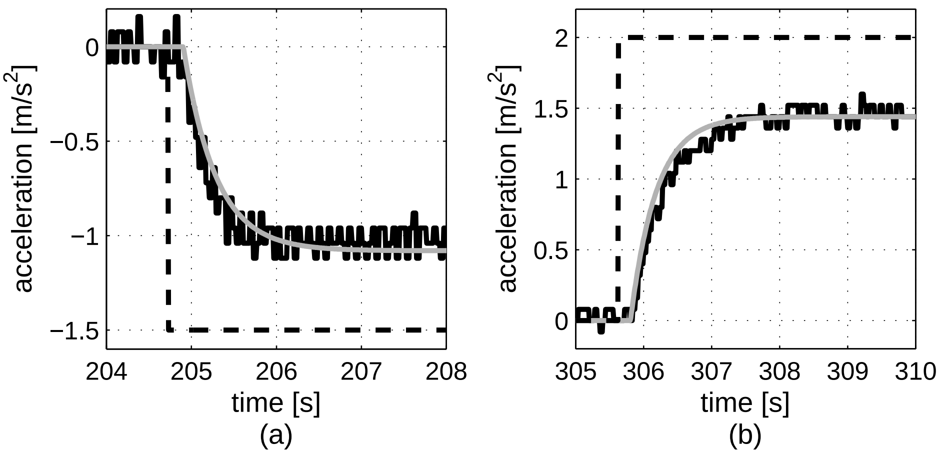


Figure 5.14: Identification step-response measurement results for (a) braking, and (b) accelerating. The reference step input (dashed black), the measurement results (solid black) and the corresponding simulation results with the model $G_i^*(s) = G_i(s)s^2$ (solid grey) are shown.

model for the vehicle dynamics (5.3). This identification results in a bandwidth $\omega_{bw,l,i} = \tau_i^{-1} = 0.38$ rad/s, a delay time $\phi_i = 0.18$ s, and a system gain $k_{G,i} = 0.72$. Two of the measurement results and corresponding simulation results with the identified model $G_i^*(s) = G_i(s)s^2$ are shown in Figure 5.14.

Comparing the measurement and the simulation results shows that the model represents the longitudinal vehicle dynamics appropriately. The same holds for validation measurements, of which an example is shown in Figure 5.15(a). Hence, the proposed model structure for $G_i(s)$ (5.3) is sufficient to describe the low level vehicle dynamics. The main differences between the identified vehicle model $G_i(s)$ and the ideal vehicle dynamics considered in Section 5.5 are in the low-frequent gain $k_{G,i} \neq 1.0$, and the actuator delay $\phi_i \neq 0.0$ s. The consequences of these differences are evaluated later on.

Attention should be paid to the size of the acceleration signal. Measurements show that the acceleration saturates at about 1.8 m/s^2. Simulation results show that the model $G_i^*(s)$ including this saturation represents the saturated vehicle dynamics appropriately, see Figure 5.15(b). However, nonlinearities, such as this saturation, are not accounted for in the presented frequency-domain CACC design and the corresponding string-stability analysis. Hence, to be able to validate the theory presented in Section 5.5, it has been taken care of that the desired acceleration does not exceed this saturation limit during the experiments. Comparing the results of Figure 5.15(a) and (b) shows that the validation measurements in Figure 5.15(a) are already close if not at the saturation limit. However, the corresponding simulation model without the saturation limit still resembles the measurement results good.

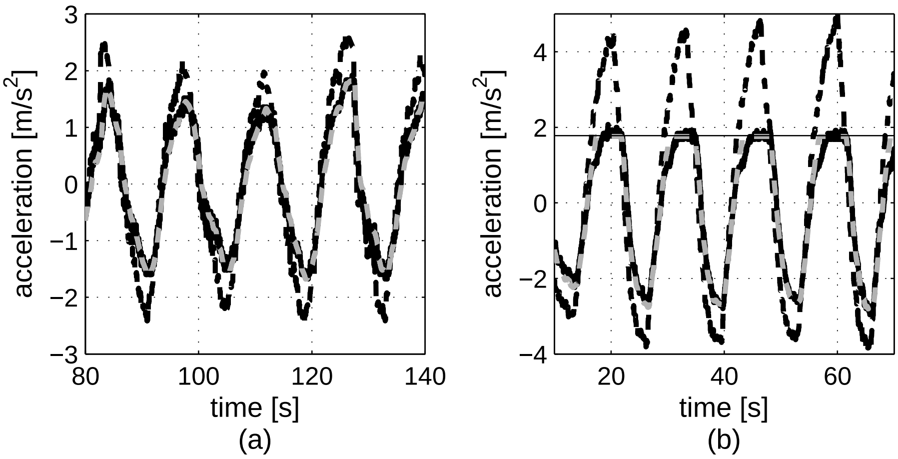


Figure 5.15: Validation measurement results for the identified vehicle model (a) without, and (b) with saturation. The desired acceleration (dashed black), the measured acceleration (solid black), the simulated model response (dashed grey), and the saturation limit (solid black) are shown. The simulated response covers most of the measured acceleration signal.

5.6.3 CACC design

The CACC design follows the setup as proposed in Section 5.5. The implemented feedback controller $K_i(s)$ is a combination of the PD-controller (5.6) and a first-order low-pass filter to prevent amplification of high-frequent noise, which is present in the lidar measurements. Furthermore, for the purpose of comparison with the results of the previous section, the identified low-frequent gain $k_{G,i}$ is compensated for by adjusting the proportional action, yielding

$$K_i(s) = \frac{\omega_{K,i}}{k_{G,i}} \frac{\omega_{K,i} + s}{\omega_{f,i} + s}, \qquad \text{for } i > 1 \tag{5.45}$$

where $\omega_{f,i}$ is the cut-off frequency of the low-pass filter, which equals half the sample frequency, i.e., $\omega_{f,i} = 100.0\pi$ rad/s. Analogous to the example in Section 5.5.2, $\omega_{K,i} = 0.5$ rad/s is used.

Combining the resulting controller with the identified vehicle model yields the open-loop response shown in Figure 5.16. The corresponding closed-loop bandwidth of the inner control loop equals $\omega'_{bw,i} \approx 0.6$ rad/s with a phase margin $\alpha'_{PM,i} \approx 1.0$ rad/s $\approx 30°$. Hence, a reasonable phase margin results and, moreover, $\omega_{bw,l,i} > \omega'_{bw,i}$ holds (see Section 5.6.2). This indicates that the design is robustly stable and that the tracking performance is not degraded by the low level longitudinal dynamics.

For the feedforward controller $F_i(s)$, the design as proposed in (5.15) is used, where the identified model for the low-level longitudinal vehicle dynamics is used (see Section

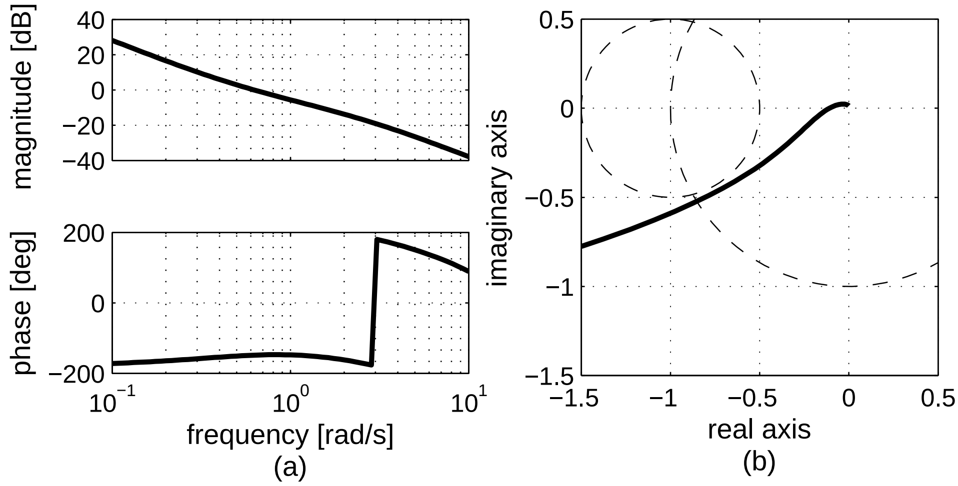


Figure 5.16: (a) Bode, and (b) Nyquist plot of the open-loop system $G_i(s)K_i(s)$.

5.6.2). For the spacing policy, an additional low-pass filter is used to filter the velocity measurements of the lidar, which are used to compute the desired inter-vehicle distance $x_{r,d,i}(t)$ (5.8). The cut-off frequency of the low-pass filter is chosen about ten times larger than the bandwidth of the inner-loop bandwidth, i.e., 5.0 rad/s. Taking a higher cut-off frequency has no practical relevance, as the inner control loop is not able to track the corresponding desired relative position $x_{r,d,i}(t)$. A lower cut-off frequency would result in too much phase lag.

5.6.4 String-stability experiments

To validate the theory of Section 5.5, three experiments are performed, which are listed in Table 5.2: with and without using wireless communication, i.e., CACC and ACC, respectively, and for different headway times $h_{d,i}$.

Using the string-stability condition (5.29), the minimal headway times that are required to ensure string stability in the case of ACC and CACC are computed numerically, being $h_{d,i,min} = 2.6$ and 0.8 s, respectively. In Figure 5.17, Bode magnitude plots of the string-stability transfer function $\mathcal{G}_{X,i}(s)$ (5.25b) are shown for $h_{d,i} \in \{0.5, 1.0, 3.0\}$ s, in the case of ACC and CACC. The Bode plots confirm the numerical results, indicating string stability both in the case of ACC with a headway time $h_{d,i} = 3.0 > 2.6$ s, and in the case of CACC with a headway time $h_{d,i} \in \{1.0, 3.0\} > 0.8$ s. However, both in the case of ACC with a headway time $h_{d,i} \in \{0.5, 1.0\} < 2.6$ s and in the case of CACC with a headway time of $h_{d,i} = 0.5 < 0.8$ s, the string-stability condition is not fulfilled. Consequently, string-stable driving behavior is expected in both Experiments i and iii, whereas string-unstable behavior is expected in Experiment ii.

Table 5.2: Overview of the experiments, where $h_{d,i,min}$ is the minimal headway time required for string stability and $h_{d,i}$ is the headway time used in the experiments.

Experiment	wireless communication	$h_{d,i,min}$ [s]	$h_{d,i}$ [s]
i	no	2.6	3.0
ii	no	2.6	1.0
iii	yes	0.8	1.0

Comparing the Bode magnitude plots of Figure 5.9 and 5.17 shows that, in this case, the influence of the non-ideal vehicle dynamics on the string stability transfer functions is small. This can be attributed to the compensation for the vehicle model gain in the feedback controller (see (5.45)), and the relatively small communication delay (see Section 5.6.2). The main difference can be related to a relatively large actuator delay $\phi_i = 0.18$ s (see Section 5.6.2).

The experiments are performed with two vehicles, (see Figure 5.12). To increase insight in the measurement results, the response of vehicle 2 is also simulated offline. Analogously, the platoon is virtually extended by simulating an additional two vehicles $i \in \{3, 4\}$. For the simulations, a model including saturation is used (see Figure 5.15). The input for the simulated vehicle 2 is derived from the communicated acceleration of vehicle 1. Furthermore, for this purpose, the velocity of vehicle 1 is also obtained through wireless communication. The input for the subsequent vehicles follows directly from the simulated vehicle 2. For clarity, homogeneous traffic is considered, i.e., identical vehicle models, identical feedback and feedforward controllers and identical headway times are used.

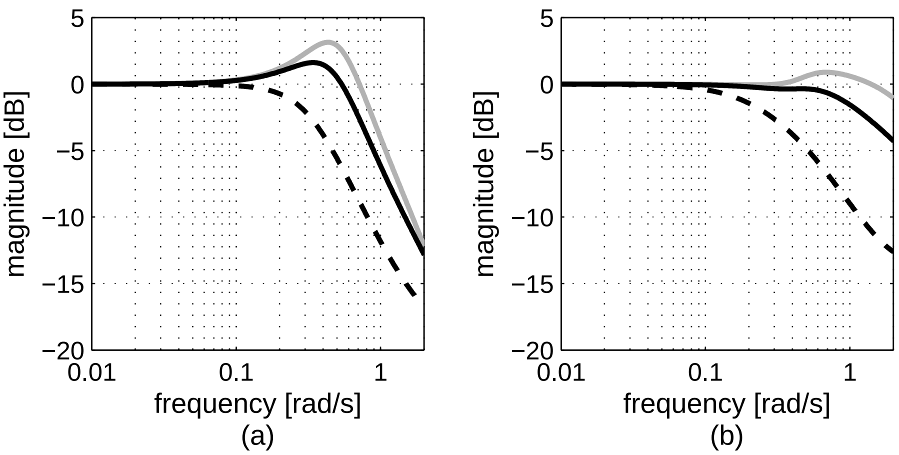


Figure 5.17: Bode magnitude plots of $\mathcal{G}_{X,i}(j\omega)$ (5.43), in the case of (a) ACC, and (b) CACC, for $h_{d,i} = 0.5$ s (solid grey), $h_{d,i} = 1.0$ s (solid black), and $h_{d,i} = 3.0$ s (dashed black).

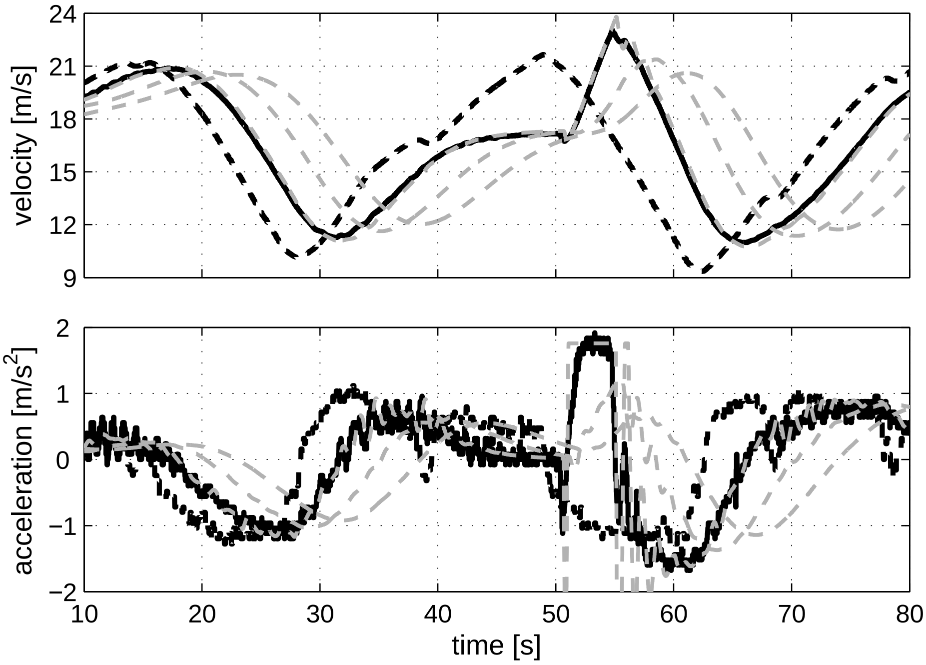


Figure 5.18: Measurement results of Experiment i (see Table 5.2). In the upper figure, the measured velocities of vehicle 1 (dashed black) and vehicle 2 (solid black), as well as the resulting velocities for the three simulated vehicles (dashed grey) are shown. In the bottom figure, the corresponding acceleration signals are shown.

Both in Experiments i and ii, wireless communication is not used, which means that an ACC system results. In Figure 5.18 and 5.19, measurement results and corresponding simulation results for Experiments i and ii are shown. For Experiment i, the amplitude of the oscillations in the velocity and the acceleration of vehicle 2 is smaller than for vehicle 1, which corresponds to the anticipated string-stable behavior. The results of the simulated vehicles confirm this conclusion. Moreover, the resemblance between the measurement results and the simulated vehicle 2 is good, validating the use of the simulation results. The peak in the measured velocity and acceleration at about 55 s is a result of a loss of the fix of the lidar, resulting in a switch to standard cruise control functionality and maximum acceleration.

For Experiment ii, the behavior of vehicle 2 is, as expected, string unstable: the oscillations in the velocity of vehicle 1 are amplified by vehicle 2. Again, the resemblance between the simulated vehicle 2 and the measurement results is good, validating the use of the results of the simulated vehicles to emphasize this observation. The results of this experiment validate the observation that the use of standard ACC functionality while driving at smaller inter-vehicle distances induces string-unstable behavior, which, for example, results in traffic jams (see Section 5.1).

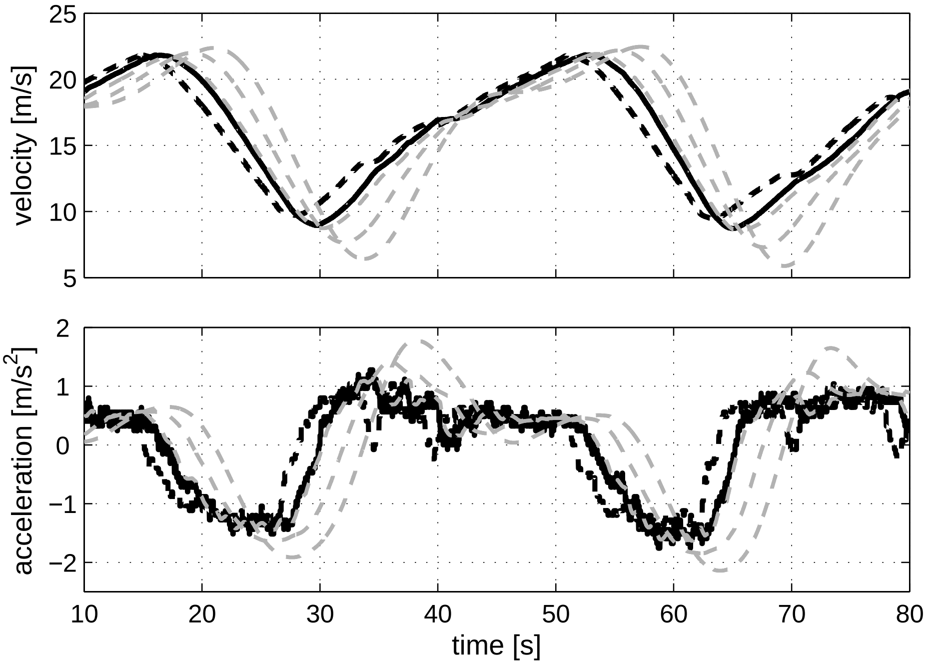


Figure 5.19: Measurement results of Experiment ii (see Table 5.2). In the upper figure, the measured velocities of vehicle 1 (dashed black) and vehicle 2 (solid black), as well as the resulting velocities for the three simulated vehicles (dashed grey) are shown. In the bottom figure, the corresponding acceleration signals are shown.

In Figure 5.20, measurement and simulation results for Experiment iii are shown. The wireless information is used, yielding a true CACC system. As anticipated, the behavior of vehicle 2 is string stable, which is confirmed by the simulation results. Comparing the results for the ACC system in Experiment ii and the CACC system in Experiment iii, validates the conclusion that, given the proposed CACC system setup, the minimal headway time required to guarantee string stability is smaller in the case of CACC, which is enabled by the wireless communication link. Moreover, comparing the results of the ACC experiments and the CACC experiments shows that the tracking performance is significantly improved in case of a CACC system. Considering the fact that the main difference between the ACC and CACC system is a feedforward filter, this result is expected.

5.7 Conclusions and recommendations

The design of a CACC system focusing on the feasibility of implementation is presented: a communication link with the directly preceding vehicle is considered, communication delay as well as heterogeneity of the traffic is taken into account, a decentralized controller

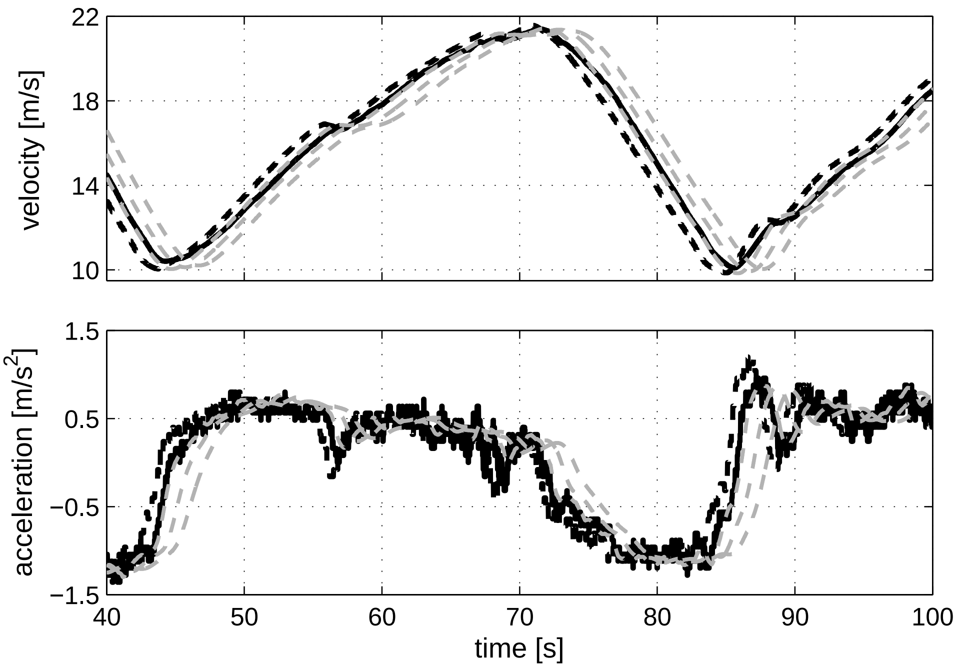


Figure 5.20: Measurement results of Experiment iii (see Table 5.2). In the upper figure, the measured velocities of vehicle 1 (dashed black) and vehicle 2 (solid black), as well as the resulting velocities for the three simulated vehicles (dashed grey) are shown. In the bottom figure, the corresponding acceleration signals are shown.

design is adopted, and, using a feedforward controller design, the system can easily degrade to a standard ACC system in case wireless communication is not available or fails.

Based on this setup, the string-stability characteristics of the CACC system are analyzed using a frequency-domain based approach. This analysis yields the following conclusions:

- output string stability has to be considered when heterogeneous traffic is taken into account,
- a velocity-dependent spacing policy is required to achieve string-stable system behavior,
- the CACC feedforward controller enables small inter-vehicle distances while maintaining string stability, whereas this is not the case for the standard ACC feedback controller alone.

Experimental results with two CACC-equipped vehicles validate these theoretical results. The experimental results illustrate that, with a relatively simple CACC system setup, which focuses on the feasibility of implementation, significant improvements regarding minimal headway time and string-stability characteristics can be achieved with respect to a standard ACC system.

Further experimental validation of the concept using a larger string of vehicles should be part of future research. Besides that, comparison of the performance and string-stability characteristics of the proposed system setup with different CACC setups is an interesting issue for further research. For example, setups adopting different communication structures or different spacing policy dynamics can be compared. Finally, important issues for future research are to include both saturations and robustness against model uncertainties and disturbances in the string-stability analysis, and to consider the behavior of mixed strings with both CACC-equipped vehicles and non-equipped, 'normal' traffic.

CHAPTER 6

Discussion

Abstract - Based on the results of the case studies, a classification of automotive control problems is made. Using this classification, the specification of performance requirements and the required modeling for automotive control problems are discussed. Furthermore, the practical applicability of the controller synthesis methods that are adopted in the case studies is evaluated.

6.1 Introduction

The main goal of this thesis is to contribute to a paradigm shift from the application of pragmatic, online tuning techniques to the application of systematic, model-based control design approaches in the automotive industry. Accordingly, the results of the case studies demonstrate corresponding possibilities and opportunities, compared to the use of online tuning and calibration techniques (see Chapters 2 to 5).

In Table 1.2 (pg. 10), typical challenges in automotive control problems are listed. The case studies consider representative control problems in the automotive industry, in the sense that two of the typical challenges according to Table 1.2 are addressed, being variable operating conditions and global performance qualifiers. Consequently, they can be regarded as exemplary cases for other automotive control problems. The vertical automotive supply chain and corresponding intellectual property (IP) issues are not explicitly considered in this research.

Accordingly, insight is obtained in the properties of control problems in the automotive industry via the case studies. Based on this insight, this chapter targets a discussion on the modeling and the specification of performance requirements for automotive control problems in general. Focus is on the effect of the mentioned challenges from Table 1.2, for the application of a systematic, model-based control design approach.

To facilitate the discussion, a classification of automotive control problems is proposed, distinguishing problems on three levels (Section 6.2). Given this classification, first the specification of performance requirements is discussed (Section 6.3). Second, the required control-oriented modeling is discussed (Section 6.4). Finally, practical applicability of controller synthesis methods that are considered in the case studies is summarized (Section 6.5).

6.2 Classification of automotive control problems

Based on insight in the properties of automotive control problems that is obtained via the case studies, a generic classification of automotive control problems is proposed in this section. In literature, various classifications for automotive control problems are presented (see Gordon et al., 2003). Commonly, focus is on the definition of generic control architectures to handle the increasing number of complex, distributed functionalities (see, e.g., Wills et al., 2001). An example is integrated vehicle control (IVC), which targets a centralized controller design, combining different control functionalities (Coelingh et al., 2002). Especially in the chassis domain, IVC is often considered as a solution to handle the increasing number of functionalities in a modern vehicle (Chang and Gordon, 2008).

The classification proposed here characterizes automotive control problems on the basis of the corresponding control design approaches, i.e., the control-oriented modeling and the specification of corresponding performance requirements that are required to handle the typical challenges in automotive control problems according to Table 1.2. Three levels of automotive control problems are distinguished, being control problems at a full-vehicle level, at an in-vehicle level, and at a component level. Comparing this classification to literature, for example Wills et al. (2001) also distinguish three control levels, being high-level supervision, mid-level control, and low-level control. However, focus of Wills et al. (2001) is on the definition of a general-purpose hierarchical architecture, rather than the characterization of control problems on the basis of the corresponding controller design approaches. Correspondingly, the control design approaches for automotive control problems at a specific level are not generic, nor do they consider the typical challenges according to Table 1.2 in the design.

The classification proposed here is summarized in Table 6.1. Figure 6.1 schematically illustrates the classification.

- Typical examples of control systems at the component level (indicated by $\mathcal{K}_c$) are electro-mechanical and electro-hydraulic actuator systems. Although the actuators actually interconnect the control software and the vehicle hardware, the relevant dynamics are typically constrained to the dynamics of the actuator itself. Hence, the

Table 6.1: Classification of automotive control problems.

classification	characterization	examples
full-vehicle level	consider the vehicle as a whole, defining desired vehicle behavior, typically taking into account the traffic situation, commonly distributed functionality	advanced driver assistance systems, such as ACC (Chapter 4), and CACC (Chapter 5), as well as setpoint generation, e.g., for cruise control (Chapter 3)
in-vehicle level	control of the complex vehicle dynamics, typically embedded systems, both distributed and local control	driver assistance systems, such as cruise control (Chapter 3), and drive-off control (Chapter 2)
component level	separate, stand-alone dynamics, facilitating local control	actuator systems, such as the clutch actuator (Chapter 2)

relevant dynamics at the component level can be regarded as stand-alone and corresponding controllers can be designed more-or-less separately from other systems in the vehicle.

- In-vehicle level control functionality (indicated by $\mathcal{K}_s$) covers larger hardware and software parts of a vehicle. Typical examples are complex embedded systems, developed by specialized suppliers. The control functionality is often distributed, using in-vehicle networks to interconnect all systems and functionalities, although some direct, local connections are also present (see Figure 6.1). Interaction with other in-vehicle systems has to be taken into account.

- At the full-vehicle level, focus is on determining desired vehicle behavior for the vehicle as a whole, for a specific traffic situation and a specific vehicle state (indicated by $\mathcal{K}_v$ in Figure 6.1). Actual control of the vehicle dynamics to facilitate this desired behavior is assumed to be taken care of by in-vehicle level control functions. Facilitated by in-vehicle networks, full-vehicle level control systems provide setpoints for distributed in-vehicle level systems. Typically, idealized vehicle dynamics are considered.

The driver fulfills both full-vehicle and in-vehicle level control functions (see Figure 6.1). Given a traffic situation and the vehicle state, the driver determines desired behavior for the vehicle as a whole. At the full-vehicle level, the driver translates this into setpoints for in-vehicle level controllers, for example, a desired cruise control velocity. At the in-vehicle level, the driver directly controls the dynamics of the vehicle via, for example, the steering wheel, the brake pedal and the throttle pedal.

The classification facilitates characterizing automotive control problems on the basis of the corresponding modeling and the specification of performance requirements. The

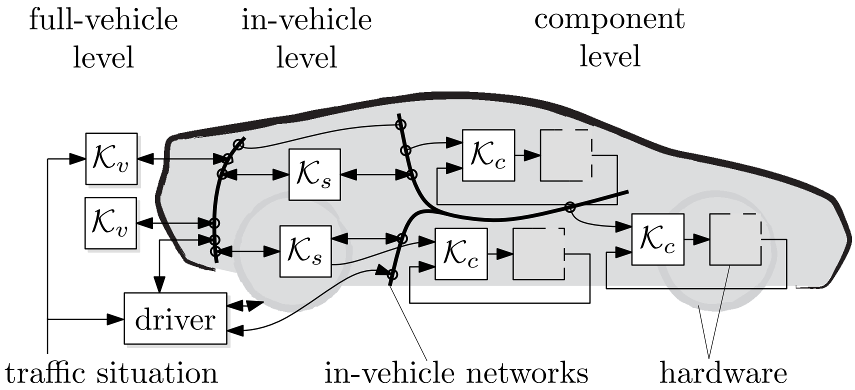


Figure 6.1: Schematic representation of the classification of automotive control problems proposed in Table 6.1, where $\mathcal{K}_c$ represent component level controllers, $\mathcal{K}_s$ are in-vehicle level controllers, and $\mathcal{K}_v$ are full-vehicle level controllers. The vehicle hardware, consisting of mechanics, hydraulics and electronics, is indicated in grey. All systems are interconnected by in-vehicle networks (the heavy black lines) and direct connections (the thin black lines).

case studies that are considered in this thesis focus on control functionality at the full-vehicle level and at the in-vehicle level. Component level examples are not considered. Accordingly, given the classification, the specification of performance requirements and the required modeling are discussed in the next sections.

6.3 Performance requirements

Performance requirements typically follow from the setpoints of higher-level control systems: at the full-vehicle level, desired vehicle behavior is translated into setpoints for system level controllers, which, in their turn, determine setpoints for component level systems. For each level, the specification of performance requirements is discussed. The case studies are used to illustrate the results (see Table 6.2).

6.3.1 Full-vehicle level

At the full-vehicle level, focus is on the specification of desired vehicle behavior. Typical challenges in the specification of corresponding performance requirements, for automotive functionality at a full-vehicle level are:

- translating global performance qualifiers into control-oriented, quantitative performance requirements and setpoints for control problems at the in-vehicle level,

Table 6.2: Overview of the case studies and the corresponding performance requirements.

case	global performance qualifiers	control-oriented requirements
I. clutch	comfort	high bandwidth, small steady-state error, robust stability
II. CC	constant performance over the operating range	high bandwidth, small overshoot, small steady-state error
III. ACC	safety and comfort	easily tunable, operating-point dependent constraints
IV. CACC	traffic throughput	string-stability behavior

- and taking into account the situation and driver dependency of global performance qualifiers.

Desired vehicle behavior is often defined via so-called global performance qualifiers, which are typically situation and driver dependent (see Section 1.1.3). Two examples are presented in the design of an adaptive cruise controller (ACC) (see Chapter 4) and in the design of a cooperative adaptive cruise controller (CACC) (see Chapter 5).

In the ACC case, the global performance qualifiers safety and comfort are translated into operating-point dependent constraints. The operating points are determined by the traffic situation and the vehicle state. In many situations, the performance qualifiers are conflicting, demanding the possibility for a personalized weighting or tuning. To facilitate a driver-dependent tuning of the controller, the global performance qualifiers are used to parameterize the design of the ACC. The tuning of the resulting ACC is directly related to the original performance qualifiers.

For the CACC design, the global performance qualifier traffic throughput is translated into a condition for string stability. Focus has been on a decentralized controller design, although throughput considers a traffic flow rather than individual vehicle behavior. Accordingly, a sufficient condition for string stability of the total traffic flow is adopted, which can be used in a decentralized design. The sufficient condition comes down to a minimal inter-vehicle spacing that is required to achieve string-stable driving behavior. As a result, situation and driver dependencies can easily be included. In the case that no wireless communication is present, the minimal inter-vehicle spacing increases. Furthermore, defining the minimal inter-vehicle spacing as a lower limit allows a driver-dependent setting.

6.3.2 In-vehicle level

Focus of control functionality at the in-vehicle level is on controlling the complex vehicle dynamics, facilitating the desired vehicle behavior as defined at the full-vehicle level. For the specification of performance requirements for automotive functionality at an in-vehicle level, typical points of attention are:

- large variations in operating conditions have to be taken into account (see Section 1.1.3),
- and it is important to focus on control-oriented performance requirements, as opposed to specification of desired setpoints and reference trajectories.

If variations in the operating conditions can be measured, they can be taken into account explicitly in the controller design. Correspondingly, either constant or operating-condition dependent performance requirements can be defined. The former approach is adopted in the case of the cruise control design (see Chapter 3). The resulting closed-loop system dynamics are constant over the operating range, facilitating intuitive controller design and setpoint generation at the higher full-vehicle level. The latter approach, using operating-condition dependent performance requirements, may reduce possible conservativeness in the performance requirements. Consider, for example, again the cruise control design. The desired performance of the setpoint tracking in the case of an empty vehicle is possibly conservative, whereas for a fully loaded vehicle, the desired performance is unrealistic and, hence, not achieved (see Section 3.5). Using loading-dependent performance requirements would facilitate an increase in performance for an empty vehicle and more realistic performance requirements for a fully loaded vehicle.

In the case that variations in the operating conditions cannot be measured, an often more conservative controller design has to be adopted. Accordingly, conservative performance requirements are defined, accounting for the worst-case operating conditions. As a result, the closed-loop performance of the system will vary similar to the (unknown) variations in the operating conditions. Consider, for example, the design of the drive-off controller for the clutch system in Chapter 2. As the effect of wear cannot be measured, a conservative, robust controller design is adopted. The resulting closed-loop dynamics are robustly stable, however, closed-loop performance of the system may vary.

Besides variations in the operating conditions, the vertical supply chain that is typical for the automotive industry can significantly influence the specification of performance requirements at the in-vehicle level. In-vehicle systems are typically designed by specialized suppliers, whereas the corresponding performance requirements are determined by the original equipment manufacturers (OEMs) (see Section 1.1.2). As a result of this transition in responsibilities, evaluation and possible re-specification of performance requirements is time consuming.

For example, in practice, performance requirements are often mixed with desired setpoints and reference trajectories. As an example, consider the step input reference signal that is used in the cruise control design (see Chapter 3). Exact tracking of this reference signal is not possible. Consequently, the desired trajectory for the vehicle velocity is determined by the closed-loop performance requirements, rather than using a corresponding setpoint generator. In practice, this is often solved via implementation of many tuning knobs, resulting in time-consuming tuning and calibration afterwards. For example, nonlinear elements in the desired trajectories or setpoints are easily incorporated via an appropriate setpoint generator, whereas this is difficult, if not impossible, using commonly adopted linear controller designs and corresponding performance requirements. Hence, the specification of appropriate, control-oriented performance requirements, as opposed to the specification of desired setpoints and reference trajectories, is important to prevent time-consuming online tuning and calibration afterwards.

6.3.3 Component level

Typical examples of component level systems are electro-mechanical and electro-hydraulic actuators. Often, these systems are developed stand-alone, being used later on as part of larger embedded systems. Consequently, generic performance specifications facilitate a large range of applications, and it can be assumed that the main objective of performance requirements at the component level is to enable high-performance control design at the in-vehicle level. Hence, the following guidelines result:

- a high closed-loop bandwidth (with respect to the closed-loop bandwidths at the corresponding in-vehicle level),
- and linear closed-loop dynamics, i.e., similar closed-loop performance over the total operating range.

Performance of in-vehicle level controllers is limited if the closed-loop bandwidth of a corresponding component level system is limited. For example, for the clutch controller design, the desired drive-off performance is not achieved (see Chapter 2). This can, at least partly, be related to the limited bandwidth of the clutch actuator system. The clutch actuator system is a typical example of a component level system. It is part of a larger system, being the automated manual transmission. As a result of IP issues, this is a more-or-less black-box system to the OEMs. Consequently, both identification and solution of this problem are difficult, requiring close cooperation between the corresponding supplier and the OEM.

Ensuring, at the component level, a high closed-loop bandwidth and constant closed-loop performance over the total operating range, decreases the identification and modeling effort that is required at the in-vehicle level. For example, consider the case of the design of a cruise controller (see Chapter 3). The engine is considered as the actuator providing the vehicle propulsion. Assuming that the closed-loop system dynamics have high bandwidth and exhibit constant closed-loop performance over the total operating range, facilitates easy modeling at the in-vehicle level.

Often, component level functionality is used as part of a larger embedded system. Hence, the incentive for suppliers of component level functionality to adopt such performance requirements is a wide range of application. However, more stringent performance requirements generally cost more. In practice, a deliberate choice has to be made between stringent performance requirements and costs.

6.4 Modeling

In this section, the modeling that is required to solve automotive control problems is discussed. An overview of the modeling that is adopted in the case studies is given in Table 6.3. Focus is on the control-oriented modeling that is used as a basis for model-based control design.

Table 6.3: Overview of the case studies and the corresponding modeling that is adopted.

case	classification	model specification
I. clutch	in-vehicle level	robust uncertainty model for wear, linear driveline model (6^{th}-order)
II. CC	in-vehicle level	mass and gear dependent linear parameter varying model of the driveline and vehicle body (10^{th}-order)
III. ACC	full-vehicle level	double integrator (2^{nd}-order)
IV. CACC	full-vehicle level	double integrator, first-order low-pass, and delay (3^{th}-order)

6.4.1 Full-vehicle level

For today's full-vehicle level control functionality, extensive modeling of the vehicle dynamics is often not required:

- focus is on specification of desired vehicle behavior for the vehicle as a whole, as opposed to actual control of its complex vehicle dynamics,
- to manage complexity of the modeling and the corresponding controller design, the hierarchy of the proposed classification can often be exploited, assuming ideal

behavior of lower level controllers and systems,

- today's advanced driver assistance systems typically focus on comfort, often requiring relatively undemanding, e.g., low-frequent, performance requirements,

- and, with respect to the dynamics of other traffic, focus is on vehicle-specific, decentralized control functionality, implying that the dynamics of other traffic can be regarded as external disturbances.

The modeling at a lower in-vehicle level, actually controlling the complex vehicle dynamics, will be rather extensive to validate the second aspect. Furthermore, if, for example, the closed-loop dynamics of lower in-vehicle level systems cannot be regarded as ideal, more extensive modeling is required. Moreover, anticipating increasingly stringent performance demands, in future, the third aspect may not hold anymore. For example, focusing on safety instead of comfort, the performance requirements become more demanding, and, accordingly, more extensive modeling will be required.

Advanced driver assistance systems are a typical example of automotive control systems at the full-vehicle level. Based on the traffic situation and the actual vehicle state, desired behavior is determined for the vehicle as a whole. Innovative sensor technology is adopted to measure the behavior of surrounding traffic. Examples are the adaptive cruise control design in Chapter 4, using a radar or lidar, and the cooperative adaptive cruise control (CACC) design in Chapter 5, combining radar or lidar measurements with wireless inter-vehicle communication.

Table 6.3 illustrates that the required modeling in the case of the ACC and CACC designs is indeed limited. In the ACC case, the vehicle is modeled using a double integrator (see Chapter 4). In the CACC case, a low-pass and a delay are added to this double integrator, accounting for the limited bandwidth and actuator delay of lower level systems (see Chapter 5).

6.4.2 In-vehicle level

Automotive control problems at the in-vehicle level target control of the complex vehicle dynamics. As a result, corresponding modeling is often relatively extensive, which is exemplified by the case studies (see Table 6.3). Typical issues for the corresponding modeling are:

- a large range of variable operating conditions,

- interaction with other in-vehicle level systems,

- and restrictions imposed by IP issues, which are a result of the vertical automotive supply chain.

Typically, a large range of variable operating conditions has to be considered (see Section 1.1.3). The required modeling of these variations depends on the corresponding performance requirements and the possibility to measure the variations. For example, in the cruise control design in Chapter 3, variations in the loading and the gear are measured online. To arrive at similar closed-loop performance over the total operating range, a set of loading and gear ratio-dependent linear parameter varying (LPV) models is derived. These models form the basis for corresponding gain-scheduling and LPV controller designs. In the design of a robustly stable clutch controller in Chapter 2, variations due to wear are not measured. Consequently, they are modeled as uncertainties and a corresponding robust controller is designed to stabilize the system in all operating conditions.

The vertical supply chain complicates identification and the corresponding modeling for both the suppliers and the OEMs (see Section 1.1.2). The design is typically done by specialized suppliers, whereas integration is done by the OEMs. As a result, the OEMs often have to deal with more-or-less black-box systems. Consider, for example, an engine idle velocity controller. Commonly, the idle velocity is controlled by a dedicated idle velocity governor. However, in a vehicle with an automated manual transmission (AMT), the AMT takes over control of the engine idle velocity when closing the clutch. The corresponding relevant dynamics are typically unknown to the supplier of the AMT, while the AMT is a black-box system for the OEM. This complicates identification and modeling of the relevant dynamics, as well as integration of the system.

Furthermore, integration of separately-designed systems often introduces (undesired) interactions. As a result, performance of individual systems will most likely be suboptimal. Accounting for interactions in the modeling, and subsequently, in the controller synthesis, enables to anticipate these interactions, for example via disturbance models.

6.4.3 Component level

The relevant dynamics of component level systems can be considered as more-or-less stand-alone. Systems at the component level are often supplied as part of a larger embedded system. Hence, problems resulting from the vertical supply chain as discussed in the previous section, are less critical for the component level. As a result, standard approaches can often be adopted for identification and modeling of the relevant dynamics. In this research, no further example of the design of component level functionality is considered.

6.5 Controller synthesis methods

The results of the case studies demonstrate the effectiveness of several controller synthesis methods to handle the typical challenges in automotive control problems according to Table 1.2, being global performance qualifiers and variable operating conditions. An overview of the controller synthesis methods that are adopted in the case studies is given in Table 6.4.

Table 6.4: Overview of the case studies, listing the controller synthesis methods used.

case	classification	controller synthesis method
I. clutch	in-vehicle level	robust control
II. CC	in-vehicle level	gain scheduling and linear parameter varying control
III. ACC	full-vehicle level	model predictive control
IV. CACC	full-vehicle level	feedforward control

In the clutch controller design, a robust controller synthesis method is adopted to account for unmeasured variations in the operating conditions, in this case wear of the clutch facings material (see Chapter 2). In the case of the cruise controller design, gain scheduling and linear parameter varying controller synthesis methods are adopted, explicitly taking into account variations in the vehicle loading and the actual gear (see Chapter 3). The resulting controller enables a constant closed-loop performance over the total operating range. Finally, an MPC synthesis is adopted in the case of the adaptive cruise control design. The MPC synthesis provides a suitable approach to take into account weighted performance requirements and constraints that are a result of the global performance qualifiers safety and comfort (see Chapter 4). Furthermore, implementing the MPC in a receding horizon manner facilitates to mimic situation dependent driver behavior.

Application of state-of-the-art controller synthesis methods yields possibilities for performance improvements, while decreasing tuning effort. Using appropriate controller synthesis methods, the controller is an automatic result, decreasing both the number of manual design steps in the controller synthesis and the amount of tuning. However, the increasing theoretical complexity of controller synthesis methods often results in an increased computational burden, while insight in the resulting controller decreases. Considering application in practice, some fine-tuning afterwards is often desirable, which is hindered by increased computation effort and decreased insight in the controller.

For example, in the case of the cruise controller design, different gain scheduling and LPV controller synthesis methods are compared (see Table 3.2). For a synthesis method with many design steps, no a-priori guarantees regarding closed-loop stability and performance can be given. Furthermore, re-tuning of the controller takes much time due to the number of design steps. However, insight in the resulting controller is high. For a completely automated synthesis method, full a-priori guarantees can be given. However, insight in the resulting controller is limited, and the computational burden to synthesize the controller prohibits online fine-tuning. Moreover, the a-priori guarantees hold for the model rather than the actual system.

Hence, in practice, a deliberate trade-off has to be made between methods that incorporates many design steps, and methods that provide a completely automated synthesis. To facilitate the application of state-of-the-art controller synthesis methods, research should focus on the development of efficient and fast algorithms to keep the corresponding computational burden down. Furthermore, research should focus on the development of tools, specifically focusing on the application of available controller synthesis methods in practice. Dedicated tools may contribute to increased insight in both the controller synthesis and the resulting controller, facilitating application in practice.

CHAPTER 7

Conclusions and recommendations

Abstract - In this chapter, the main conclusions of this thesis are summarized. Furthermore, recommendations on model-based control design in the automotive industry are given.

7.1 Conclusions

The application of a systematic, model-based control design approach for four automotive control cases is presented in the Chapters 2 to 5. The results of these case studies demonstrate the possibilities and opportunities for the application of a systematic, model-based control design approach, as opposed to the usual approach of pragmatic, online tuning and calibration techniques: better performance can be achieved, a-priori stability and performance guarantees can be given and both the automated controller synthesis and the systematic approach reduce tuning effort.

Furthermore, the case studies show that controller synthesis methods are available that are specifically suitable to handle the typical challenges of variable operating conditions and global performance qualifiers (see Table 1.2, pg. 10). In this research, robust control, model predictive control (MPC), and gain scheduling (GS) or linear parameter varying control (LPV) are considered. Finally, in all cases, a practical implementation is presented, using test vehicles and a hardware-in-the-loop (HIL) setup. The results validate the practical applicability of the controller synthesis methods used.

- Case I: The effect of clutch judder can be modeled as an uncertainty that destabilizes the driveline dynamics. The model facilitates the design of a robust controller, stabilizing the dynamics during drive-off maneuvers. Judder-induced oscillations in the driveline are resolved, preventing re-tuning of the controller if the dynamics change due to wear (see Chapter 2).

- Case II: A cruise control with constant behavior over the operating range is achieved by explicitly taking into account large variations in both the gear ratio and the vehicle loading in the modeling and the subsequent controller synthesis. Comparison of classical GS and modern LPV controller syntheses indicates that semi-automated methods are the most suitable for application in practice (see Chapter 3).
- Case III: The design of a parameterized adaptive cruise control (ACC) is proposed, providing a systematic approach to translate global performance qualifiers (or key characteristics) into intuitive design parameters, thus promising a significant reduction in tuning effort (see Chapter 4).
- Case IV: The design of a cooperative ACC focusing on the feasibility of implementation is presented. Assessing string stability of this design requires evaluating oscillations in the velocity and acceleration of the vehicles in a platoon, as opposed to the tracking error. Compared to standard ACC functionality, string stability can be guaranteed for a velocity-dependent inter-vehicle spacing, while improving traffic throughput (see Chapter 5).

The case studies consider representative control problems in the automotive industry, in the sense that the typical challenges according to Table 1.2 are addressed, being variable operating conditions and global performance qualifiers. Insight in the properties of control problems in the automotive industry is obtained via the case studies. Based on this insight, a classification is derived, facilitating characterization of automotive control problems on the basis of the required modeling and the specification of performance requirements (see Chapter 6).

Problems at i) a full-vehicle level, ii) an in-vehicle level, and iii) a component level are distinguished. In this research, in particular the former two are considered.

- At the full-vehicle level, focus is on the specification of desired vehicle behavior for the vehicle as a whole, rather than actually controlling the complex vehicle dynamics. Typically, the required level of detail of modeling the vehicle dynamics is limited, and global performance qualifiers have to be translated into control-oriented performance requirements and setpoints that can be used at the in-vehicle level.
- At the in-vehicle level, the modeling and the specification of performance requirements are typically influenced by a wide variety of operating conditions. Selection and modeling of the variations differs per situation and requires engineering insight and experience. Both for unmeasured and measured variations dedicated model-based controller synthesis methods are available.

The results of the case studies demonstrate the practical applicability of the controller synthesis methods used. However, the results also indicate that the theoretical complexity of the controller synthesis methods can be limiting. As a result of the theoretical complexity, typically, the computational burden increases, while insight in the resulting controller decreases, complicating online fine-tuning of the controller. In practice, a choice has to be made in particular between the number of non-automated and automated design steps in the methods. In general, the number of a-priori guarantees increases for a decreasing number of non-automated design steps, while the complexity of the synthesis increases simultaneously. See, for example, the discussion on the use of classical GS techniques or the use of more recent LPV methods in Section 3.6.

7.2 Recommendations

In the introduction of this research, three typical challenges in automotive control problems are identified (see Table 1.2). The research has focused in particular on two of them, being variable operating conditions and global performance qualifiers. First, research should focus on methods and tools to cope with the third challenge, i.e., the problems that are induced by the vertical supply chain. For example, intellectual property issues hinder modeling and the specification of performance requirements. Furthermore, integration of separately-designed, more-or-less black-box systems will inevitably introduce undesired interactions, most likely limiting performance (Gordon et al., 2003).

Second, it is recommended to detail the challenges in Table 1.2 further. For example, one could think of the identification of typical disturbance forces or the limitations imposed by implementation of controllers on an actual ECU. Accordingly, the discussion in Chapter 6 can be detailed further, and more specific guidelines can be given on the required modeling and the specification of corresponding performance requirements.

To improve and facilitate implementation of state-of-the-art controller synthesis methods in practice, research should focus on efficient and fast synthesis algorithms. The theoretical complexity of state-of-the-art controller synthesis methods often translates into a high computational burden, in practice hindering (online) fine-tuning of the controller. In Chapters 2 through 4, complexity of the modeling and the variety of performance requirements is deliberately limited to keep the computational burden reasonably small.

Simultaneously, insight in the controller synthesis and the resulting controller structure is often decreased by the theoretical complexity of controller synthesis methods. The development of tools specifically focusing on the application of available controller synthesis methods in practice, may contribute to increased insight in both the controller synthesis and the resulting controller, facilitating application in practice.

Furthermore, only a specific set of controller synthesis methods is addressed in this research. The methods are selected as particularly suitable to handle the typical challenges in automotive control problems according to Table 1.2. It is recommended to address the specific advantages and disadvantages of a larger set of controller synthesis methods.

Finally, industry should take the lead in actual implementation of systematic, model-based control design approaches in practice. In many sectors, today's market demands for new functionality are readily resolved using electronics and control software, the automotive industry being only one example. Consequently, the possibilities for application of a systematic, model-based control design approach are numerous.

Moreover, the automotive industry could fulfill a leading role, being one of the largest investors in research and development, investing almost €85 billion per year. The annual turnover is equivalent to the 6th largest economy in the world, or 10% of the world's gross domestic product (OICA, 2006). As a result, the automotive industry plays a key role in the technology level of other industries and of the society. Consequently, the automotive industry is an ideal industry to fulfill a leading role in demonstrating the use of available tools and methods adopting a systematic, model-based control design approach.

References

Ahmed, M., Saraydar, C. U., ElBatt, T., Yin, J., Talty, T., and Ames, M. (2007). Intra-vehicular wireless networks. In *IEEE Globecom Workshops*, Washington, WA, USA, Nov., 2007. p. 1–9.

Alfieri, E., Amstutz, A., and Guzzella, L. (2009). Gain-scheduled model-based feedback control of the air / fuel ratio in diesel engines. *Control Eng. Pract.*, 17(12), p. 1417–1425.

Apkarian, P. and Gahinet, P. (1995). A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controllers. *IEEE Trans. Autom. Control*, 40(5), p. 853–864.

Apkarian, P., Gahinet, P., and Becker, G. (1995). Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems: a design example. *Automatica*, 31(9), p. 1251–1261.

Arem, B. van, Driel, C. J. G. van, and Visser, R. (2006). The impact of co-operative adaptive cruise control on traffic flow characteristics. *IEEE Trans. Intell. Transp. Syst.*, 7(4), p. 429–436.

Barooah, P. and Hespanha, J. P. (2005). Error amplification and disturbance propagation in vehicle strings with decentralized linear control. In *Proc. 44th IEEE Conf. Decis. Control*, Dec., 2005. p. 4964–4969.

Baslamisli, S. Ç., Köse, I. E., and Anlaş, G. (2007). Robust control of anti-lock brake system. *Veh. Syst. Dyn.*, 45(3), p. 217–232.

Baumann, J., Swarnakar, A., and Kiencke, U. (2005). A robust controller design for anti-jerking. *SAE Techn. Pap. Ser.*, (2005-01-0041).

Becker, G. and Packard, A. (1994). Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback. *Syst. Control Lett.*, 23(23), p. 205–215.

Beenakkers, M. A. (2007). *Drive-off control to prevent clutch judder*. Master's thesis, Eindhoven University of Technology, Eindhoven, the Netherlands. DCT 2007-138.

Bemporad, A., Heemels, W. P. M. H., and De Schutter, B. (2002a). On hybrid systems and closed-loop MPC systems. *IEEE Trans. Autom. Control*, 47(5), p. 863–869.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. N. (2002b). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), p. 3–20.

Bengtsson, J., Strandh, P., Johansson, R., ål, P. Tunest, and Johansson, B. (2006). Multi-output control of a heavy duty HCCI engine using variable valve actuation and model predictive control. *SAE Techn. Pap. Ser.*

Bleek, R. A. P. M. van den (2007). *Design of a hybrid adaptive cruise control stop-&-go system*. Master's thesis, Eindhoven University of Technology, Eindhoven, the Netherlands. DCT 2007.125.

Bode, H. W. (1945). *Network analysis and feedback amplifier design.* Van Nostrand Co., New York.

Borelli, F., Falcone, P., Keviczky, T., Asgari, J., and Hrovat, D. (2005). MPC-based approach to active steering for autonomous vehicle systems. *Int. J. Veh. Auton. Syst.*, 3(2–4), p. 265–291.

Bosch, R. GmBH (2007). *Automotive electrics, automotive electronics.* Chichester: Wiley, 5th edition.

Bostwick, C. C. and Szadkowski, A. (1998). Self-excited vibrations during engagements of dry friction clutches. *SAE Techn. Pap. Ser.*, 982846.

Bristol, E. H. (1966). On a new measure of interaction for mutivariable process control. *IEEE Trans. Autom. Control*, AC-11(1), p. 133–134.

Broy, M. (2006). Challenges in automotive software engineering. In *Proc. 28th Int. Conf. Softw. Eng.*, Shanghai, China, May, 2006. p. 33–42.

Bruce, M. (2004). Powertrain modeling and experimental validation in a heavy duty vehicle. In *7th Int. Symp. Adv. Veh. Control*, Arnhem, Netherlands, Aug., 2004. p. 71–76.

Bruce, M., Egardt, B. S., and Pettersson, S. (2005). On powertrain oscillation damping using feedforward and LQ feedback control. In *Proc. IEEE Conf. Control Appl.*, Toronto, Canada, Aug., 2005. p. 147–156. 21–25.

Buckland, J. H. and Cook, J. A. (2005). Automotive emissions control. In *Proc. Am. Control Conf.*, Portland, OR, USA, Jun., 2005. p. 3290–3295.

Cassius, I. and Kun, A. L. (2007). Vehicle telematics: a literature review. Technical Report ECE.P54.2007.9.

Centea, D. N., Rahnejat, H., and Menday, M. T. (1999). The influence of the interface coefficient of friction upon the propensity to judder in automotive clutches. *Proc. Inst. Mech. Eng.*, 213(D), p. 245–258.

Chan, C. C. (2007). The state of the art of electric, hybrid, and fuel cell vehicles. *Proc. IEEE*, 95(4), p. 704–718.

Chang, S. and Gordon, T. J. (2008). A flexible hierarchical model-based control methodology for vehicle active safety systems. *Veh. Syst. Dyn.*, 46(1), p. 63–75.

Chang, Y. J. and Rasmussen, B. P. (2008). Stable controller interpolation for LPV systems. In *Proc. Am. Control Conf.*, Seattle, WA, USA, Jun., 2008. p. 3082–3087.

Chen, J. S. (1997). Active control of torsional vibrations for vehicle drive trains. In *Proc. Am. Control Conf.*, Albuquerque, New Mexico, Jun., 1997. p. 1152–1156. 4–6.

Claveau, F., Chevrel, P., and Knittel, K. (2007). A 2DOF gain-scheduled controller design methodology for a multi-motor web transport system. *Control Eng. Pract.*, 16(5), p. 609–622.

Coelingh, E., Chaumette, P., and Andersson, M. (2002). Open-interface definitions for automotive systems - application to a brake by wire system. *SAE Techn. Pap. Ser.*, (2002-01-0267).

Cook, J. A., Sun, J., Buckland, J. H., Kolmanovsky, I. V., Peng, H., and Grizzle, J. W. (2006). Automotive powertrain control - a survey. *Asian J. Control*, 8(3), p. 237–260.

Corona, D. and De Schutter, B. (2008). Adaptive cruise control for a SMART car: a comparison benchmark for MPC-PWA control methods. *IEEE Trans. Control Syst. Technol.*, 16(2), p. 365–372.

Crowther, A., Zhang, N., Liu, D. K., and Jeyakumaran, J. K. (2004). Analysis and simulation of clutch engagement judder and stick-slip in automotive powertrain systems. *Proc. Inst. Mech. Eng.*, 218, p. 1427–1446.

De Caigny, J., Camino, J. F., and Swevers, J. (2009). Interpolating model identification for SISO linear parameter-varying systems. *Mech. Syst. Signal Process.*, 23, p. 2395–2417.

De Caigny, J., Camino, J. F., Oliveira, R. C. L. F., Peres, P. L. D., and Swevers, J. (2010). Gain-scheduled $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control of discrete-time polytopic time-varying systems. *IET Control Theory Appl.*, 4(3), p. 362–280.

Dettori, M. and Scherer, C. W. (2002). MIMO control design for a compact disc player with multiple norm specifications. *IEEE Trans. Control Syst. Technol.*, 10(5), p. 635–645.

Di Cairano, S., Yanakiev, D., Bemporad, A., Kolmanovsky, I. V., and Hrovat, D. (2008). An MPC design flow for automotive control and applications to idle speed regulation. In *Proc. 47th IEEE Conf. Decis. Control*, Cancun, Mexico, Dec., 2008. p. 5686–5691.

Diepen, K. van (2009). *Design of an integrated vehicle speed controller*. Master's thesis, Eindhoven University of Technology, Eindhoven, The Netherlands. DCT 2009.094.

Dinh, M., Scorletti, G., Fromion, V., and Magarotto, E. (2005). Parameter dependent $\mathcal{H}_\infty$ control by finite dimensional LMI optimization: application to trade-off dependent control. *Int. J. Robust Nonlin. Control*, 15(9), p. 383–406.

Dorey, R. E., McLaggan, J. D., Harris, J. M. S., Clarke, D. P., and Gondre, B. A. C. (2001). Transient calibration on the TestBed for emissions and driveability. *SAE Techn. Pap. Ser.*, (2001-01-0215).

Doyle, J. C. (1984). Lecture notes on advances in multivariable control. In *ONR / Honeywell Workshop*, Minneapolis, USA, 1984.

Driel, C. J. G. van, Hoedemaeker, M., and Arem, B. van (2007). Impacts of a congestion assistant on driving behaviour and acceptance using a driving simulator. *J. Transp. Res.*, F 10, p. 139–152.

Flexray (2002). FlexRay - the communication system for advanced automotive control applications. http://www.flexray.com, 2002.

Fröberg, J., Ækerholm, M., Sandström, K., and Norström, C. (2007). Key factors for achieving project success in integration of automotive mechatronics. *Innov. Syst. Softw. Eng.*, 3(2), p. 141–155.

Gaspar, P., Szaszi, I., and Bokor, J. (2003). Design of robust controllers for active vehicle suspension using the mixed μ-synthesis. *Veh. Syst. Dyn.*, 40(4), p. 193–228.

Gianelli, M. and Primbs, J. A. (2000). An analysis for optimization based control applied to quasi-LPV plants. In *Proc. Am. Control Conf.*, Chicago, IL, USA, Jun., 2000. p. 1909–1913.

Gietelink, O. J. (2007). *Design and validation of advanced driver assistance systems*. Ph.D. thesis, Delft University of Technology, Deflt, the Netherlands.

Gordon, T., Howell, M., and Brandao, F. (2003). Integrated control methodologies for road vehicles. *Veh. Syst. Dyn.*, 40(1), p. 157–190.

Grimm, K. (2003). Software technology in an automotive company: major challenges. In *Proc. 25th Int. Conf. Softw. Eng.*, Portland, OR, USA, 2003. p. 498–503.

Groot Wassink, M. B., Wal, M. van de, Scherer, C. W., and Bosgra, O. (2004). LPV control for a wafer stage: beyond the theoretical solution. *Control Eng. Pract.*, 13, p. 231–245.

Guerra, T. M. and Vermeiren, L. (2001). Control laws for Takagi-Sugeno fuzzy models. *Fuzzy Sets Syst.*, 120, p. 95–108.

Güvenç, B. A., Güvenç, L., and Karaman, S. (2009). Robust yaw stability controller design and hardware-in-the-loop testing for a road vehicle. *IEEE Trans. Veh. Technol.*, 58(2), p. 555–571.

Guzzella, L. (2009). Automobiles of the future and the role of automatic control in those systems. *Annu. Rev. Control*, 33(1), p. 1–10.

Heckmann, A. (2006). A brake model with thermoelastic disc for the analysis of vehicle judder vibrations. *Veh. Syst. Dyn.*, 44(1), p. 360–367.

Heinecke, H., Schnelle, K.-P., Fennel, H., Bortolazzi, J., Lundh, L., Leflour, J., Maté, J.-L., Nishikawa, K., and Scharnhorst, T. (2004). AUTomotive Open System ARchitecture - an industry-wide initiative to manage the complexity of emerging automotive E/E-architectures. *SAE Techn. Pap. Ser.*, (2004-21-0042).

Helmersson, A. (1996). Parameter dependent Lyapunov functions based on linear fractional transformation. In *Proc. IFAC World Congr.*, 1996.

Hingwe, P., Tan, H.-S., Packard, A. K., and Tomizuka, M. (2002). Linear parameter varying controller for automated lane guidance: Experimental study on tractor-trailers. *IEEE Trans. Control Syst. Technol.*, 10(6), p. 793–806.

Hofman, T., Van Druten, R. M., Serrarens, A. F. A., and Steinbuch, M. (2007). Rule-based energy management strategies for hybrid vehicles. *Int. J. Electric Hybrid Veh.*, 1, p. 71–94.

Hovda, M. and Skogestad, S. (1994). Sequential design of decentralized controllers. *Automatica*, 30(10), p. 1601–1607.

Hu, J., Unbehauen, H., and Bohn, C. (1996). A practical approach to selecting weighting functions for $\mathcal{H}_\infty$ control and its application to a pilot plant. In *UKACC Int. Conf. Control*, Bochum, Germany, 1996, 2, p. 998–1003.

Huisman, R. G. M. and Veldpaus, B. (2005). Truck Simulation Models for SIL and HIL to verify Vehicle Stability Control Systems. *JSAE Rev. Automot. Eng.*, 26(1), p. 99–105.

Huppe, X., Lafontaine, J. de, Beauregard, M., and Michaud, F. (2003). Guidance and control of a platoon of vehicles adapted to changing environment conditions. *IEEE Int. Conf. Syst. Man Cyb.*, 4, p. 3091–3096.

IEEE Computer Society (2003). IEEE standard 802.11g (TM). The Institute of Electrical and Electronics Engineers, Inc., 3 Park Avenue, New York, NY 10016-5997, USA, Jun., 2003.

International Organization for Standardization (2002). Transport information and control systems - adaptive cruise control systems - performance requirements and test procedures. *NEN-ISO 15622*.

International Organization for Standardization (1997). Mechanical vibration and shock - evaluation of human exposure to whole-body vibration. *NEN-ISO 2631*.

Inagaki, H., Kato, H., and Sebe, N. (2007). Development of motor control for hybrid vehicle using gain-scheduled $\mathcal{H}_\infty$ control. *SAE Techn. Pap. Ser.*, (2007-01-0269).

Ioannou, P. A. and Chien, C. C. (1993). Autonomous intelligent cruise control. *IEEE Trans. Veh. Technol.*, 42(4), p. 657–672.

Iwadare, M., Ueno, M., and Adachi, S. (2007). Multi-variable air-path management for a clean diesel engine using model predictive control. *SAE Int. J. Engines*, 2(1), p. 764–773.

Iwasaki, T. and Hara, S. (2005). Generalized KYP lemma: unified frequency domain inequalities with design applications. *IEEE Trans. Automat. Control*, 50(1), p. 41–59.

Johansson, R. and Rantzer, A. (2003). *Nonlinear and hybrid systems in automotive control.* SAE Warrendale.

Kandasamy, N., Hayes, J. P., and Murray, B. T. (2005). Dependable communication synthesis for distributed embedded systems. *Reliab. Eng. Syst. Saf.*, 89(1), p. 81–92.

Keulen, T. A. C. van, Jager, A. G. de, Serrarens, A., and Steinbuch, M. (2009a). Optimal energy management in hybrid electric trucks using route information. *Oil Gas Sci. Technol. - Rev. IFP*.

Keulen, T. A. C. van, Naus, G. J. L., Jager, A. G. de, Molengraft, M. J. G. van de, Steinbuch, M., and Aneke, N. P. I. (2009b). Predictive cruise control in hybrid electric vehicles. *WEVA J.*, 3(ISSN 2032–6653).

Keulen, T. A. C. van, Naus, G. J. L., Jager, A. G. de, Molengraft, M. J. G. van de, Steinbuch, M., and Aneke, N. P. I. (2009c). Predictive cruise control in hybrid electric vehicles. In *EVS24 Int. battery, hybrid fuel cell electr. veh. symp.*, Stavanger, Norway, May, 2009c.

Khalil, H. K. (2002). *Nonlinear systems*. Prentice Hall, Inc., Upper Saddle River, NJ, USA, 3rd edition.

Khatir, M. E. and Davison, E. J. (2004). Decentralized control of a large platoon of vehicles using non-identical controllers. In *Proc. Am. Control Conf.*, Boston, MA, Jul., 2004. p. 2769–2776.

Kiencke, U. and Nielsen, L. (2005). *Automotive control systems, for engine, driveline, and vehicle*. Springer-Verlag Berlin Heidelberg, 2nd edition.

Kiencke, U., Dais, S., and Litschel, M. (1986). Automotive serial controller area network. *SAE Techn. Pap. Ser.*, (860391), p. 6.

Kluge, T., Allen, J., and Dhaliwal, A. (2009). Advantages and challenges of closed-loop HIL testing for commercial and off-highway vehicles. *SAE Techn. Pap. Ser.*, (2009-01-2841).

Kolmanovsky, I. V. (2008). Developments and research directions in control of automotive systems. Sep., 2008. Eindhoven, The Netherlands.

Kolmanovsky, I. V. and Gilbert, E. G. (1997). Theory and computation of disturbance invariant sets for discrete-time linear systems. *Math. Probl. Eng.*, 4, p. 317–367.

Kolmanovsky, I. V. and Winstead, V. (2006). A receding horizon optimal control approach to active state and parameter estimation in automotive systems. In *Proc. IEEE Int. Conf. Control Appl.*, Munich, Germany, Oct., 2006. p. 2796–2801.

Kvasnica, M., Grieder, P., Baotic, M., and Christophersen, F. J. (2006). *Multi-parametric toolbox (MPT)*.

Kwiatkowski, A., Werner, H., Blath, J. P., Ali, A., and Schultalbers, M. (2009). Linear parameter varying PID controller design for charge control of a spark-ignited engine. *Control Eng. Pract.*, 17(11), p. 1307–1317.

Laan, E. P. van der (2009). *Seat belt control, from modeling to experiment*. Ph.D. thesis, Department of mechanical engineering, Eindhoven University of Technology, Eindhoven, the Netherlands.

Larses, O. (2005). *Architecting and Modeling Automotive Embedded Systems*. Ph.D. thesis, Department of Machine Design, Royal Institute of Technology, Stockholm, Sweden.

Lee, W., Park, S., and Sunwoo, M. (2004). Towards a seamless development process for automotive engine-control system. *Control Eng. Pract.*, 12(8), p. 977–986.

Leen, G. and Heffernan, D. (2002). Expanding Automotive Electronic Systems. *IEEE Comp. Soc. Press*, 35(1), p. 88–93.

Leite, V. J. S. and Peres, P. L. D. (2005). Pole location control design of an active suspension system with uncertain parameters. *Veh. Syst. Dyn.*, 43(8), p. 561–579.

Leith, D. J. and Leithead, W. E. (2000). Survey of gain-scheduling analysis and design. *Int. J. Control*, 73(11), p. 1001–1025.

Levine, W. and Athans, M. (1966). On the optimal error regulation of a string of moving vehicles. *IEEE Trans. Autom. Control*, 11(3), p. 355–361.

Liang, C. Y. and Peng, H. (1999). Adaptive cruise control with guaranteed string stability. *Veh. Syst. Dyn.*, 32(4–5), p. 313–330.

Liang, C. Y. and Peng, H. (2000). String stability analysis of adaptive cruise controlled vehicles. *JSME Int. J. Mechan. Syst., Machine Elements Mfg.*, 43, p. 671–677.

Liu, X. and Mahal, S. S. (2001). Effects of communication delay on string stability in vehicle platoons. In *Proc. IEEE Intell. Transp. Syst. Conf.*, Aug., 2001. p. 625–630.

Löfberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proc. CACSD Conf.*, Taipei, Taiwan, 2004.

Lu, M., Wevers, K., and Heijden, R. van der (2005). Technical feasibility of advanced driver assistance systems (ADAS) for road traffic safety. *Transp. Plan. Technol.*, 28(3), p. 167–187.

Lu, X.-Y. and Hedrick, J. K. (2005). Heavy-duty Vehicle Modeling and Longitudinal Control. *Veh. Syst. Dyn.*, 43(9), p. 653–669.

Lu, X.-Y., Shladover, S., and Hedrick, J. K. (2004). Heavy-Duty Truck Control: Short Inter-Vehicle Distance Following. In *Proc. Am. Control Conf.*, Boston, MA, Jul., 2004, **5**, p. 4722–4727.

Maciejowski, J. M. (2002). *Predictive control with constraints*. Harlow: Prentice Hall - Pearson Education Limited, ISBN: 0-201-39823-0.

Marcos, A. and Balas, G. J. (2004). Development of linear-parameter-varying models for aircraft. *J. Guid., Control, Dyn.*, 27(2), p. 218–228.

Martinez, J.-J. and Canudas-de-Wit, C. (2007). A safe longitudinal control for adaptive cruise control and stop-and-go scenarios. *IEEE Trans. Control Syst. Technol.*, 15(2), p. 246–258.

Masubuchi, I., Ohara, A., and Suda, N. (1995). LMI-based output feedback controller design using a convex parameterization of full-order controllers. In *Proc. Am. Control Conf.*, Seattle, WA, USA, Jun., 1995, **5**, p. 3473–3477.

McCann, R. and Le, A. (2008). Gain scheduling in commercial vehicles with electrohydraulic power steering. *SAE Int. J. Commer. Veh.*, 1(1), p. 481–487.

McIntyre, M. L., Ghotikar, T. J., Vahidi, A., Song, X., and Dawson, D. M. (2009). A two-stage Lyapunov-based estimator for estimation of vehicle mass and road grade. *IEEE Trans. Veh. Technol.*, 58(7), p. 3177–3185.

Meulen, S. H. van der, Jager, A. G. de, Noll, E. van der, Veldpaus, F. E., Sluis, F. van der, and Steinbuch, M. (2009). Improving pushbelt continuously variable transmission efficiency via extremum seeking control. In *18th IEEE Int. Conf. Control Appl.*, St. Petersburg, Russia, Jul., 2009. p. 357–3692.

Miège, A. J. P. and Popov, A. A. (2005). Truck tyre modelling for rolling resistance calculations under a dynamic vertical load. *Proc. IMechE., Part D: J. Automobile Engineering*, 219, p. 441–456.

Mo, C. Y., Beaumont, A. J., and Powell, N. N. (1996). Active control of driveability. *SAE Techn. Pap. Ser.*, (960046).

Möbus, R., Baotić, M., and Morari, M. (2003). Multi-object adaptive cruise control. *Lect. Notes Comput. Sci. Hybrid Syst.: Comput. Control*, 2623, p. 359–374.

Moon, S. and Yi, K. (2008). Human driving data-based design of a vehicle adaptive cruise control algorithm. *Veh. Syst. Dyn.*, 46(8), p. 661–690.

Moon, S., Moon, I., and Yi, K. (2009). Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance. *Control Eng. Prac.*, 27, p. 442–455.

Mössinger, J. (2010). Software in automotive systems. *IEEE Softw.*, 27(2), p. 92–94.

Motor Presse Stuttgart GmbH & Co. KG (2006). Auto, Motor und Sport - Abstandsregeltempomaten: Nah dran. http://www.auto-motor-und-sport.de/tests/technik/hxcms_article_132252_14114.hbs [online], Nov., 2006.

Nagai, M. (2007). The perspectives of research for enhancing active safety based on advanced control technology. *Veh. Syst. Dyn.*, 45(5), p. 413–431.

Naus, G. J. L. (2007a). Evaluation of controller design and tuning at DAF Trucks N.V. Technical report 51051/07-071, DAF Trucks N.V.

Naus, G. J. L. (2007b). Evaluation of controller design and tuning at Integrated Safety, business unit Automotive, TNO Helmond. Technical report, TNO Automotive.

Naus, G. J. L., Ploeg, J., Molengraft, M. J. G. van de, and Steinbuch, M. (2008a). Explicit MPC design and performance-based tuning of an adaptive cruise control stop-&-go. In *Proc. Intell. Veh. Symp.*, Eindhoven, the Netherlands, Jun., 2008a.

Naus, G. J. L., Bleek, R. A. P. M. van den, Ploeg, J., Scheepers, B., Molengraft, M. J. G. van de, and Steinbuch, M. (2008b). Explicit MPC design and performance evaluation of an ACC stop-&-go. In *Proc. Am. Control Conf.*, Seattle, WA, USA, Jun., 2008b. p. 224–229.

Naus, G. J. L., Beenakkers, M. A., Huisman, R. G. M., Molengraft, M. J. G. van de, and Steinbuch, M. (2008c). Robust control to suppress clutch judder. In *Proc. 8th Int. Symp. Adv. Veh. Control*, Kobe, Japan, Oct., 2008c, **I**, p. 140–145.

Naus, G. J. L., Vugts, R. P. A., Ploeg, J., Molengraft, M. J. G. van de, and Steinbuch, M. (2009a). Towards on-the-road implementation of cooperative adaptive cruise control. In *Proc. 16th World Congr. and Exhib. Intell. Transp. Syst. Serv.*, Stockholm, Sweden, Sep., 2009a.

Naus, G. J. L., Vugts, R. P. A., Ploeg, J., Molengraft, M. J. G. van de, and Steinbuch, M. (2009b). CACC string stability tests. http://www.youtube.com/watch?v=aNYh4MjeSpU [online], Sep., 2009b.

Naus, G. J. L., Vugts, R. P. A., Ploeg, J., Molengraft, M. J. G. van de, and Steinbuch, M. (2010a). Cooperative adaptive cruise control, design and experiments. In *Proc. Am. Control Conf.*, Baltimore, MD, USA, Jul., 2010a. p. 6145–6150.

Naus, G. J. L., Ploeg, J., Molengraft, M. J. G. van de, Heemels, W. P. M. H., and Steinbuch, M. (2010b). Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach. *Control Eng. Pract.*, 18(8), p. 882–892.

Naus, G. J. L., Beenakkers, M. A., Huisman, R. G. M., Molengraft, M. J. G. van de, and Steinbuch, M. (2010c). Robust control of a clutch system to prevent judder-induced driveline oscillations. *Veh. Syst. Dyn.* (accepted for publication).

Naus, G. J. L., Vugts, R. P. A., Ploeg, J., Molengraft, M. J. G. van de, and Steinbuch, M. (2010d). String-stable CACC design and experimental validation, a frequency-domain approach. *IEEE Trans. Veh. Technol.* (accepted for publication).

Naus, G. J. L., Ploeg, J., Molengraft, M. J. G. van de, Heemels, W. P. M. H., and Steinbuch, M. (2010e). A model predictive control approach to design a parameterized adaptive cruise control. In L. Del Re, F. Allgöwer, L. Glielmo, C. Guardiola, and I. Kolmanovsky (eds.), *Automotive model predictive control: models, methods and applications*, *Springer Lect. Notes Control Inf. Sci.*, **402**, p. 273–284. Springer Berlin / Heidelberg, 1st. edition.

Navet, N. (2009). In-vehicle networking: a survey and look forward. In *Workshop Spec. Netw.*, Palma, Spain, Sep., 2009.

Navet, N., Song, Y., Simonot-Lion, F., and Wilwert, C. (2005). Trends in automotive communication systems. *Proc. IEEE*, 93(6), p. 1204–1223.

Netherlands Institute for Transport Policy Analysis - KIM (2008). Mobiliteitsbalans 2008, congestie in perspectief (in Dutch). Technical Report, Ministry of Transport, Public Works and Water Management, The Hague, the Netherlands.

Nichols, R. A., Reichert, R. T., and Rugh, W. J. (1993). Gain scheduling for $\mathcal{H}_\infty$ controllers: a flight control example. *IEEE Trans. Control Syst. Technol.*, 1(2), p. 69–79.

Nolte, T., Hansson, H., and Lo Bello, L. (2005). Automotive Communications - Past, Current and Future. *10th IEEE Conf. Emerg. Technol. Fact. Autom.*, 1, p. 985–992.

OICA (2006). The world's automotive industry, some key figures. Technical Report, The International Organisation of Motor Vehicles Manufacturers.

Pacejka, H. B. (2002). *Tyre and vehicle dynamics.* Butterworth-Heinemann, Jordan Hill, Oxford OX2 8DP.

Paijmans, B., Symens, W., Van Brussel, H., and Swevers, J. (2006). A gain-scheduling-control technique for mechatronic systems with position-dependent dynamics. In *Proc. Am. Control Conf.*, Minneapolis, MN, USA, Jun., 2006. p. 2933–2938.

Paijmans, B., Symens, W., Van Brussel, H., and Swevers, J. (2008). Identification of interpolating affine LPV models for mechatronic systems with one varying parameter. *Eur. J. Control*, 14(1), p. 16–29.

Peppard, L. (1974). String stability of relative-motion PID vehicle control systems. *IEEE Trans. Autom. Control*, 19(5), p. 579–581.

Persson, M., Botling, F., Hesslow, E., and Johansson, R. (1999). Stop & go Controller for adaptive cruise control. In *Proc. IEEE Int. Conf. Control Appl.*, Kohala Coast, HI, USA, Aug., 1999, **2**, p. 1692–1697.

Prestl, W., Sauer, T., Steinle, J., and Tschernoster, O. (2000). The BMW active cruise control ACC. *SAE Techn. Pap. Ser.*, (2000-01-0344).

Pretschner, A., Broy, M., Krüger, I. H., and Stauner, T. (2007). Software engineering for automotive systems: a roadmap. In *Future Softw. Eng.*, Washington, WA, USA, 2007. p. 55–71.

Rajamani, R. and Zhu, C. (2002). Semi-Autonomous Adaptive Cruise Control Systems. *IEEE Trans. Veh. Technol.*, 51(5), p. 1186–1192.

Rajamani, R., Choi, S. B., Law, B. K., Hedrick, J. K., Prohaska, R., and Kretz, P. (2000). Design and experimental implementation of longitudinal control for a platoon of automated vehicles. *J. Dyn. Syst., Meas., Control*, 122(3), p. 470–476.

Raković, S. V., , Grieder, P., Kvasnica, M., Mayne, D. Q., and Morari, M. (2004). Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances. In *Proc. 43rd IEEE Conf. Decis. Control*, Dec., 2004. p. 1418–1423.

Rantzer, A. (1996). On the Kalman-Yakubovich-Popov lemma. *Syst. Control Lett.*, 28(1), p. 7–10.

Rao, C. V. and Rawlings, J. B. (2000). Linear programming and Model Predictive Control. *Journal of Process Control*, 10(2), p. 283–289.

Reichardt, E. (2007). ACC stop-&-go, simulation and visualization. Bachelor's thesis DCT 2007.042, Eindhoven University of Technology, Eindhoven, the Netherlands.

Richter, K. and Ernst, R. (2006). How OEMs and suppliers can face the network integration challenges. In *Proc. Conf. Design, Autom. Test Eur.: Des. Forum*, Munich, Germany, 2006. p. 183–188.

Rijkswaterstaat (2007). *Wegen naar de toekomst: de rij-assistent, systemen die het autorijden ondersteunen (in Dutch)*. Ministry of Transport, Public Works and Water Management.

Rossignol, L., Scorletti, G., and Fromion, V. (2003). Filter design: a finite dimensional convex optimization approach. *Int. J. Robust Nonlin. Control*, 12, p. 1317–1335.

Rugh, W. J. and Shamma, J. S. (2000). Research on gain scheduling. *Automatica*, 36, p. 1401–1425.

Saerens, B., Diehl, M., Swevers, J., and Van den Bulck, E. (2008). Model predictive control of automotive powertrains - first experimental results. In *Proc. 47th IEEE Conf. Decis. Control*, Cancun, Mexico, Dec., 2008. p. 5692–5697.

Santhanakrishnan, K. and Rajamani, R. (2003). On spacing policies for highway vehicle automation. *IEEE Trans. Intell. Transp. Syst.*, 4(4), p. 198–204.

Scherer, C. W. (1999). *Robust mixed control and LPV control with full block scalings*. Siam, Philadelphia.

Scherer, C. W. (2001). LPV control and full block multipliers. *Automatica*, 37, p. 361–375.

Scherer, C. W., Gahinet, P., and Chilali, M. (1997). Multiobjective output-feedback control via LMI optimization. *IEEE Trans. Autom. Control*, 42, p. 896–911.

Schuette, H. and Waeltermann, P. (2005). Hardware-in-the-loop testing of vehicle dynamics controllers - a technical survey. *SAE Techn. Pap. Ser.*, (2005-01-1660).

Seiler, P., Pant, A., and Hedrick, J. K. (2004). Disturbance propagation in vehicle strings. *IEEE Trans. Autom. Control*, 49(10), p. 1835–1841.

Seron, M. M., Braslavsky, J. H., and Goodwin, G. C. (1997). *Fundamental limitations in filtering and control*. Springer-Verlag, London, UK.

Shaw, E. and Hedrick, J. K. (2007a). Controller design for string stable heterogeneous vehicle strings. In *Proc. 46th IEEE Conf. Decis. Control*, Dec., 2007a. p. 2868–2875.

Shaw, E. and Hedrick, J. K. (2007b). String stability analysis for heterogeneous vehicle strings. In *Proc. Am. Control Conf.*, New York, NY, Jul., 2007b. p. 3118–3125.

Sheikholeslam, S. and Desoer, C. A. (1992). A system level study of the longitudinal control of a platoon of vehicles. *J. Dyn. Sys., Meas., Control*, 114(2), p. 286–292.

Sheikholeslam, S. and Desoer, C. A. (1993). Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: a system level study. *IEEE Trans. Veh. Technol.*, 42(4), p. 546–554.

Shladover, S. E. (2005). Automated vehicles for highway operations (automated highway systems). *Proc. IMechE., Part I: J. Syst. Control Eng.*, 219(1), p. 53–75.

Shladover, S. E., Desoer, C. A., Hedrick, J. K., Tomizuka, M., Walrand, J., Zhang, W.-B., McMahon, D. H., Peng, H., Sheikholeslam, S., and McKeown, N. (1991). Automated vehicle control developments in the PATH program. *IEEE Trans. Veh. Technol.*, 40(1), p. 114–130.

Skogestad, S. and Postlethwaite, I. (2005). *Multivariable feedback control - Analysis and design.* John Wiley & Sons, Ltd., Chichester, UK, 2nd edition.

Steinbuch, M., Terlouw, J. C., Bosgra, O. H., and Smit, S. G. (1992). Uncertainty modelling and structured singular value computation applied to an electro-mechanical system. *IEE Proc. - D*, 139(3), p. 301–307.

Stilwell, D. J. and Rugh, W. J. (2000). Stability preserving interpolation methods for the synthesis of gain scheduling controllers. *Automatica*, 36, p. 665–671.

Stroop, J. and Stolpe, R. (2006). Prototyping of automotive control systems in a time-triggered environment using FlexRay. *Proc. IEEE Conf. Comp. Aided Control Syst. Design*, p. 2332–2337.

Sugiyama, Y., Fukui, M., Kikuchi, M., Hasebe, K., Nakayama, A., Nishinari, K., Tadaki, S., and Yukawa, S. (2008). Traffic jams without bottlenecks - experimental evidence for the physical mechnism of the formation of a jam. *New J. Physics*, 10(033001), p. 7.

Sun, J., Kolmanovsky, I., Cook, J. A., and Buckland, J. H. (2005). Modeling and control of automotive powertrain systems: a tutorial. In *Am. Control Conf.*, Portland, OR, USA, Jun., 2005. p. 3271–3283.

Swaroop, D. and Hedrick, J. K. (1999). Constant spacing strategies for platooning in automated highway systems. *J. Dyn. Syst., Meas., Control*, 121(3), p. 462–470.

Swaroop, D., Hedrick, J. K., Chien, C. C., and Ioannou, P. A. (1994). A comparison of spacing and headway control strategy for automatically controlled vehicles. *Veh. Syst. Dyn.*, 23(8), p. 597425.

Swaroop, D., Hedrick, J. K., and Choi, S. B. (2001). Direct Adaptive Longitudinal Control of Vehicle Platoons. *IEEE Trans. Veh. Technol.*, 50(1), p. 150–161.

TNO Automotive (2009). PreScan, VEHIL (vehicle harware in the loop) test faciliy, and MACS (modular automotive control system). `http://www.tno.nl/automotive/` [online], Aug., 2009. TNO Automotive, Helmond, the Netherlands.

The MathWorks^{TM} (2010). Software for technical computing and model-based design. `http://www.mathworks.com` [online], May, 2010.

Tóth, R., Felici, F., Heuberger, P. S. C., and Van den Hof, P. M. J. (2007). Discrete time LPV I/O and state space representations, differences of behavior and pitfalls of interpolation. In *Proc. Eur. Control Conf.*, Kos, Greece, Jul., 2007. p. 5418–5425.

Das V-Modell (2006). Fundamentals of the V-modell. Technical Report, IABG.

Vahidi, A. and Eskandarian, A. (2003). Research advances in intelligent collision avoidance and adaptive cruise control. *IEEE Trans. Intell. Transp. Syst.*, 4(3), p. 143–153.

Venhovens, P., Naab, K., and Adiprasito, B. (2000). Stop and go cruise control. *Int. J. Automot. Technol.*, 1(2), p. 61–69.

Vugts, R. P. A. (2010). *String-stable CACC design and experimental validation.* Master's thesis, Eindhoven University of Technology, Eindhoven, the Netherlands. DCT 2009.131.

WABCO (2010). WABCO vehicle control systems. `http://www.wabco-auto.com` [online], Jan., 2010. Brussels, Belgium.

Ward, D. K. and Fields, H. L. (2000). A Vision of the Future of Automotive Electronics. *SAE Techn. Pap. Ser.*, (2000-01-1358).

Warnick, S. C. and Rodriguez, A. A. (1994). Longitudinal control of a platoon of vehicles with multiple saturating nonlinearities. In *Proc. Am. Control Conf.*, Baltimore, MD, Jun., 1994, **I**, p. 403–407.

Weik, W., Friedmann, O., Anger, I., and Werner, O. (2004). Die Kupplung, das Herz des Doppelkupplungsgetriebes. *Getr. Fahrz., VDI-Ges.*, (S), p. 65–88.

Widmann, G. R., Daniels, M. K., Hamilton, L., Humm, L., Riley, B., Schiffmann, J. K., Schnelker, D. E., and Wishon, W. H. (2000). Comparison of lidar-based and radar-based adaptive cruise control systems. *SAE Techn. Pap. Ser.*, (2000-01-0345).

Wijnheijmer, F. P., Naus, G. J. L., Post, W. J. A. E. M., Steinbuch, M., and Teerhuis, A. P. (2006). Modeling and LPV control of an electro-hydraulic servo system. In *IEEE Conf. Control Appl.*, Munich, Germany, Oct., 2006. p. 3116–3121.

Wills, L., Kannan, S., Sander, S., Guler, M., Heck, B., Prasad, J. V. R., Schrage, D., and Vachtsevanos, G. (2001). An Open Platform for Reconfigurable Control. *IEEE Control Systems Magazine*, 21(3), p. 49–64.

Winkel, M., Karch, G., and Steinel, K. (2004). Kupplungsrupfen, Entstehung und Gegenmassnahmen bei AS-Getriebe. *Getr. Fahrz., VDI-Ges.*, (S), p. 253–364.

Wismans, J. S. H. M. (2007). Characteristics of the automotive industry in Holland. 1th CCAR Sci. Meet., Jun., 2007.

World Health Organization (2009). *Global status report on road safety: time for action.* World Health Organization, Dept. of Violence and Injury Prevention and Disability. ISBN: 9241563842.

Wu, F. and Dong, K. (2006). Gain-scheduling control of LFT systems using parameter-dependent Lyapunov functions. *Automatica*, 42, p. 39–50.

Wu, F., Packard, A., and Balas, G. (1995). LPV control design for pitch-axis missile autopilots. In *Proc. 34th Conf. Decis. Control*, New Orleans, LA, USA, Dec., 1995, **1**, p. 188–193.

Wu, F., Yang, X. H., Packard, A., and Becker, G. (1996). Induced $L2$-norm control for LPV systems with bounded parameter variations rates. *Int. J. Robust Nonlin. Control*, 6(9–10), p. 983–998.

Yamada, N. and Ando, K. (1995). An analysis of clutch self-excited vibration in automotive drive-line. *SAE Techn. Pap. Ser.*, 951319.

Yanakiev, D. and Kanellakopoulos, I. (1998). Nonlinear spacing policies for automated heavy-duty vehicles. *IEEE Trans. Veh. Technol.*, 47(4), p. 1365–1377.

Yanakiev, D., Eyre, J., and Kanellakopoulos, I. (1998). Analysis, design and evaluation of AVCS for heavy-duty vehicles with actuator delays. Technical Report UCB-ITS-PRR-98-18, California PATH Research Report.

Yin, G., Chen, N., and Li, P. (2007). Improving handling stability performance of four-wheel steering vehicle via μ-synthesis robust control. *IEEE Trans. Veh. Technol.*, 56(5), p. 2432–2439.

Yu, F., Li, D.-F., and Crolla, D. A. (2008). Integrated vehicle dynamics control - state-of-the-art review. In *Proc. Veh. Power Prop. Conf.*, Harbin, China, Sep., 2008.

Yu, J. and Sideris, A. (1995). H_∞ control with parametric Lyapunov functions. In *Proc. 34th Conf. Decis. Control*, 1995.

Zalm, G. M. van der, Huisman, R. G. M., Steinbuch, M., and Veldpaus, F. (2008). Frequency domain approach for the design of heavy-duty vehicle speed controllers. *Int. J. Heavy Veh. Syst.*, 15(1), p. 107–123.

Zhang, J. and Ioannou, P. (2004). Integrated Roadway / adaptive cruise control System: safety, performance, environmental and near term deployment considerations. Technical Report UCB-ITS-PRR-2004-32, California PATH Research Report.

Zhou, K., Doyle, J. C., and Glover, K. (1996). *Robust and optimal control*. Prentice Hall, New Yersey.

Samenvatting

Het gebruik van gedistribueerde regelsystemen in moderne voertuigen is in de afgelopen decennia exponentieel toegenomen. Vandaag de dag wordt het merendeel van de prestatieverbeteringen en innovaties in de automobielindustrie opgelost met behulp van embedded regelsystemen. Een moderne auto of vrachtwagen kan dan ook als een complex, mechatronisch systeem worden beschouwd. Echter, in de praktijk komt het ontwerp van regelsystemen vaak neer op tijdrovend online instellen en kalibreren van een handmatig gekozen regelaarstrucuur, in plaats van meer systematische, model-gebaseerde synthese van de regelaar.

Het hoofddoel van dit proefschrift is het leveren van een bijdrage aan verandering van deze werkwijze. Het doel is om een systematische, model-gebaseerde aanpak toe te passen, waarbij het gebruik van regeltechnisch-georiënteerde modellen en de definitie van bijbehorende prestatie-eisen als basis dienen voor het ontwerp van een regelaar. Het gebruik van een dergelijke aanpak is een voorwaarde voor de toepassing van beschikbare regelaar-synthese-methoden, waarmee het mogelijk is garanties te geven ten aanzien van robuustheid, stabiliteit en prestaties van het resulterende systeem. Vanuit een praktisch oogpunt vormt dit de basis om het gebruik van tijdrovende instel- en kalibratietechnieken terug te dringen en draagt het bij tot de realisatie van steeds strengere prestatie-eisen.

Om deze mogelijkheden te demonstreren zijn vier verschillende voorbeelden gebruikt. In alle voorbeelden zijn de resultaten in praktijk geëvalueerd met testvoertuigen en een zogenaamde 'hardware-in-the-loop' (HIL) opstelling.

- Voorbeeld I: Het ontwerp van een robuust stabiele regeling voorkomt trillingen in de aandrijflijn die met name tijdens het wegrijden door de koppeling worden geïntroduceerd. Het ontwerp voorkomt de noodzaak om de instellingen van het regelsysteem te wijzigen indien de relevante dynamica ten gevolge van slijtage in de koppeling verandert. Een HIL opstelling is gebruikt voor experimentele validatie.

- Voorbeeld II: Grote variaties in de belading en in de overbrenging van de versnellingsbak zijn typisch voor vrachtwagens. Door in het ontwerp van een snelheidsregeling expliciet rekening te houden met deze variaties, blijft het gesloten-lus gedrag hetzelfde over het hele werkgebied. Voor experimentele validatie zijn testen met een DAF XF105 uitgevoerd, zowel met als zonder oplegger.

- Voorbeeld III: Een systematische aanpak voor het ontwerp van een adaptieve snelheidsregeling verkort de tijd die nodig is om het systeem in te stellen. Bovendien is er een intuïtieve koppeling tussen de instellingen en de gewenste prestatiecriteria, in dit geval comfort en veiligheid. Het ontwerp is gevalideerd met een Audi S8.
- Voorbeeld IV: Een coöperatieve adaptieve snelheidsregeling is ontworpen, waarbij de nadruk ligt op de praktische uitvoerbaarheid van het ontwerp. Op basis van het ontwerp is een noodzakelijke en voldoende voorwaarde voor de 'string stabiliteit' van een rij voertuigen afgeleid. Twee Citroën C4's zijn gebruikt voor experimentele validatie van deze regelaar, die de doorstroming van het verkeer verbetert, terwijl string stabiliteit van de verkeersstroom wordt gegarandeerd.

De vier voorbeelden behandelen typische uitdagingen in het ontwerp van regelsystemen in de automobielindustrie, namelijk i) de aanwezigheid van een breed scala aan variabele werkpunten, en ii) het gebruik van globale prestaties-eisen. Op basis van het verkregen inzicht in de aanpak van deze uitdagingen is een generieke classificatie van regeltechnische functies in de automobielindustrie opgesteld. De classificatie onderscheidt functies op drie niveaus, namelijk het voertuig als geheel, de systemen in een voertuig, en de systemen op een component niveau. Aan de hand van de classificatie kunnen regeltechnische problemen worden gekarakteriseerd op basis van de te volgen modellering en de definitie van prestatie-eisen die als basis dienen voor het uiteindelijke regelaarontwerp.

Functionaliteit op het niveau van het volledige voertuig concentreert zich op de definitie van het gewenste gedrag van het voertuig als geheel. De vereiste modellering is typisch beperkt. Daarnaast moeten globale prestatie-eisen worden vertaald in regeltechnisch-georiënteerde prestatie-eisen. Functionaliteit op het niveau van in-voertuig systemen concentreert zich op het feitelijke aansturen van de complexe voertuig dynamica. Daarnaast worden de modellering en de definitie van prestatie-eisen typisch beïnvloed door een grote verscheidenheid aan werkpunten.

De voorbeelden laten zien dat er synthese-methoden beschikbaar zijn die bijzonder geschikt zijn om de genoemde uitdagingen aan te pakken. In dit onderzoek zijn 'robust control', 'model predictive control', en 'gain scheduling' of 'linear parameter varying control' toegepast. De resultaten tonen bovendien de mogelijkheden voor praktische implementatie van deze methoden. Echter, de complexiteit van de methoden vertaalt typisch in lange rekentijden, terwijl het inzicht in de resulterende regelaar vermindert. Hierdoor wordt bijvoorbeeld de online fijnafstelling van de regelaar bemoeilijkt. Om de praktische toepassing van deze technieken te verbeteren zijn snelle en efficiënte algoritmen nodig.

Dankwoord

Een proefschrift is nooit af en toch komt er een moment waarop je er de laatste hand aan legt. Dit is dat moment. Ondanks dat er slechts één naam op de voorkant staat, hebben veel personen ieder op hun eigen manier bijgedragen aan het bereiken van dit moment. Bedankt!

Als eerste wil ik Maarten en René bedanken voor de mogelijkheid die ze mij gegeven hebben om dit promotietraject te doorlopen. Maarten, bedankt voor je enthousiasme, vertrouwen, interesse en alle mogelijkheden die je mij hebt gegeven om me te ontwikkelen op meer dan alleen het wetenschappelijke vlak. René, bedankt voor je begeleiding, je kritische blik, en het meedraaien in jouw 'onderwijs- en andere activiteitenteam': een erg leerzame afwisseling en aanvulling op de dagelijkse bezigheden.

Rudolf en Jeroen, namens DAF en TNO vertegenwoordigen jullie de automobielindustrie, de praktijk. Bedankt voor jullie enthousiasme, tijd en begeleiding. De praktijktesten die jullie mogelijk hebben gemaakt, blijven voor mij het mooiste moment van een regelaarontwerp: met een laptop op schoot in een truck of personenwagen de controls in handen! Ik vond het een uitdaging om de ideeën en visies van drie partijen te stroomlijnen. Jullie interesse in de theorie en brede blik heeft hier zeker bij geholpen, we werken op dit moment niet voor niets in hetzelfde gebouw. Bedankt!

Tot slot wil ik in dit rijtje Maurice en Vital bedanken. Jullie frisse blik op verschillende onderdelen in het onderzoek heb ik als uitdagend en bijzonder motiverend ervaren. Daarnaast wil ik Jan Swevers, Bart De Schutter en Paul van den Bosch bedanken voor het plaatsnemen in de kerncommissie en het verdiepen in mijn proefschrift. Met jullie feedback heb ik de puntjes op de 'i' kunnen zetten.

Dit proefschrift was niet tot stand gekomen zonder de hulp van een aantal prima studenten: Menno, Roel, Koen, Peter, René, GertJan, Erik en Joris bedankt voor jullie bijzondere bijdrages, inzet en enthousiasme! Zonder jullie had ik niet zoveel verschillende onderwerpen kunnen behandelen en onze creaties niet in het echt kunnen testen. Hierbij wil ik ook iedereen van DAF en TNO bedanken die heeft meegeholpen bij het optuigen van de testvoertuigen en uitvoeren van de testen: Michiel, Giel, Johan, Peter, Edsel, Bart, Thijs, Thijs, Hanno en alle anderen die hier, soms tot in de late uurtjes, aan hebben gewerkt.

Naast alle directe bijdrages zijn er veel personen die indirect een bijdrage hebben geleverd aan de totstandkoming van dit proefschrift. Op de eerste plaats mijn collega's binnen de DCT groep. Bedankt voor de geweldig goede sfeer op de gang, de koffie 's morgens, de gezelligheid op het secretariaat, de activiteiten naast de dagelijkse werkzaamheden en de zakenreizen met de bijbehorende acclimatisatieperiodes. In het bijzonder wil ik hierbij mijn roomies en het team van Tech United bedanken. Zonder Tech United had dit proefschrift er wellicht eerder gelegen, maar was ik nooit vice-wereldkampioen geworden! Heren en dame, bedankt voor de vele gezellige klusuren, de demo's en natuurlijk de onvergetelijke toernooien. Francois, René en Benjamin. Bedankt voor de gezelligheid, onze discussies over het systeem, zo nu en dan een koffie en het prima arbeidsklimaat op onze kamer.

Ook buiten mijn werk heb ik op veel vlakken de mogelijkheid gehad mijn blik te verbreden en energie kwijt te kunnen. Hiervoor wil ik in het bijzonder mijn familie en vrienden bedanken. Bedankt voor de gezelligheid, feestjes, uitjes, wandelingen, het klimmen en de vakanties. Voor de repetities en fantastische concerten. Voor de interesse in mijn werk en voor de mogelijkheden om ook op organisatorisch vlak wat te ondernemen. Voldoende excitatie op andere vlakken is voor mij onmisbaar geweest. Bedankt!

Tenslotte degenen die eigenlijk als eerste hadden moeten staan. Pap en mam. Bedankt voor jullie onvoorwaardelijke steun, jullie interesse, enthousiasme en doorzettingsvermogen. Jullie zijn een voorbeeld voor me. Ilse. Je bent er vanaf het begin bij en staat altijd voor me klaar. Dank je voor onze avonturen, je liefde, en ons thuis. Dank je wel voor alles.

Gerrit Naus
september 2010

Curriculum Vitae

Gerrit Naus was born on September 29, 1979 in Roermond, the Netherlands. After finishing his pre-university education at Bisschoppelijk College Broekhin in 1998, he studied mechanical engineering at the Eindhoven University of Technology. In 2005 he received his Master's degree on the topic of 'Modeling and control of a direct current up-set resistance butt welding process'. His graduation project was performed at Fontijne Grotnes B.V. in Vlaardingen, at which, after receiving his Master's degree, he continued working for five months as a development engineer at the R&D department. From 2006, he started as a Ph.D. student in the Control Systems Technology group at the department of mechanical engineering of the Eindhoven University of Technology on the topic of 'Model-based control for automotive applications', which resulted in this thesis. During his Ph.D. project, he completed the educational program of the Graduate School DISC and was a member of the Tech United team in the RoboCup competition.