Design of a Hybrid Adaptive Cruise Control Stop-&-Go system

R.A.P.M. van den Bleek

DCT 2007.125

TU/e Master's Thesis
October 2006 - October 2007

Coaches: ir. B.T.M. Scheepers (TNO)
ir. J. Ploeg (TNO)
ir. G.J.L. Naus (TU/e)
dr. ir. M.J.G. van de Molengraft (TU/e)

Supervisor: prof. dr. ir. M. Steinbuch (TU/e)

Engineering thesis committee: prof. dr. ir. M. Steinbuch (Chairman, TU/e)
dr. ir. M.J.G. van de Molengraft (TU/e)
ir. G.J.L. Naus (TU/e)
dr. ir. I. Lopez (TU/e)
ir. B.T.M. Scheepers (Advisor, TNO)

TNO Science & Industry
Business Unit Automotive
Department of Integrated Safety

Technische Universiteit Eindhoven
Department of Mechanical Engineering
Control System Technology Group

Preface

This report is the result of a traineeship at TNO, Science & Industry, Business Unit Automotive, department of Integrated Safety in Helmond in partial fulfillment of my graduation at the Technische Universiteit Eindhoven, faculty Mechanical Engineering.

I would like to thank Gerrit Naus at TU/e, Bart Scheepers and Jeroen Ploeg at TNO for providing the opportunity of this master's project, their support, guidance and creating a nice and inspiring work environment during this project. I would also like to thank René van de Molengraft and Maarten Steinbuch (both TU/e) for their supervision and criticism. For participating in my graduation committee, I would like to thank Ines Lopez.

Further, I would like to thank my family for their support, especially my parents for supporting me mentally as well as financially. And last but not least, I would like to thank my girlfriend Suzan Oomen for her mental support and the necessary distraction during this project.

Roel van den Bleek
R.A.P.M.v.d.Bleek@gmail.com

Breda / Helmond
October 29, 2007

Summary

This master's thesis focusses on the design of an easy-to-tune and vehicle-independent control strategy for Adaptive Cruise Control Stop-&-Go (ACC Stop-&-Go) applications. ACC Stop-&-Go provides assistance to the driver in the task of longitudinal control of the vehicle during driving. ACC controls the velocity and distance of the host vehicle to safely follow a target vehicle with a possibly varying velocity, intended to reduce driver workload. Also, the appearance of a new target vehicle with a different relative distance and relative velocity should be handled correctly. The addition of Stop-&-Go gives the possibility to do this for low or zero velocities as well (e.g. stopping in traffic jams). In case of no target vehicle, the host vehicle drives in Cruise Control mode.

Recently, TNO Automotive has developed an ACC Stop-&-Go control system for a heavy duty truck and has implemented it successfully. The basis of the ACC concept is, in fact, a simple PD-controller. However, in order to obtain a good compromise between comfort and safety for different situations, the final tuning (by means of situation-dependency of the gains) of the ACC Stop-&-Go controller has become rather complicated and time consuming. To be able to use the controller concept in a practical way, an easy-to-tune (re)design of the ACC Stop-&-Go controller is required.

A literature survey on ACC Stop-&-Go reveals the objectives, requirements and constraints. After this, an appropriate control framework is adopted, taking into account the previous mentioned aspects. Model Predictive Control (MPC) can deal with constrained multivariable control problems. Instead of conventional control which uses a pre-computed control law, MPC chooses a control action by repeatedly solving on-line an optimal control problem over a future horizon. The MPC controller for this ACC Stop-&-Go application is designed explicitly (off-line) to be able to use the MPC controller for fast sampling real-time applications. Explicit MPC control is also known as Hybrid control. The result is a Hybrid Adaptive Cruise Control Stop-&-Go system.

For the actual implementation in a vehicle, several functionalities are added. An example is the addition of cruise control: the objective for the host vehicle is to drive at the driver specified cruise control velocity. The validation and performance evaluation of the designed hybrid ACC Stop-&-Go controller has been carried out. Firstly, simulations are performed with MATLAB, SIMULINK and PRESCAN. These simulations make it possible to test the ACC controller for a complete envelope of working conditions. Secondly, experiments with an instrumented vehicle have been performed. Two situations are presented and discussed on a simulation level as well as on an experimental level. Based on these results, the main conclusion of this thesis is that an Adaptive Cruise Control Stop-&-Go system is successfully designed with Hybrid control, in simulations as well as road experiments.

Samenvatting (Dutch)

De focus van dit afstudeerverslag is het ontwerp van een makkelijk in-te-stellen en voertuig- onafhankelijke regelstrategie voor Adaptive Cruise Control Stop-&-Go (afgekort: ACC Stop-&-Go) applicaties. ACC assisteert de bestuurder in de taak van het longitudinaal regelen van het voertuig tijdens het rijden. ACC regelt de snelheid en afstand van het voertuig om veilig een voorganger te kunnen volgen, met een mogelijk variërende snelheid, om de werkdruk van de bestuurder te verlagen. Daarnaast moet het verschijnen van een nieuwe voorganger met een verschillende relatieve afstand en relatieve snelheid correct afgehandeld worden. De toevoeging Stop-&-Go maakt het mogelijk dit ook te doen voor lage snelheden of bij stilstand (bijvoorbeeld stoppen en wegrijden in files). Wanneer geen voorganger aanwezig is, zal het voertuig in Cruise Control mode rijden.

TNO Automotive heeft recent een ACC Stop-&-Go regelsysteem ontwikkeld voor een vrachtwagen. Dit systeem is succesvol geïmplementeerd. De basis van dit ACC concept is, in feite, een simpele PD regeling. Echter, de instelling (door middel van situatie-afhankelijke versterkingen) van de ACC Stop-&-Go regeling is gecompliceerd en tijdrovend om een goed compromis tussen comfort en veiligheid voor verschillende situaties te verkrijgen. Om het regelaar concept op een praktische manier te kunnen gebruiken, is een (her-)ontwerp van de ACC Stop-&-Go regelaar nodig.

Een literatuurstudie naar ACC Stop-&-Go resulteert in doelstellingen, vereisten en limitaties voor deze applicatie. Hierna wordt een geschikte regelstrategie geïntroduceerd, welke rekening houdt met de genoemde aspecten. Model Predictive Control (MPC) kan omgaan met multivariabele regelsystemen met begrenzingen aan ingangen, states en/of uitgangen. In plaats van conventionele strategieën, welke gebruik maken van een vooraf berekende regelwet, berekent MPC een regelactie door herhaaldelijk on-line een optimaal regelprobleem op te lossen. De MPC regeling voor deze ACC applicatie wordt expliciet (off-line) ontworpen, zodat de MPC regeling voor snel samplende applicaties gebruikt kan worden. Deze expliciete MPC regeling wordt ook wel een Hybride regeling genoemd. Het resultaat is een Hybride Adaptive Cruise Control Stop-&-Go systeem.

Voor de daadwerkelijke implementatie in een voertuig zijn verschillende functionaliteiten toegevoegd. Een voorbeeld hiervan is Cruise Control: de doelstelling voor het voertuig is om met een - door de bestuurder - ingestelde snelheid te rijden. Validatie en evaluatie van de ontworpen Hybride ACC Stop-&-Go regeling is uitgevoerd. Als eerste zijn simulaties uitgevoerd met Matlab, Simulink en PreScan. Deze simulaties maken het mogelijk om de ACC Stop-&-Go regeling te testen voor een compleet werkgebied. Ten tweede zijn experimenten met een geïnstrumenteerd voertuig uitgevoerd. Twee situaties worden gepresenteerd op zowel simulatie niveau als op experimenteel niveau. Gebaseerd op deze resultaten is de conclusie voor dit afstudeerproject dat een Adaptive Cruise Control Stop-&-Go regeling succesvol is ontworpen met Hybride regeltheorie, voor zowel simulaties als voor wegexperimenten.

List of abbreviations

Abbreviation	Description
ACC	Adaptive Cruise Control
ABS	Anti-lock Braking System
ADAS	Advanced Driver Assistance Systems
AGV	Automated Guided Vehicle
CACC	Coöperative Adaptive Cruise Control
CAN	Controller Area Network
CarLab	Car Laboratory
CC	Cruise Control
CFTOC	Constrained Finite Time Optimal Control
DAS	Driver Assistance System
FMCW	Frequency Modulated Continuous Wave
HMI	Human Machine Interface
KKT	Karush-Kuhn-Tucker
LMI	Linear Matrix Inequalities
LP	Linear Program(ming)
LTI	Linear Time-Invariant
MB	Moving Base
MILP	Mixed Integer Linear Program(ming)
MIO	Most Important Object
MIQP	Mixed Integer Quadratic Program(ming)
MLD	Mixed Logical Dynamical
MPC	Model Predictive Control
mpQP	multi-parametric Quadratic Program(ming)
mpLP	multi-parametric Linear Program(ming)
mpMILP	multi-parametric Mixed Integer Linear Program(ming)
mpMIQP	multi-parametric Mixed Integer Quadratic Program(ming)
MPT	Multi-Parametric Toolbox
PID	Proportional-Integral-Derivative controller
PSD	Power Spectral Density
PWA	PieceWise Affine
PWL	PieceWise Linear
QP	Quadratic Program(ming)
RHC	Receding Horizon Control
SS	State-Space
VeHIL	Vehicle Hardware-In-the-Loop
YALMIP	Yet Another LMI Parser
ZOH	Zero-Order Hold

List of symbols

Symbol	Unit	Description
x	m	position
$\overline{x}$	-	system state vector
x_{rr}	m	radar range
v	$\mathrm{m\,s^{-1}}$	velocity
v_{CC}	$\mathrm{m\,s^{-1}}$	CC velocity
a	$\mathrm{m\,s^{-2}}$	acceleration
j	$\mathrm{m\,s^{-3}}$	jerk
u	$\mathrm{m\,s^{-2}}$	control output
δu	$\mathrm{m\,s^{-3}}$	derivative of control output
$\overline{\varepsilon}$	-	error vector
$\overline{y}$	-	output vector
λ	-	Lagrange multipliers
t	s	continuous time
t_{hw}	s	time headway
TTC	s	time to collision
T_s	s	sample time
N_y	-	output horizon
N_u	-	input horizon
N_c	-	control horizon
Q	-	state weighting matrix
R	-	control weighting matrix
R	-	number of regions

Super and subscripts

Symbol	Description
$(\cdot)^T$	transpose of $(\cdot)$
$(\cdot)_h$	host vehicle
$(\cdot)_{rt}$	real target vehicle
$(\cdot)_{vt}$	virtual target vehicle
$(\cdot)_r$	relative, i.e. $x_r = x_t - x_h$
$(\cdot)_d$	desired value
$(\cdot)_k$	time increment
$(\cdot)_{k+1}$	next time increment
$(\cdot)_{max}$	maximum value
$(\cdot)_{min}$	minimum value
$(\cdot)_e$	extended state, i.e. 4D instead of 3D
$(\cdot)_{th}$	throttle
$(\cdot)_{br}$	brake
$(\cdot)_i$	region number

Contents

Chapter 1

Introduction

This chapter presents the assignment, together with the focus of this project. Finally, the outline of this master's thesis is presented. To start with, the company TNO (Automotive), where this master's thesis has been carried out, is introduced.

1.1 TNO

TNO is a knowledge organization for companies, government bodies and public organizations. The daily work of some 5.000 employees is to develop and apply knowledge. TNO provides contract research and specialist consultancy as well as grant licences for patents and specialist software. TNO tests and certifies products and services, and issues an independent evaluation of quality. Also, TNO sets up new companies to market innovations [51].
TNO is active in five core areas:

- Quality of Life
- Defence, Security and Safety
- Science and Industry
- Built Environment and Geosciences
- Information and Communication Technology

TNO Science and Industry - TNO Automotive

TNO Science and Industry addresses on the next markets: High-end equipment, Automotive, Space, Process industry, Sports, Medical and Public sector.
The business unit Automotive aims to contribute to make our future means of transport safer, cleaner and more efficient. Research and development of automotive systems and components is the core business. Powertrains, Homologation, Integrated safety and Crash testing are the four departments of the business unit Automotive.

Integrated safety

Traffic safety has improved considerably in recent decades with the application of better test methods, simulation methods and modeling. When it comes to the field of safety in the automotive market of today, the borders between active and passive safety are fading. Integrated Safety combines knowledge of human body modeling with key automotive issues such as PreCrash, (Advanced) Driver Assistance Systems and Vehicle Control. TNO strives to provide solutions for the automotive market, searching for ways to implement these topics in their own safety systems and other systems intended to relieve the driver of vehicle control tasks.

1.2 Assignment

Recently, TNO Automotive has developed an ACC Stop-&-Go control system for a heavy duty truck and implemented it successfully. ACC stands for Automated (or Automatic / Advanced / Autonomous / Adaptive) Cruise Control and enables automatic following of a predecessor in controlling the distance to this predecessor by adjusting the velocity of the vehicle automatically. The addition Stop-&-Go indicates the possibility to do this for low or zero velocities as well (e.g. stopping in traffic jams). A sensor is able to detect the relative distance and relative velocity between the predecessor and the (ACC equipped) vehicle. The basis of the ACC concept is, in fact, a simple PD-controller. However, in order to obtain a good compromise between comfort and safety for different situations, the final tuning (by means of situation-dependency of the gains) of the ACC Stop-&-Go controller has become rather complicated and time consuming. To be able to use the controller concept in a practical way, an easy-to-tune (re)design of the controller is required.

This leads to the following assignment:

(Re-) Design of an easy-to-tune and vehicle-independent Adaptive Cruise Control
Stop-&-Go system for a velocity range from zero to high velocities

From the problem statement the following research questions and/or goals are formulated:

1. Gain insight in ACC Stop-&-Go and analyse control strategies for ACC Stop-&-Go. The new control concept should aim at an easy-to-tune structure.
2. Design of the controller for ACC Stop-&-Go. Define the control objectives, requirements and constraints and translate these into the control framework.
3. Evaluate the new ACC Stop-&-Go controller with simulations and experiments.

This master thesis is a case study within the CCAR-project 'Robust Design and Automated Tuning of Automotive Controllers'. A subsequent study of techniques and methods is executed for automated tuning of automotive controllers. The project is executed in association with DAF Trucks N.V., Eindhoven and TNO Automotive, Helmond.

1.3 Focus assignment

The ACC Stop-&-Go structure can be divided into three generic parts (see Figure 1.1). The first one includes the sensor and the corresponding traffic situation (dependant on environment and driver), resulting in the identification of a target vehicle. This is depicted in the upper part, denoted by 1 in Figure 1.1. Secondly, the master control loop determines the actual behaviour of the ACC Stop-&-Go system (middle part of Figure 1.1), denoted by 2 in the figure. Thirdly, the slave control loop comprises the actual longitudinal controller of the vehicle, determines the vehicle response and controls the appropriate vehicle actuators. This third part is shown in the lower part of Figure 1.1. Decoupling the engine and brake control tasks from the overall vehicle control task enables a generic design of the master control loop applicable to any type of vehicle with its vehicle specific slave loop controller. In the following list the different blocks are explained.

- Environment
  The environment is the combination of the vehicle with ACC (so-called host vehicle) together with other vehicles (so-called target vehicles), the road including traffic lights and the interaction between them. No communication is assumed with, e.g. traffic lights or other vehicles.
- Sensor, target identification
  The sensor is used to detect the environment of the vehicle. The sensor information needed by an ACC system comprises relative velocity v_r [m/s], relative distance x_r [m], angle [°] with respect to a target vehicle, Most Important Object (MIO) detected [0/1] & Object IDentification (ObjectID). This is specific for the used Groeneveld/RoadEye Forward Looking Radar, which is a Frequency Modulated Continuous Wave (FMCW) radar. In this study however, focus lies on the actual controller design rather than selection of the best sensors and identification algorithms. No inter-vehicle communication is available, such as Cooperative Adaptive Cruise Control (CACC) systems and sensor fusion (e.g. vision for detection of other vehicles).
- Driver
  ACC systems are designed to act as natural as possible in the sense of driver's behaviour. As driver's behaviour is very subjective, it is important that the ACC system provides clear information feedback about its operating conditions to the driver. In this way, the driver is able to supervise the system and intervene when necessary. The driver commands manual on / off functionality, manual setting of the host vehicle velocity (the 'cruise control velocity') and a desired time headway relationship to fit the individual's driving preferences. Time headway is the time it takes for the following vehicle to reach the point where the preceding vehicle is at present velocity, in seconds.
  $$t_{hw} = \frac{x_r}{v_h} \tag{1.1}$$
  Generally, a time headway of 1.0 to 1.5 s is reasonable and the driver is able to command the system in discrete steps, e.g. 1.0, 1.2 or 1.6 s [44]. The minimum time headway is limited by the maximum allowable braking deceleration value, the sensor accuracy and the speed of data processing. Intervention of the ACC system by application of the brakes or the gas pedal is possible.
- Master control loop
  This master's project focuses on this part: design of a control strategy to deliver a desired acceleration and deceleration based on the outputs of the sensor/host vehicle, control objectives, constraints of the system and the driver's settings (shaded part in Figure 1.1).
- Slave control loop
  The slave control loop comprises the actual longitudinal control of the vehicle and thus is vehicle (configuration) specific. Depending on the desired vehicle acceleration the throttle or the brakes is/are actuated and controlled. The slave control loop is out of the scope of this project.
- Vehicle
  The vehicle equipped with the ACC system has a sensor, a rapid control prototyping system and an electronic throttle and brake pedal interface.

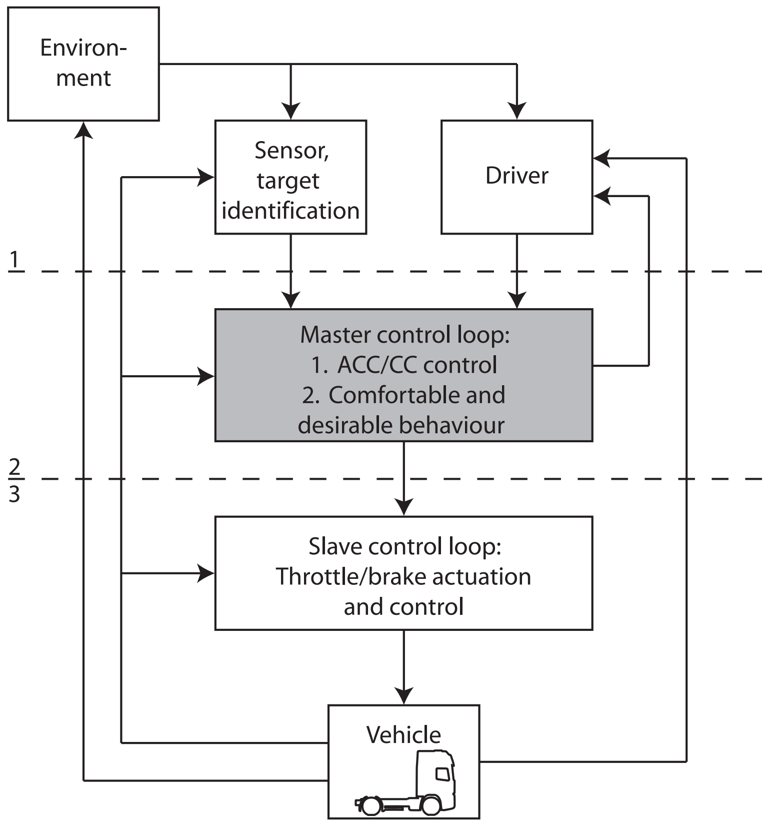


Figure 1.1: The ACC Stop-&-Go structure is divided into three generic parts. Each part contains one or more subsystems. This master's thesis focuses on the the shaded part: master control loop.

1.4 Outline

This thesis is organized as follows. In Chapter 2, ACC known from literature will be discussed including driving related aspects enabling performance evaluation. Besides that, different control strategies / frameworks for an ACC application are compared. Chapter 3 treats the application of a Model Predictive Control (MPC) framework for ACC. In Chapter 4, the implementation of the overall control structure is introduced. The MPC framework for ACC is extended with necessary functionalities. Chapter 5 discusses the test program, simulation and experimental results of two situations of this test program. Finally, in Chapter 6 the conclusions will be drawn and recommendations are presented.

The appendices Appendix A, Appendix B and Appendix C, concerning hybrid systems, MPC and mathematical programming, serve as a detailed background for further reading. In Appendix D, the implementation of switching between an existing/non-existing predecessor in the ACC Stop-&-Go controller is considered. Appendix E shows two different overal control schemes for applying ACC Stop-&-Go in a vehicle. Both schemes are compared at simulation level. In Appendix F simulation results of the designed ACC Stop-&-Go controller are depicted for several defined situations. For the real experiments, a instrumented vehicle is used, which is introduced in Appendix G.

Chapter 2

Adaptive Cruise Control in literature

This chapter consists of three parts: 1. Adaptive Cruise Control as presented in literature 2. Considerations of control strategies for Adaptive Cruise Control, resulting in a choice for a control framework. 3. Several automotive applications of this framework.

2.1 Adaptive Cruise Control in literature

2.1.1 Introduction

Cruise Control (CC) is a common and well known automotive Driver Assistance System (DAS). The driver sets a reference velocity and the throttle is controlled so that this reference velocity is maintained under influence of external loads such as wind, road slope or changing vehicle parameters.

Further development resulted in ACC. ACC takes the traffic in front of the car into account, support the driver in its driving process and the goal is to increase driving comfort and traffic safety. Such systems are generally known as Advanced Driver Assistance Systems (ADAS). An ADAS is defined as a vehicle control system that uses environment sensors to improve driving comfort and/or traffic safety by assisting the driver in recognizing and reacting to potentially dangerous traffic situations. Some examples of these ADAS include Navigation systems, Adaptive Cruise Control, Lane departure warning, Collision detection, Intelligent Speed Adaptation, Car-to-car/car-to-infrastructure communication and Adaptive lighting [19].
In Figure 2.1, the ACC system is introduced: the left vehicle is the host vehicle equipped with an ACC system, the right vehicle is the target vehicle.

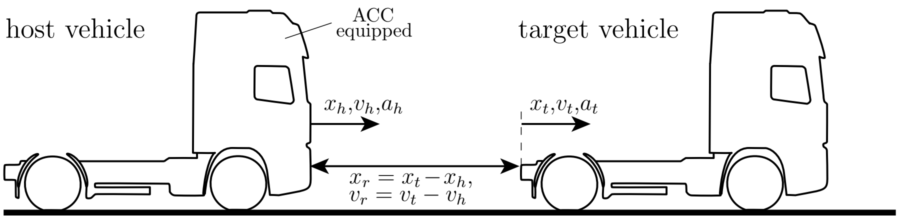


Figure 2.1: Main variables of an ACC system. x, v and a are the position, velocity and acceleration, with the subscript h and t indicating the following host vehicle respectively the target vehicle. x_r and v_r represents the relative distance and relative velocity between the two vehicles.

The relative distance x_r between the target vehicle and the host vehicle is formulated as

$$x_r = x_t - x_h \tag{2.1}$$

with x_t the position of the target vehicle, x_h the position of the host vehicle.

The relative velocity is the velocity difference between the target vehicle v_t and the host vehicle v_h

$$v_r = v_t - v_h \tag{2.2}$$

ACC uses a radar for detecting the presence of a predecessor and measuring the relative distance x_r and the relative velocity v_r between the two vehicles. ACC controls the velocity and distance of the host vehicle automatically to safely follow a target vehicle with a possibly varying velocity intended to reduce driver workload. Hence, concerning the situation of following a predecessor, cruise control can be regarded as open-loop control, whereas ACC proposes a closed-loop solution. Primarily, ACC is marketed as a comfort system and not as a safety system, since the braking capacity is usually limited to around $-3\,\mathrm{m\,s^{-2}}$. ACC can therefore not be regarded as a collision avoidance system. Collision avoidance systems need higher decelerations, in the order of $-8\,\mathrm{m\,s^{-2}}$. In emergency situations, the driver has to take over and remains fully responsible for the vehicle maneuvering [55]. The driver is able to over-ride the system at any time by activating the brake or accelerator pedal and can switch it on or off at different stages of the journey. Secondarily, ACC is likely to have a positive effect on traffic safety and economy because an ACC may maintain a perfect, short and still safe headway to the vehicle in front. By means of smoother accelerations and deceleration profiles, the system potentially contributes to a more stable traffic flow and possibly enables driving at shorter headways and higher velocities, which increase roadway capacity (time headway of $1.2\,\mathrm{s}$ or less) [28, 53].

The 'Stop-&-Go' functionality is an extension of ACC. To improve driver comfort, also velocities below $30\,\mathrm{kn\,h^{-1}}$ can be handled (ACC systems function above velocities of $30\,\mathrm{kn\,h^{-1}}$). In traffic jams or slowly moving traffic, ACC Stop-&-Go enables following a target vehicle and automatically stop behind that target. ACC Stop-&-Go will also accelerate the host vehicle when the target vehicle moves again. There is no minimum velocity and the system can recognize stationary vehicles.

ACC systems are currently available in commercially vehicles as well as in trucks. Stop-&-Go functionality is present since 2006 in the Mercedes-Benz S-Class and Audi Q7. Future development of ADAS will lead to ACC systems with, for example, sensor fusion: a vision camera is added as an extra sensor. Alternatively, communication between vehicles can be added to improve situation awareness: Coöperative Adaptive Cruise Control (CACC).

2.1.2 Driver related aspects

This section introduces some general driver related aspects, such as acceptance and workload, available in literature about the use of ACC systems. Next, two performance criteria or metrics for evaluating ACC systems are introduced: comfort and desirability. Through measurements of these metrics, different ACC controllers can be compared.

Additional value of ACC systems

In general, the additional value of ACC systems depends on different variables. Acceptance, workload, driving comfort, traffic flow, reliability and Human Machine Interface (HMI) are important factors, which can affect the success of ACC. ACC systems are experienced as profitable, comfortable, useful and enjoyable. Research shows that drivers are less likely to use ACC (without Stop-&-Go functionality) in heavy traffic and are more likely to engage it when driving in the fog, driving at night on an unlit highway, driving for longer periods of time, driving in low density traffic, and driving on an unknown road network [37]. Negative effect is decreased alertness of the driver, which possibly results in an inadequate reaction in critical situations [29]. To access driver acceptance of ACC, [28] has conducted a questionnaire study and two driving simulator studies. It was found that shorter headway times increase the traffic flow, however close following distances can be stressful for some drivers. This concludes that an adjustable headway is necessary to meet interests of different drivers. In some rural road driving scenarios, such as passing other cars or driving on curved roads, it was observed that ACC could be more dangerous than helpful. To avoid such problems, the driver needs to understand the limits of ACC and learn to disengage the system when required. Based on results of their research, [28] concludes that a well-designed ACC system can improve the driving comfort and also harmonizes the traffic flow as it reduces velocity variations.

Another simulator study [47] studied the driver workload with ACC. The investigators observed reduced mental workload when ACC is engaged, as the driver is relieved from some of the decision-making elements of the driving task. The possibility exists that, due to lower levels of workload, the driver will be out of the vehicle control loop. Similar to other human supervisory tasks, reduced levels of attention associated with lower levels of workload may affect the ability of the driver to maintain awareness of the status of the system. [47] believe that inter-vehicle spacing for ACC mode should be larger than the manual mode to provide the drivers with enough time to reclaim the control of the vehicle in an emergency scenario. [47] also suggest that more attention is required on the driver interface of the ACC system to help keeping the drivers in the control loop. Such an interface would help the driver to develop appropriate internal mental representations that will enable him to understand the limitations [4]. Such a user-centered design will satisfy the user needs resulting in a high level of product acceptance.
[23] emphasizes that safe and effective ACC design requires that the operational limits of ACC should be detectable and interpretable by human drivers. They count four basic factors for safe operation of ACC: 1. The dynamic behaviour of the ACC system should be predictable by the driver. 2. The ACC should decrease physical workload without placing unrealistic demands on attentional management and human decision-making. 3. The transfer of authority between automation and human should be seamless. 4. The operational limits of ACC performance should be easily detectable and identifiable.

Performance metrics

Regarding the driveability of an ACC equipped vehicle, a distinction can be made between the comfort and the desirability of a driving action [43]. In the next subsections, these two metrics are presented.

Comfort
The comfort of a driving action is strongly related to the actual acceleration and jerk levels, which define the smoothness of the driving action. The smoothness is measured by somatosensorily and vestibularly detectable variables. The sense of touch is mediated by the somatosensory system. Touch may be considered one of five human senses: when a person touches something or somebody, this gives rise to various feelings: the perception of pressure. An example of this perception of pressure is motion. As our movements consist of rotations and translations, the vestibular system comprises two components: the semicircular canals of the inner ear, which indicate rotational movements and the Otolith organs, which are sensitive to gravity and acceleration in translation movement. For this ACC Stop-&-Go application only the translation movements are relevant. [24] presents experiments which show that both jerk and acceleration contribute significantly to the perceived strength of motion. Distinction between acceleration and jerk is difficult. However, noticing jerk levels when starting to brake can be desirable and feel safe. In this way, the driver is informed the ACC system reacts on the preceding vehicle. Measurement of acceleration and jerk levels is an objective measurement.

Concerning ride comfort as experienced by people in or on a vehicle, the measurable motion environment contain, besides translational accelerations, also vibration shocks [48]. Frequency analysis can deliver information about the vibrational comfort. The frequency range $3 - 20$ Hz contains the eigenfrequenties of the human body [35]. For example: the body/shoulder combination has a eigenfrequency around $3-5$ Hz, the stomach around $4-5$ Hz, the heart at 7 Hz and the head at 20 Hz. For a comfortable driving action, the magnitude for this specific frequency range must be low.
Vibrational comfort, experienced by a human being, is analyzed by spectral estimation. The goal of spectral estimation is to describe the distribution (over frequency) of the power contained in a signal, based on a finite set of data. To achieve this, a Power Spectral Density (PSD) analysis is executed. The unit of the PSD is energy per unit of frequency. Integration of the PSD with respect to frequency yields the average energy. In this case, a nonparametric method is used. Nonparametric methods are those in which the PSD is estimated directly from the signal (here, host acceleration) itself [38]. For this PSD analysis, the Welch's method is used. This method consists of dividing the time series data into (possibly overlapping) segments, computing a modified periodogram of each segment and then averaging the PSD estimates. A periodogram is defined as the squared magnitude of the discrete-time Fourier transform of the samples of the process [38].

Desirability
The desirability of a driving action is strongly related to the actual traffic situation. This is mainly observed by perceptional variables, such as visual inputs. The use of these perceptional variables such as the time headway t_{hw} and the time-to-collision (TTC), rather than the headway distance or relative velocity, is more effective in analyzing driving behaviour characteristics when following a preceding vehicle [57, 27]. Time headway is already introduced in Section 1.3.

Time to collision (TTC) is defined as the time required for two vehicles to collide if they continue at their present velocity and on the same path

$$\mathrm{TTC} = \frac{x_r}{v_r} \tag{2.3}$$

When approaching a target vehicle, a minimum, i.e. TTC_{min} is attained. In principle, the lower the TTC_{min}, the higher the risk of a collision has been. In literature, this variable is often pointed out to form objective metrics for desirability of the corresponding driver behaviour [54].

Desirability can be contrary to comfort. For example, acceleration and jerk levels can be high (uncomfortable) in a possible critical situation of an approach of slower driving vehicle. However, this is desirable behaviour in order to prevent a collision. Safety perception is included in desirability. Dangerous (i.e. unsafe) situations are defined as situations in which the driver might doubt the system to be capable to solve the situation in a safe and smooth way.
Desirability is a subjective measurement. The actual time to collision and the time headway can be measured (and are thus objective measurements). However how critical or unsafe the situation is (different levels for TTC_{min} and t_{hw}), according to the driver, depends on the situation.

2.1.3 TNO Adaptive Cruise Control Stop-&-Go

In this section the TNO ACC Stop-&-Go is discussed. TNO Automotive has created a Car Laboratory (CarLab) for the development of advanced driver assistance and longitudinal control. For this purpose, a vehicle is equipped with several sensors for vehicle dynamics measurements and an interface for electronic braking and throttle. This CarLab, with a functional ACC Stop-&-Go system using a Groeneveld/RoadEye Forward Looking Radar, is used for road experiments. The vehicle is equipped with a DSPACE AUTOBOX computer for rapid control prototyping to provide a control system which can control the engine and the brakes of the vehicle.
The ACC software consists of [45]:

1. ACC S&G control
   Depending on the radar data and cruise control velocity requested by the driver, a desired vehicle acceleration / deceleration is calculated. The settings of the controller determine the ACC behaviour. The controller is independent of the vehicle.

2. Ax control
   The longitudinal acceleration control converts the desired acceleration from the ACC S&G controller to engine and brake setpoints. This part of the controller is specific for the type of vehicle.

Main ACC control algorithm
The output of this main controller consists of two accelerations. *ACCcruising* is the desired acceleration to be realized by the engine. *ACCbraking* is the acceleration to be realized by the brakes. This separation is introduced to create a more comfortable driving behaviour when approaching a predecessor. The desired distances to the predecessor ($x_{r,d,max}$ and $x_{r,d,min}$) depend of the vehicle velocity v_h and the defined time headway.

$$x_{r,d,max} = t_{hw} \cdot v_h + x_{r,0} \tag{2.4}$$
$$x_{r,d,min} = (t_{hw} - t_{hw,\text{brake difference}}) \cdot v_h + x_{r,0} \tag{2.5}$$

The parameter $t_{hw,\text{brake difference}}$ causes a small difference in time headway: $x_{r,d,min}$ is always smaller than $x_{r,d,max}$. The driver can choose between three preset values of the time headway with the use of a button on the dashboard. The parameter $x_{r,0}$ is introduced to maintain a minimum distance at a $v_h = 0$. This is necessary to avoid collisions in Stop-&-Go situations. *ACCcruising* and *ACCbraking* are PD controllers with the following equations:

$$ACCcruising = k_{ACC}(k_{x_r} \cdot (x_r - x_{r,d,max}) + k_{v_r} \cdot v_r) \tag{2.6}$$

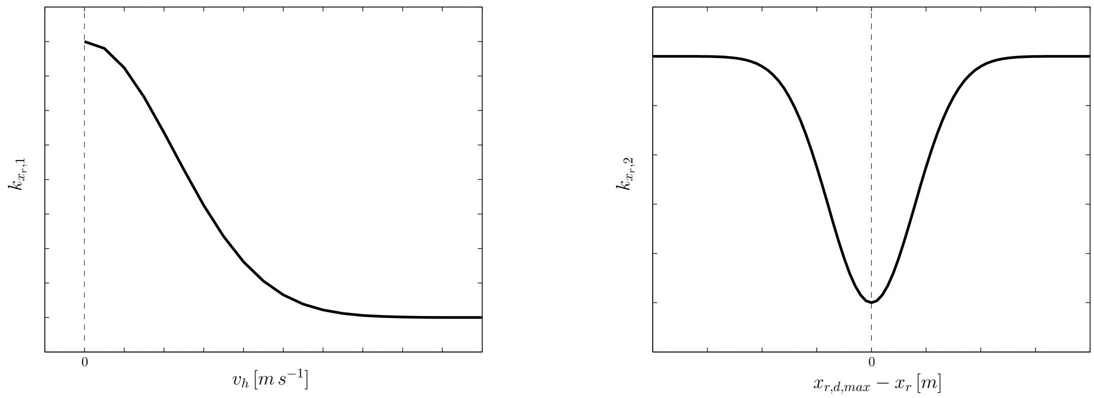
$$ACCbraking = k_{ACC}(k_{x_r} \cdot (x_r - x_{r,d,min}) + k_{v_r} \cdot v_r) \tag{2.7}$$

x_r and v_r are the outputs from the radar and represent the distance and velocity difference with respect to the target vehicle (target quantity - host quantity). k_{ACC} is the overall gain of the main ACC control algorithm. A higher value for k_{ACC} will result in higher control signals, and therefore in safer, but less comfortable ACC behaviour. k_{ACC} is constant. k_{x_r} is the P action: this gain is tuned to regulate the distance error to zero ($x_r - x_{r,d,max} = 0$ for *ACCcruising* and $x_r - x_{r,d,min} = 0$ for *ACCbraking*). The D action of this PD controller is represented by k_{v_r}, the regulated error is v_r. Because *ACCcruising* is smaller than *ACCbraking* (due to different desired distance), it takes some time before the brakes are activated after the throttle is released.

k_{x_r} and k_{v_r} are adaptive parameters. k_{x_r} is the gain for the distance error and could therefore be compared to a spring constant. The value is dependent of the vehicle velocity ($k_{x_r,1}$) and the distance error ($k_{x_r,2}$). The overall value of k_{x_r} is obtained by multiplying $k_{x_r,1}$ and $k_{x_r,2}$: $k_{x_r} = k_{x_r,1}(v_h) \cdot k_{x_r,2}(x_{r,d,max} - x_r)$.

- $k_{x_r,1}$
  Reaching the desired distance is important for correct Stop-&-Go behaviour (low velocities). At high velocities, smooth behaviour is more important. Taking these criteria into account results in a dependency of $k_{x_r,1}$ for the vehicle velocity as can be seen in Figure 2.2(a).

- $k_{x_r,2}$
  Once the desired distance is reached, smooth behaviour is more important than correcting disturbances in distance error. When the truck is fully loaded, only a small desired acceleration results in large throttle requests. Therefore, $k_{x_r,2}$ is small in case of small distance errors. Figure 2.2(b) shows the resulting dependency of $k_{x_r,2}$ as function of the distance error.



(a) Dependency of $k_{x_r,1}$ for the host velocity

(b) Dependency of $k_{x_r,2}$ for the distance error

Figure 2.2: $k_{x_r,1}$ and $k_{x_r,2}$

- k_{v_r}
  For the same reason as the dependency of $k_{x_r,2}$ for distance error, the value of k_{v_r} is small for small distance errors and large distance errors. See Figure 2.3.

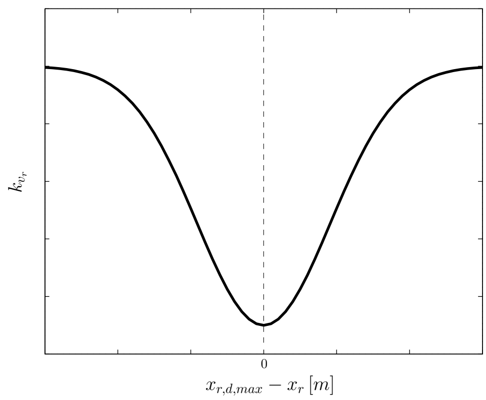


Figure 2.3: Dependency of k_{v_r} for the distance error

This main ACC control algorithm deals with the general situations: following a predecessor. For special situations, the main algorithm Eq. (2.6) & Eq. (2.7) is adjusted by adapting parameters, gains and/or errors. Examples of these situations are (not complete): 1. Cruise control: the goal of the CC controller is reaching the cruise control velocity v_{CC} prescribed by the driver. In normal conditions, CC only controls the throttle. So when the CC velocity is smaller than the real velocity the vehicle won't use the brakes to decrease the velocity. The maximum deceleration is reached when the throttle is fully released. 2. Positive v_r handling: if the predecessor is driving away from the host vehicle (positive v_r), there is no reason for activating the brakes. In that case, *ACCbraking* is set to zero. 3. Full stop: Eq. (2.6) and Eq. (2.7) do not suffice for Stop-&-Go situations. This is due to the fact that the controller wants to reach the desired headway. This causes unwanted acceleration if the vehicle has stopped behind a stopped predecessor, and the descried headway just hasn't been reached. 4. Sharp curve MIO loss: the radar isn't able to track objects (Most Important Object (MIO) loss) in sharp curves (radius R smaller than 50 m) due to the limited opening angle.

As already stated in Section 1.2, the final tuning of the ACC Stop-&-Go controller has become rather complicated and time consuming. This is due to the fact that the parameters $k_{x_r,1}$, $k_{x_r,2}$ and k_{v_r} are state independent: these parameters are used in the main algorithm Eq. (2.6) & Eq. (2.7) to deliver a desired acceleration for all situations. Furthermore, by adapting these parameters in one state to affect the performance positively, desired behaviour in other states can be affected negatively.

2.2 Control strategies for ACC

In order to design an easy-to-tune and vehicle-independent control strategy for ACC applications, different control strategies / frameworks have been studied and compared. In this section these strategies are discussed. The goal is to choose an appropriate framework for ACC applications.

As ACC takes over control from the driver, it should resemble driver behaviour to some extent. A driver however situation-dependent behaviour. Furthermore, constraints arising from subjective specifications of desirable behaviour are present. Because of the need to constraint e.g. the control output, classical control, such as Proportional-Integral-Derivative (PID) controllers, are inadequate.
The use of neural networks or fuzzy control is assumed to model the human driver behaviour from a psychological point of view perfectly [11]: using measured data to train artificial neural networks might be a suitable access to this problem, but this approach requires a couple of measurements for all different driving situations and involves a higher computation effort than other techniques.
A linguistic formulation for the comfort requests by the driver using fuzzy logic means a good access with less effort to a nonlinear controller design. This technique has the disadvantage that the generated controller is only valid for one specific driver-vehicle-load combination. Due to the variety in performance metrics, corresponding controller design and tuning is rather complex, these theories are not easy accessible and hardly employed in practice.

Parameter scheduling or gain scheduling may be used in cases of moderate nonlinearities. This method is based on a linearization around several operating points and the controller is designed by fulfilling certain design requirements in each operating point. This technique is suitable for every single operating point but not for dynamical change of operating points as for example during the change of the desired velocity in cruise control. Applying parameter scheduling by hand can cause stability and feasibility problems.

The most popular approaches for designing controllers for linear systems with constraints fall into two categories: anti-windup control and Model Predictive Control (MPC). Anti-windup schemes are widely used in practice because in most Single-Input-Single-Output (SISO) situations they are simple to design and work adequately. MPC has become the accepted standard for complex constrained multivariable (Multi-Input Multi-Output or MIMO) control problems in process industries. The main idea of MPC is to choose the control action by repeatedly solving on-line an optimal control problem. This aims at minimizing a performance criterion over a future horizon and yields an optimal control sequence, possibly subject to constraints on the manipulated inputs and outputs, where the future behaviour over a specified time horizon is computed according to a model of the plant. Closed-loop MPC performs an optimization in every time-step, yielding state or situation dependent control. Closed-loop MPC can deal with stability and feasibility issues. Stability can be fulfilled by defining a weight on the terminal state (the end state of the prediction horizon of MPC), implemented in the performance criterion, or analyzed afterwards. Analyzing feasibility results in a subset of the MPC controller for which constraints satisfaction is guaranteed.

The big drawback of MPC is the on-line computational effort that is associated with it. This limits its applicability to relatively slow processes. However, the range of applicability of MPC can be increased largely by moving the computational effort off-line while preserving all its other characteristics. This is done by using multi-parametric programming techniques, presented in [10]. The solutions obtained in this way take the form of a piecewise affine (PWA) state feedback law. In particular, the state-space is partitioned into polyhedral sets and for each of those sets the optimal control law is given as an affine function of the state. Such controllers can then easily be implemented on-line in the form of a look-up table. Moreover, such an explicit form of the controller provides additional insight for better understanding of the control policy of MPC.

Application of closed-loop MPC to a linear system yields a hybrid system [8]. Hybrid systems combine the continuous behaviour evolution specified by differential equations with discontinuous changes specified by discrete event switching logic. In other words: hybrid dynamical models describe systems where both analog (continuous) and logical (discrete) components are relevant and interacting. See Appendix A for more details on hybrid systems. Due to the equivalence of hybrid systems (such as Piecewise Affine (PWA) systems, Mixed Logical Dynamical (MLD) systems and Linear Complementarity (LC) systems, see Appendix A), proofs regarding stability and feasibility originating from other hybrid systems are applicable for closed-loop MPC as well [7].

Disadvantages of explicit MPC are the rapid growth of size in the explicit solution as the problem size increases. Furthermore, the look-up table is only valid for a specific configuration of the system. Instead of simple adaption of parameters (e.g. in PD control), recomputation of the table is required, when one or more parameter(s) is/are changed.

Design difficulties as situation dependency, MIMO character and nonlinear behaviour in combination with constraints arising from specifications related to safety and comfort issues, makes the hybrid control framework (also called explicit MPC framework) a suitable choice for ACC applications [15].

2.3 Hybrid control in automotive applications

Hybrid systems have captured the attention of the research community in the past few years. Indeed, interesting theoretical results, as well as control applications have been reported in the literature in recent years. Automotive control has been the first field where hybrid systems have revealed their potential [1].

In the paper entitled 'Multi-object Adaptive Cruise Control' [41] an algorithm is proposed for solving a Multi-Object Adaptive Cruise Control problem. In a multi-object traffic scene the optimal acceleration is to be found respecting traffic rules, safety distances and driver intentions. The objective function is modeled as a quadratic cost function for the discrete time piecewise affine system. The optimal state-feedback control law is found by solving the underlying constrained finite time optimal control problem via dynamic programming. The PWA model exists of two different cost functions. The two cost functions are defined as 1. CC cost function and 2. ACC cost function.
Differences between this project and this paper are the additional infrared sensor (larger angle: up to 180°, shorter range), stereo vision systems (lane recognition) and the availability of the target acceleration.

Other examples of hybrid control in automotive applications are:

- Semi-Active Suspension [22]
- Traction Control [13]
- Direct Injection Stratified Charge Engines [21]
- Electronic throttle [3]
- Automotive Robotized Gearbox [5]
- Dry clutch engagement [31, 6]
- Magnetically Actuated Mass Spring Dampers for Automotive Applications [17]

These applications use models, which are characterized by nonlinear dynamics or as a switching piecewise affine system with tight performance specifications and/or constraints on inputs and/or outputs. Due to these reasons, MPC is an appealing control framework for such systems. A MPC controller, based on online quadratic optimization, is tuned and the effectiveness of the approach can be shown through simulations. Then, using an offline multiparametric optimization procedure, the controller is converted into an equivalent explicit piecewise affine form that is easily implementable in an automotive microcontroller through a lookup table of linear gains. In this way, an explicit MPC or a Hybrid control framework is designed. Experiments show that good (and robust) performance is achieved in a limited development time by avoiding the design of ad hoc supervisory and logical constructs usually required by controllers developed according to standard techniques.

The ever increasing demands on passengers' comfort, safety, emissions and fuel consumption demands automotive control systems of increasing complexity. Because of this, hybrid control can be seen as a possible control framework to design, simulate, test and implement high-performance automotive control systems for a variety of automotive problems.

Chapter 3

Hybrid Adaptive Cruise Control

In this chapter the design of a hybrid controller is explained. Based on the previous chapter, a MPC controller is chosen as a control framework and the ACC related control problem is embedded into this framework.

3.1 Introduction

In Chapter 2, MPC is adopted as a suitable framework for ACC applications. In this chapter this framework is applied to the ACC control problem. First, a model describing two vehicles is presented, in continuous as well as in discrete time. After that, the control problem is explained: control objectives, requirements, constraints and system constraints are determined. This leads to a MPC control problem, which is solved explicitly to obtain a state feedback solution which is continuous and piecewise affine. After that, stability and feasibility analysis have been carried out. Finally, the implementation in MATLAB with the aid of the MPT toolbox is explained.

3.2 Model predictive control

As already stated in Section 2.2, MPC can deal with open-loop unstable, constrained and/or multivariable control problems. Instead of conventional control which uses a pre-computed control law, MPC calculates the control action by repeatedly solving on-line an optimal control problem, see Eq. (3.1). This aims at minimizing a performance criterion over a future horizon and yields an optimal control sequence, possibly subject to constraints on the manipulated inputs and outputs, where the future behaviour over a specified time horizon, is computed according to a model of the plant. The performance criterion shown in Eq. (3.1) is a so-called Quadratic Program (QP), because of the presence of the state vector x and the input u, both quadratically. The optimization problem containing the performance criterion and the possible constraints is solved at each time k (discrete time), where $x_{k+t|k}$ denotes the predicted state vector at time $k+t$ obtained by applying the input sequence $u_k, \cdots, u_{k+t-1}$ to a linear time-invariant model (A, B, C) starting form the state $x(k)$. The weightings Q and R enable tuning of the optimization problem. The different variables of the state vector can be weighted with elements in Q. The control output is weighted by assigning R.

$$
\begin{aligned}
\min_{U \triangleq \{u_k,\dots,u_{k+N_u-1}\}} & \left\{ J(U,x(k)) = \sum_{k=0}^{N_y-1} \left[x_{k+t|k}^T Q x_{k+t|k}\right] + \sum_{k=1}^{N_u} \left[u_{k+t}^T R u_{k+t}\right] \right\} \\
\text{subject to} \quad & y_{min} \leq y_{k+t|k} \leq y_{max}, \quad t = 0, \cdots, N_y - 1, \\
& u_{min} \leq u_{k+t|k} \leq u_{max}, \quad t = 1, \cdots, N_c, \\
& x_{k|k} = x(k), \\
& x_{k+t+1|k} = A x_{k+t|k} + B u_{k+t}, \quad t \geq 0 \\
& y_{k+t|k} = C x_{k+t|k}, \quad t \geq 0 \\
& u_{k+t} = K x_{k+t|k}, \quad N_u \leq t < N_y
\end{aligned}
\tag{3.1}
$$

This future behaviour is usually called the prediction horizon and is denoted by N_y as depicted in denoted in Eq. (3.1) and depicted in Figure 3.1. At each discrete-time instant k, the measured variables and the process model (linear, nonlinear or hybrid) are used to (predict) calculate the future behaviour of the controlled plant [8]. This is achieved by considering a future control scenario, which is usually called control horizon and is denoted by N_c (see Figure 3.1), as the input sequence applied to the process model, which must be calculated such that certain desired constraints and objectives are fulfilled. The first control in this sequence is applied to the plant. At the next time step, the computation of the optimization is repeated starting form the new state and over a shifted horizon, leading to a moving horizon policy.

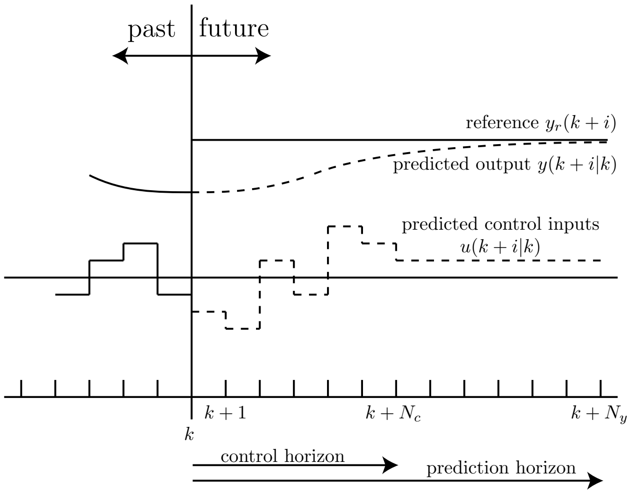


Figure 3.1: A graphical illustration of Model Predictive Control

The reader is referred to Appendix B for a detailed explanation of MPC and explicit MPC / hybrid MPC.

3.3 Model

Firstly, the continuous-time model based on the (translational) kinematic equations of motion is presented. Secondly, the corresponding discrete-time model is presented. This discrete-time model will be used in the (already in discrete-time discussed) MPC controller framework (Section 3.2).

3.3.1 Continuous-time model

Modeling the two vehicles, see Figure 2.1, with the already introduced relative distance $x_r(t)$, the relative velocity $v_r(t)$ and the host velocity $v_h(t)$ delivers the following MIMO Linear Time Invariant (LTI) model

$$\dot{x}_r(t) = v_r(t) \tag{3.2a}$$
$$\dot{v}_r(t) = a_r(t) \tag{3.2b}$$
$$\dot{v}_h(t) = a_h(t) \tag{3.2c}$$

with $a_r(t) = a_t(t) - a_h(t)$ the acceleration difference between both vehicles. Due to the unknown acceleration of the target vehicle, $a_t(t)$, only the known acceleration of the host vehicle, $a_h(t)$, is introduced and $a_t(t)$ is assumed to be zero. The acceleration of the target vehicle is unknown because of radar limitations: only the relative distance and relative velocity can be distinguished with a FMCW radar. The manipulated variable is the host acceleration. The measured variables are x_r, v_r and v_h respectively.

With the state vector $\overline{x}(t) = (x_r(t), v_r(t), v_h(t))^T$, the input or manipulated variable $u(t) = a_h(t)$ and the output vector $\overline{y}(t) = (x_r(t), v_r(t), v_h(t))^T$, the system $\mathcal{S}$ Eq. (3.2) can be written in a State-Space (SS) representation as:

$$\mathcal{S} : \begin{cases} \dot{\overline{x}}(t) & = & A\overline{x}(t) + Bu(t) \\ \overline{y}(t) & = & C\overline{x}(t) \end{cases} \tag{3.3}$$

with

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.4}$$

For simplicity, road friction and drag forces are not considered in this model. Due to the closed-loop controller structure, the effect of these frictions will be measured in the variable v_h (and v_r, x_r) and thus the effect of these variables will appear in the state vector $\overline{x}$.
This ACC-specific open-loop system is marginally stable: it has three poles at $0\,\mathrm{rad\,s^{-1}}$. As already stated in the previous section (Section 3.2), MPC can deal with unstable systems (thus also with systems which are marginally stable).

Controllability and observability

For the control of MIMO LTI models, it is important to know if it is possible to influence the system and thus the system's output. The system can be steered from a random initial state $\overline{x}_{t_0}$ to a random final state $\overline{x}_{t_e}$ in a finite time interval t_0 to t_e, if all elements of the state vector $\overline{x}$ can be influenced. This is called controllability. Controllability is relevant for the design of control systems and is defined as: a time-invariant linear system with input $u(t)$ and output $y(t)$ is said to be output controllable if for any y_1, y_2 an input $u(t)$, $t \in [t_1, t_2]$ exists with $t_1 < t_2$ that brings the output from $y(t_1) = y_1$ to $y(t_2) = y_2$. Consider a plant

$$G(s) = C(sI - A)^{-1}B \tag{3.5}$$

with l outputs and let r denote the normal rank of $G(s)$. In order to control all outputs independently we must require $r = l$ and the plant is said to be 'functionally controllable' [46, Definition 6.1].
For this ACC control problem the rank of G is $\mathit{rank}(G(s)) = 1$, which implies that the defined system

is not functionally controllable ($r < l$). There are $l - r = 2$ output directions, which cannot be affected. This is obvious, because only the host velocity can directly be controlled. The other state variables x_r and v_r are the result from the host velocity and host position together with the target position and target velocity. Although the target acceleration is assumed to be zero, the initial position and initial velocity of the target are unknown.

The system, is said to be observable if for any unknown initial state $\overline{x}(0)$, there exists a finite $t_0 > 0$ such that $\overline{x}(0)$ can be exactly evaluated over $[t_0, t_e]$ from the input u and the output y. Otherwise the system is said to be unobservable. If a system is not observable, this means the current values of some of its states cannot be determined through output sensors: this implies that their value is unknown to the controller and, consequently, that it will be unable to fulfil the control specifications referred to these outputs.
The observability matrix O_b is defined as

$$O_b = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{3.6}$$

When O_b has full column rank $\rho(G^o) = n$, the system is said to be obsevable.

The observability matrix O_b for this ACC control problem is

$$O_b = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{3.7}$$

The rank of O_b is 3, which is equal to the norm n of the system $n = 3$, thus this system is completely observable.

Summarizing, this MIMO LTI system can not be steered in all random states. The control input can only adjust the host velocity. Concerning observability, the system is observable, which implies the output of the system is known for a finite time interval (t_0 to t_e).

3.3.2 Discrete-time model

In this subsection, the discrete-time model is presented based on the continuous-time mode (Subsection 3.3.1). Discretizing the model $\mathcal{S}$ with 'Zero-Order Hold' (ZOH) with time step k yields the discretized model $\mathcal{S}_D$.

$$\mathcal{S}_D : \begin{cases} \overline{x}(k+1) &= A\overline{x}(k) + Bu(k) \\ \overline{y}(k) &= C\overline{x}(k) \end{cases} \tag{3.8}$$

with

$$A = \begin{pmatrix} 1 & T_s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} \frac{1}{2}T_s^2 \\ -T_s \\ T_s \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.9}$$

The argument k denotes the discrete time, corresponding to the continuous time $t = k \cdot T_s$ where T_s is the sample time. Just as the continuous time model, $a_t(t)$ is regarded zero.

3.4 Control problem

In this section, the performance metrics of Subsubsection 2.1.2 are translated into control objectives, requirements, constraints. All these items will adjust the cost function (Eq. (3.1)) for implementing ACC Stop-&-Go into the MPC control framework. The state boundaries are defined to limit the state-space which will be explored. This prevents unnecessary complexity of the MPC problem.

3.4.1 Control objectives

The control objectives for ACC are defined as:

1. Desired relative distance
   Typically, the desired relative distance is determined based on the so-called time-headway t_{hw}. t_{hw} is defined as the time needed for the host vehicle to reach the current position of the target vehicle if the host vehicle continues at its present velocity, $t_{hw} \triangleq x_r/v_r$. In order to create a desired time headway the desired distance $x_{r,d}(t_{hw})$ is used as a control objective
   $$x_r = x_{r,d}(t_{hw}) \tag{3.10}$$
   with
   $$x_{r,d} = x_{r,0} + v_h \cdot t_{hw} \tag{3.11}$$
   $x_{r,0}$ is the minimal distance between the two vehicles at $v_h = 0$. The driver is enabled to vary the time headway.

2. Desired relative velocity
   The second control objectives is to realize a zero velocity difference $v_h = v_t$, i.e. $v_{r,d} = 0$. v_t is considered to be the desired velocity trajectory to be tracked by the host vehicle v_h.

This yields the following control objectives $\mathcal{O}$

$$\mathcal{O} : \begin{cases} x_{r,d} &= x_{r,0} + v_h \cdot t_{hw} \\ v_{r,d} &= 0 \end{cases} \tag{3.12}$$

3.4.2 Constraints

The control objectives of the previous section must be reached in limited time, however considering comfort and desirability of the longitudinal movement, the acceleration and jerk should be limited.

1. Minimum acceleration
   ACC is a comfort system, which means that the accelerations are limited. ACC is not a collision avoidance system for which higher decelerations are necessary. Here, for the ACC application the minimum acceleration is limited with $a_{h,min} = u_{min} = -3.0\,\mathrm{m\,s^{-2}}$.

2. Maximum acceleration depending on host velocity
   This (linear) dependency of the acceleration on the host velocity is a comfort specification. At higher velocities the engine of the vehicle is not able or has difficulties generating high accelerations. To avoid this, the maximum acceleration is bounded. Because of the chosen MPC framework, a linearly dependent relation (dependent on the host velocity) must be chosen, here the relation $a_{h,max} = u_{max} = 3(1 - 0.025 \cdot v_h)\,\mathrm{m\,s^{-2}}$ is used to bound the maximum host acceleration. For the maximum host velocity $40\,\mathrm{m\,s^{-1}}$ is assumed. $v_h = 40\,\mathrm{m\,s^{-1}}$ results in $a_{h,max} = 0$.

3. Maximum and minimum jerk
   Out of comfort reasons, the absolute value of the jerk (or the change of accelerations) is limited by constraints (minimum and maximum). Based on simulations results (compared with the TNO ACC controller), this constraint is set to $|\delta u| = |j_h| = 5\,\mathrm{m\,s^{-3}}$.

4. Minimum relative distance
   The relative distance $x_r = y_1$ must be larger than a minimum relative distance $x_{r,0} = y_{1,min}$ in order to avoid a collision with the preceding vehicle, $y_{1,min} < y_1$.

Summarizing, the constraints $\mathcal{C}$ are defined as

$$\mathcal{C} : \begin{cases} u_{min} \leq u(k) \leq u_{max}(v_h) \\ |j_h(k)| \leq j_{h,max} \\ y_{1,min} < y_1(k) \end{cases} \tag{3.13}$$

3.4.3 Requirements

Concerning comfort, the definition of limitations on the acceleration and the jerk of the longitudinal movement (see previous subsection) is not enough. The change of jerk should be also limited. For implementation into the control framework, the next requirement is defined:

- Weight on jerk
  If changes in the control signals are penalized via a weight on the jerk, rapid changes in the control signals become expensive and are therefore, as long as possible, avoided. Penalizing changes in the control signal can be introduced in the MPC framework by extending the model. This new model will presented in the next section.

3.4.4 State boundaries

State boundaries on states limit the state-space that will be explored. These boundaries do not impose any constraints on predicted states, in contrary to the constraints of Subsection 3.4.2.

1. Minimum and maximum host velocity
   For ACC applications, only the forward longitudinal movement of the host vehicle is considered. The maximum v_h will be the maximum velocity of the host vehicle $v_{h,max}$.

2. Minimum and maximum target velocity
   For ACC applications, only the forward longitudinal movement of the target vehicle is considered $0 \leq v_t \leq v_{t,\max}$ and it is assumed that the target vehicle is driving in the same direction as the host vehicle. Assumed is that the maximum velocity of the target vehicle is equal to the maximum velocity of the host vehicle $v_{t,max} = v_{h,max}$.

3. Bound on relative velocity
   The velocity of the target vehicle is $0 \leq v_t \leq v_{t,\max}$. If the target vehicle is standing still (target velocity is zero), the relative velocity $v_r = v_t - v_h$ results in $v_r = -v_h$. This results in the state boundary $v_r \geq -v_h$.

4. Maximum relative distance
   The maximum value for $x_{r,max} = x_{rr}$ is the maximum range of the radar.

The state boundaries $\mathcal{B}$ are now summarized by

$$\mathcal{B}: \begin{cases} 0 < x_r \leq x_{rr} \\ 0 \leq v_h \leq v_{h,max} \\ 0 \leq v_t \leq v_{t,max} \\ -v_h \leq v_r \leq v_{t,max} \end{cases} \tag{3.14}$$

In Figure 3.2, the control objectives (No. 1 and 2), constraints (only No. 4) and the state boundaries are depicted (No. 1 to No. 4).

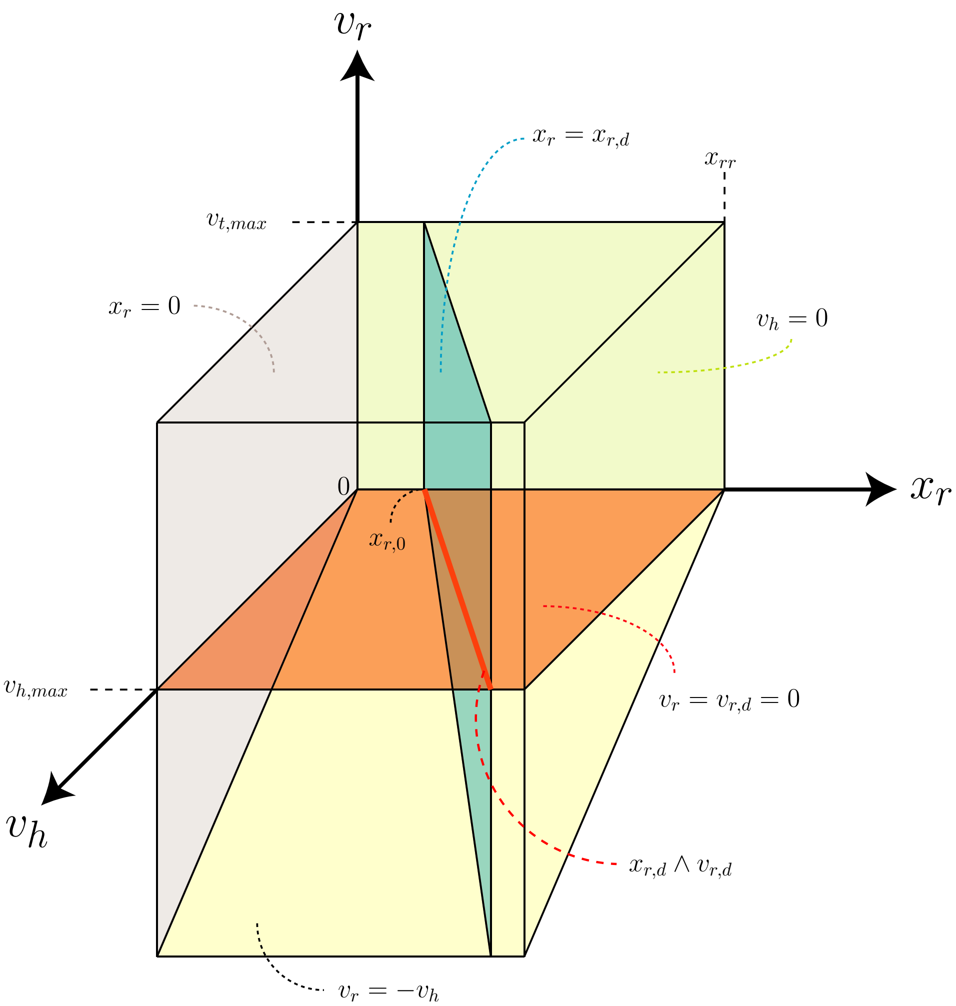


Figure 3.2: Visualization of the 3D state-space $\overline{x}(t)$: the control objectives (Eq. (3.12)), constraints (Eq. (3.13), only No. 4) and state boundaries (Eq. (3.14)) are depicted.

3.5 Model predictive control for ACC

3.5.1 Cost criterion

In this section, the introduced model of Section 3.3 together with the control problem of the previous section are implemented into the MPC framework.

Considering the control objectives $\mathcal{O}$ (Eq. (3.12)), the corresponding tracking error is defined as

$$\overline{\varepsilon}(k) = \overline{r}(k) - \overline{y}(k) \tag{3.15}$$

with $\overline{r}(k) = (x_{r,d}(k), 0, v_h(k))^T$.

Based on the introduced requirement (Subsection 3.4.3) and the constraints on the derivative of the control signal (3rd item of Subsection 3.4.2), adjusting the model is necessary. Hence, extension of the state vector with the previously implemented control value $u(k-1)$ results in $\overline{x}_e(k) = (x^T(k), u(k-1))^T$. The state space with this extension, by including an integral action, delivers

$$\mathcal{S}_{D,e} : \begin{cases} \dot{\overline{x}}_e(k) & = & \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \overline{x}_e(k) + \begin{bmatrix} B \\ I \end{bmatrix} \delta u(k) \\ \\ \overline{y}(k) & = & \begin{bmatrix} C & 0 \end{bmatrix} \overline{x}_e(k) \end{cases} \tag{3.16}$$

This extension results in a different output of the MPC controller: the derivative of the control output signal: δu. Now, constraints and a weight can be defined on this derivative of the control output signal. The $u(k)$ is now defined as the addition of derivative of the control output δu and the previous implemented control output $u(k-1)$

$$u(k) = u(k-1) + \delta u(k) \tag{3.17}$$

The reference vector will be $\overline{r}_e(k) = (r^T(k), u_d(k-1))^T$ and corresponding extension of the output and the error vector to $\overline{y}_e(k)$ respectively $\overline{\varepsilon}_e(k)$. Considering the predicted tracking error $\overline{\varepsilon}_e(k+n|k)$, at $n \geq 0$ time steps in future, the time-varying quadratic cost criterion becomes

$$J(\Delta\overline{U}(k), \overline{\xi}(k)) = \sum_{n=1}^{N_y} \left[\overline{\varepsilon}_e^T(k+n|k)\, Q\, \overline{\varepsilon}_e(k+n|k)\right] + \sum_{n=0}^{N_u-1} \left[\delta u^T(k+n)\, R\, \delta u(k+n)\right] \tag{3.18}$$

With the column vector $\Delta\overline{U}(k) \triangleq (\delta u(k), \ldots, \delta u(k+N_u-1))^T$ and $\overline{\xi}(k) \triangleq (\overline{x}_e^T(k), \overline{r}_e^T(k))^T$ a column vector comprising the variables needed to compute the estimate $\overline{\varepsilon}_e(k+n|k)$.

Assume that a full measurement of the state $\overline{x}_e(k)$ of the system $\mathcal{S}_{D,e}$ (Eq. (3.16)) is available at the current time k. The MPC optimization problem is then formulated as

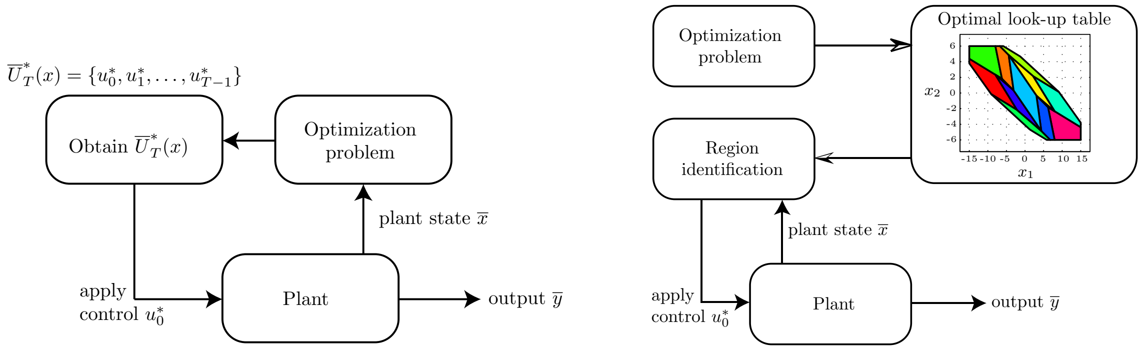
$$\min_{\Delta\overline{U}(k)} \left\{ J(\Delta\overline{U}(k), \overline{\xi}(k)) \right\} \tag{3.19}$$

subject to the system and the constraints $\mathcal{C}$ defined in Eq. (3.13) in which the states and output at time $k+n$ are replaced by the predictions $k+n|k$, which are based on $\overline{\xi}(k)$, with Q, R the weightings on the tracking error respectively the derivative of the control output and N_y, N_u the output respectively the control horizons, with $N_u \leq N_y$; for $N_u \leq n < N_y$ the control signal is kept constant, i.e. $\delta u(k+n) = 0$. Furthermore, $\bar{\varepsilon}_e(k|k) = \bar{\varepsilon}_e(k)$ and $u(k+n) = u(k+n-1) + \delta u(k+n)$, for $t \geq 1$. The computed optimal $\delta u^*(k)$ is used to compute the new control output $u(k) = u(k-1) + \delta u^*(k)$, which is applied to the system (Eq. (3.16)), after which the optimization (Eq. (3.19)) is computed again.

3.5.2 Explicit MPC

In this section, the MPC structure is made explicit, as already introduced in Section 2.2.

Due to on-line optimization, the applicability of MPC has been limited to slow systems, such as chemical processes, where large sampling times allow to solve large optimization problems each time new measurements are collected from the plant (see Figure 3.3(a)). The optimization problem can be solved offline (see Figure 3.3(b)) for all the expected measurement values through multiparametric (mp) solvers, see [2], [12], [25] and [50]. This is due to the possibility of stating constrained MPC problems as multi-parametric Quadratic Program (mpQP) programs, which has allowed computationally efficient explicit solutions to problems which previously required computationally demanding real-time optimization [49].



(a) Online MPC (b) Offline MPC

Figure 3.3: Online MPC versus offline MPC (explicit MPC)

In order to start solving the problem, an initial vector $\overline{x}_0$ inside the polyhedral set $\mathcal{X} = \{x : H\overline{x} \leq S\}$ of parameters is needed over which the problem can be solved such that the QP problem Eq. (3.18) is feasible for $\overline{x} = \overline{x}_0$. A good choice for $\overline{x}_0$ is the center of the largest ball contained in $\mathcal{X}$ for which a feasible z exists. For this ACC application, $\overline{x}_0 = \overline{x}_c$ is characterized with: 1) center $\overline{x}_c = [22.5, 17.5, 17.5]^T$ 2) radius $r_c = 22.5$. The reader is referred to Section B.4 for the calculation of this $\overline{x}_c$. For this center point $\overline{x}_0 = \overline{x}_c$, the mpQP is solved to find the optimal set of active constraints $\mathcal{A}$.

The result is a linear state feedback solution F_1 with a constant part g_1

$$\delta u(k) \quad = \quad F_1\,\overline{x}_e(k) + g_1, \tag{3.20}$$

The optimal solution for the calculated $\overline{x}_c$ is

$$F_1 = \begin{pmatrix} 0.0003 & 0.0012 & -0.0008 & -0.0021 \\ 0.0001 & 0.0005 & -0.0003 & -0.0011 \\ 0.0577 & 0.2309 & -0.1442 & -0.9975 \end{pmatrix}, \quad g_1 = \begin{pmatrix} -0.0016 \\ -0.0006 \\ -0.2885 \end{pmatrix} \tag{3.21}$$

Due to the used prediction horizon $N_u = 3$ both F_1 and g_1 contain 3 rows. Each row shows the optimal solution for the time step $n = 1, 2, 3$. Then the solution and the critical region CR_0 corresponding to $\mathcal{A}$ are characterized. Again, for the calculated $\overline{x}_c$ the next presented critical region $CR_0 = R_1$

$$R_1 : H_1 \, \overline{x}_e(k) \leq S_1 \tag{3.22}$$

with

$$H_1 = \begin{pmatrix} -0.0557 & -0.223 & 0.1393 & 0.9632 \\ 0.0557 & 0.223 & -0.1393 & -0.9632 \\ -0.1189 & -0.4757 & 0.2973 & 0.8192 \\ 0.1189 & 0.4757 & -0.2973 & -0.8192 \\ -1 & -0.001 & 0 & 0 \\ -1 & -0.002 & 0 & 0 \\ -1 & -0.003 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -0.7071 & -0.7071 & 0 \end{pmatrix}, \quad S_1 = \begin{pmatrix} -0.2737 \\ 0.2834 \\ 1.3124 \\ 2.5017 \\ 0 \\ 0 \\ 0 \\ 180 \\ 40 \\ 0 \\ 0 \end{pmatrix} \tag{3.23}$$

is the result for which Eq. (3.20) is optimal. The critical region CR_0 is a polyhedron in the $\overline{x}$-space and represents the largest set of $x \in \mathcal{X}$ such that the combination of active constraints at the minimizer remains unchanged. A polyhedron is a convex, geometric object which is bounded with flat faces and straight edges.

The state-space $\mathcal{X} \in \mathbb{R}^{n_x}$ with $n_x = dim(\overline{x}_e)$ is now further iteratively divided into $CR^{rest} = \mathcal{X} - CR_0$ polyhedral regions $R_i(\overline{x}_e)$, $i = 1, \dots, R$. For each new critical region R_i an optimal control law (consisting of values for F_i and g_i) is computed.

The overall result is stored in a look-up table, with $F(R_i, \overline{x}_e(k)) \in \mathbb{R}^{R \times 4}$ and a constant part $g(R_i) \in \mathbb{R}^{R \times 1}$. With this setup, a state feedback solution is obtained which is continuous and piecewise affine:

$$\begin{aligned} \delta u(k, R_i) &= F_i \, \overline{x}_e(k) + g_i, \quad \text{for} \\ \overline{x}(k) &\triangleq \{\overline{x}_e : H_i \, \overline{x}_e \leq S_i\}, \quad i = 1, \dots, R \end{aligned} \tag{3.24}$$

The system input $u(k)$ is already defined in Eq. (3.17). With this explicit MPC or hybrid controller the state will be driven towards a state, with a corresponding lower value of the designed cost function. In other words: the manipulated variable will respond on the current output variables and the future prediction output responses, both to minimize the differences between the predicted and the desired output variables. This controller can now be applied off-line enabling fast-sampling MPC for the ACC Stop-&-Go application. For this application, the sample frequency is 1000 Hz.

3.5.3 Tuning

The parameters N_y, N_u, Q and R are the basic MPC tuning parameters.

The weight matrices $Q \in \mathbb{R}^{4\times4}$ and $R \in \mathbb{R}^{1\times1}$ can be freely chosen as tuning parameters to give a trade off between performance and ride comfort and to obtain smooth natural vehicle behaviour. Q is the error weighting matrix, containing weights on the relative distance error, relative velocity error, error on host velocity and the previous control output. R is the control weighting matrix.

$N_y \in \mathbb{R}^{1\times1}$ and $N_u \in \mathbb{R}^{1\times1}$ define how large the prediction and control horizon are in the cost function (Eq. (3.18)). N_u can be chosen equal or smaller than N_y. A reason for choosing $N_u < N_y$ is to reduce the computational effort.

For the actual tuning, first the parameters in the state weighting matrix Q are tuned. In this way, the control objectives are satisfied. Afterwards, the weighting of the input (R) is adjusted to avoid unnecessary frequent and rapid changes of the input.

3.5.4 Controller

In Figure 3.4 a schematic representation is depicted of the Hybrid ACC Stop-&-Go controller. The input is the state feedback vector $\overline{x}_e(k) = (x_r(k),\, v_r(k),\, v_h(k),\, u(k-1))^T$. The output of the MPC controller is a desired acceleration $a_d(k) = u(k) = u(k-1) + \delta u(k)$.

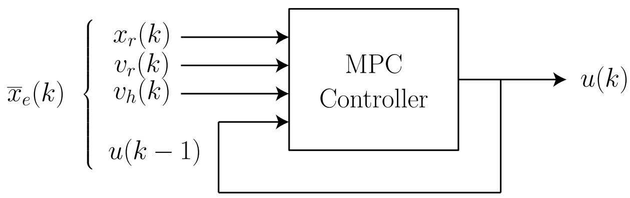


Figure 3.4: Hybrid ACC controller with input the state feedback vector and as output the desired acceleration

The result is a 4D solution space comprising a continuous, piecewise affine state feedback control law, which is dependent on the 4D state vector $\overline{x}_e(t)$. In Figure 3.5, three 2D crosscuts of this solution space are shown. The grey surfaces represent regions with a constant control law. The three crosscuts are made at a constant x_r and show the variation in regions as a function of $x_{e,4} = u(k-1)$. As Figure 3.5 shows, the number and size of the regions R_i changes. Furthermore, the solution space decreases as a result of the velocity dependency of the constraint $u(t) \leq u_{max}(v_h)$ (introduced in Subsection 3.4.2).

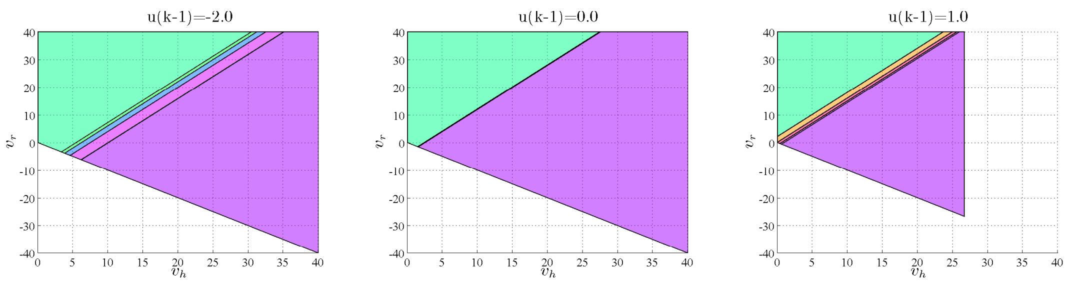


Figure 3.5: Three 2D crosscuts of the solution space at constant $x_r = 10\,\mathrm{m}$ with varying $x_{e,4} = u(k-1) \in \{-2.0,\, 0.0,\, 1.0\}\,\mathrm{m\,s^{-2}}$.

3.5.5 Feasibility and stability

The main problem of MPC is that is does not, in general, a priori guarantee stability. Furthermore, MPC might drive the state to a part of the state space where no solution to the finite time optimal control problem satisfying the constraints exists. Thus, MPC is only applicable if stability and feasibility have been proven (see for more details also Subsection B.3.3).

For the hybrid controller for the ACC Stop-&-Go application, no feasibility guarantee can be given a priori. An invariant subset of the controller must be calculated, such that constraints satisfaction is guaranteed for all time.
When reaching the objectives within the constraints is not possible, two situations can occur: 1. in case of a fast accelerating target vehicle, the host vehicle will be in CC mode. 2. when the host acceleration of $a_h = -3\,\mathrm{m\,s^{-2}}$ is not enough to avoid a possible collision, driver intervention in necessary. In Figure 3.6, only this deceleration boundary is shown. Below this boundary driver intervention in necessary: with e.g. an alarm the driver is warned to apply the brakes. This boundary is only valid for a target acceleration of zero $a_t = 0\,\mathrm{m\,s^{-2}}$ and can be formulated by substituting $t = v_r/a_h$ in

$$x_r(t) = x_r(0) + v_r(0)t + \frac{1}{2}a_h t^2 \tag{3.25}$$

with $x_r(t) = 0$, Eq. (3.25) delivers

$$a_h = \frac{1}{2}\frac{v_r^2}{x_r} \tag{3.26}$$

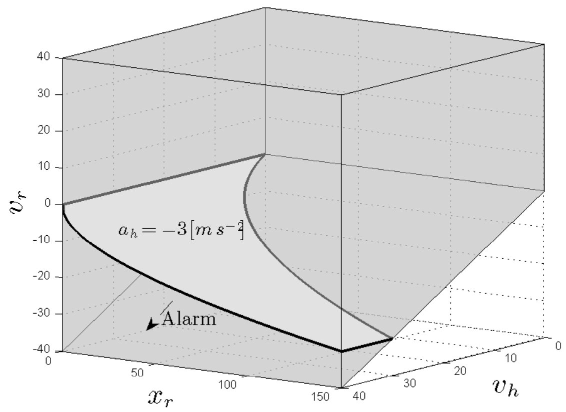


Figure 3.6: Feasibility analysis: below the boundary of $-3\,\mathrm{m\,s^{-2}}$ (constraints satisfaction not achieved): an alarm should warn the driver to apply the brakes

As stated in Section 2.2, stability can be fulfilled by defining a weight on the terminal state, implemented in the performance criterion, or analyzed afterwards. With the definition of a weight imposed on the terminal state, the cost function is changed, denoted by Q_f in Eq. (B.3). For this ACC Stop-&-Go controller, the stability is analyzed afterwards.
MPC of constrained systems is nonlinear, which requires the use of Lyapunov stability theory. In order to test if the closed-loop MPC system is asymptotically stable, a common quadratic Lyapunov function $V(\overline{x})$ is computed according to the LMI-based algorithm of [40], which proves quadratic stability of the hybrid system.

$$V(\overline{x}) = \overline{x}^T P_{Lyap}\,\overline{x} \tag{3.27}$$

Sufficient conditions on the symmetric P_{Lyap} for asymptotic stability are given by the LMIs

$$P > 0 \tag{3.28}$$
$$A^T PA - P < 0 \tag{3.29}$$

This first condition (Eq. (3.28), P is positive definite) implies that $V(\overline{x})$ (Eq. (3.27)) is positive definite [30, section 4.1], which is required for stability. The calculation of the quadratic Lyapunov function for the ACC Stop-&-Go application delivers

$$P_{Lyap} = \begin{bmatrix} 18.25 & 0 & 0 \\ 0 & 36.50 & 0 \\ 0 & 0 & 18.25 \end{bmatrix} \tag{3.30}$$

For this P_{Lyap}, both conditions of Eq. (3.28) and Eq. (3.29) are valid which proves quadratic stability of the hybrid system. This Lyapunov function is independent of the switching law. Switches can occur arbitrarily quickly or even randomly, and the the hybrid system is still stable.

3.5.6 Implementation in MPT toolbox

MPT [32] is a MATLAB toolbox for multi-parametric optimization and computational geometry. MPT is a software tool for computation, analysis and visualization of explicit control laws for linear and Piecewise Affine (PWA) systems. Here, the toolbox is used for the defining, computation, analysis, simulation and experimentation of the explicit MPC framework for this ACC system. The option of custom design of MPC problems with YALMIP (Yet Another LMI Parser) [33] is chosen.
This design is divided into three parts

1. Design phase
   In this part, general constraints (Subsection 3.4.2), requirements (Subsection 3.4.3) and a corresponding cost function (Eq. (3.18)) are formulated.

2. Modification phase
   In this part, the user is allowed to add custom constraints and/or to modify the cost function. Here, the control objectives (Subsection 3.4.1) are introduced. Besides this, also the constraints concerning the jerk are here introduced.

3. Computation phase
   In this part, the explicit controller which respects user constraints, control objectives and requirements is computed, respecting the state boundaries (Subsection 3.4.4).

This results in a solution for this optimal control problem, defined in Section 3.5. This solution can be analyzed (e.g. stability and feasibility, see Subsection 3.5.5 and implementation in e.g. SIMULINK is possible.

Chapter 4

Implementation of Hybrid Adaptive Cruise Control

The ACC MPC controller of the previous chapter serves as a basis for ACC. For implementation in a vehicle several functionalities need to be added, which are introduced here. Next, the overal control structure is presented.

4.1 Introduction

In the previous chapter, the MPC framework is adopted. Based on the output of the radar (relative distance and relative velocity to the target vehicle) and the host velocity, a desired acceleration is generated. However, for implementation, this framework does not cover all possible working conditions. Firstly, CC has to be implemented, when no target vehicle is present. Also, switching between ACC and this CC should incorporated. Secondly, the desired acceleration can not directly actuate the throttle and the brake. Thirdly, the actual implementation needs two workarounds to circumvent unreliable radar data and host velocity measurement data. In the next sections, these additions are treated. The functionalities will be presented in a standard negative feedback control scheme as depicted in Figure 4.1. In this figure 'C' is the controller, 'H' is the plant, i.e. the host vehicle with the slave control loop and the radar. The signals $\overline{x}$ and $\overline{u}$ represent the state feedback vector and the controller output or plant input, respectively.

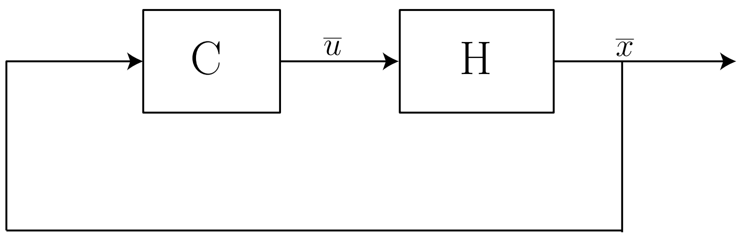


Figure 4.1: Standard control scheme with controller 'C' and plant 'H'

4.2 Adaptive Cruise Control

As already introduced in Section 2.1, ACC controls the velocity and distance of the host vehicle automatically to follow a target vehicle. In Figure 4.2, the plant 'H' and the controller 'C' are shown. The plant contains the host vehicle and the real target. $x_{r,rt}$ and $v_{r,rt}$ are the outputs of the radar of the host vehicle. Together with the host velocity, the output of this plant is the state feedback vector $\overline{x}_{rt}$, which is defined as

$$\overline{x}_{rt} = (x_{r,rt}\ ,\ v_{r,rt}\ ,\ v_h)^T \tag{4.1}$$

This state feedback vector is the input for the controller. The output of this controller is $u = a_{h,d}$, which is the input for the plant.

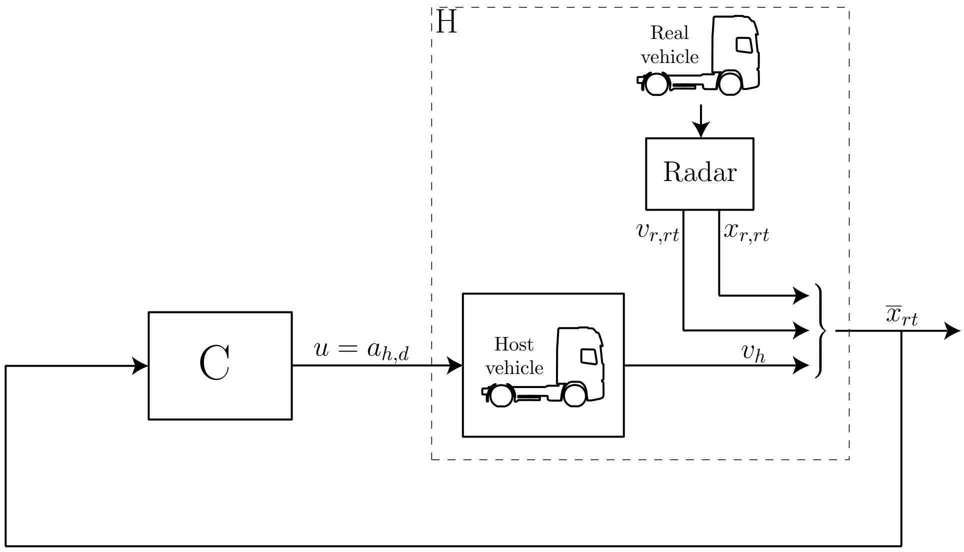


Figure 4.2: Plant 'H', containing the host vehicle and the real target. The input of this plant is the controller output $u = a_{h,d}$. The output is the state feedback vector $\overline{x}_{rt} = (x_{r,rt},\ v_{r,rt},\ v_h)^T$

4.3 Cruise control

In comparison with ACC, the objective for the Cruise Control (CC) controller is different: the velocity of the host vehicle should reach/hold the desired CC velocity. This desired CC velocity will be reached and maintained if no real target vehicle is present. This is realized with a so-called 'virtual target'. This vehicle 'drives' at the desired distance $x_{r,vt} = x_{r,d}(v_h)$, thus $x_{vt} = x_{r,d}(v_h) + x_{r,0}$. The velocity of the virtual target is equal to the desired cruise control velocity $v_{vt} = v_{CC}$ and an acceleration of $a_{vt} = 0$. In Figure 4.3, the introduction of the virtual target is visualized. The relative distance corresponding to the real target $x_{r,rt}$ can be larger as well as smaller than the desired relative distance $x_{r,d}(v_h)$. The driver is able to change the desired CC velocity v_{CC} of the host vehicle.

The plant 'H' now consists of the host vehicle and virtual target (exists always). The plant is depicted in Figure 4.4. In the host vehicle, the vehicle-specific slave control loop is present to represent the host vehicle dynamics: translation of the desired host acceleration $a_{h,d}$ into the achieved host acceleration a_h. The output vector $\overline{x}_{vt}$ contains the relative distance $x_{r,vt}$ and the relative velocity $v_{r,vt}$ between the host vehicle and the virtual target and the host velocity v_h

$$\overline{x}_{vt} = (x_{r,vt}\ ,\ v_{r,vt}\ ,\ v_h)^T \tag{4.2}$$

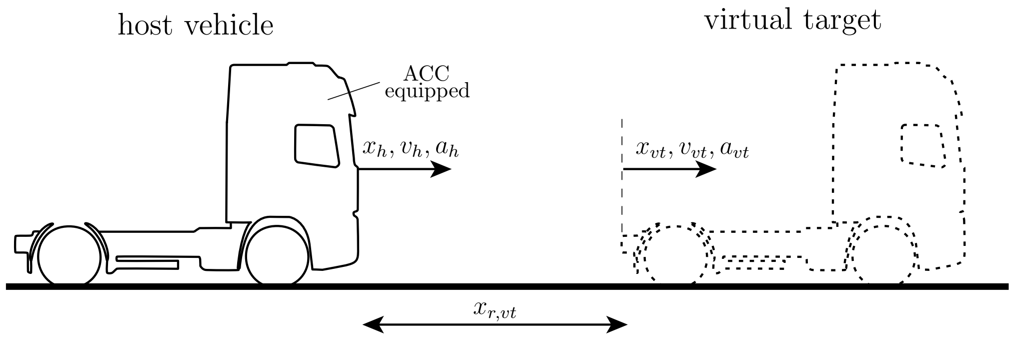


Figure 4.3: Host vehicle with virtual target with $x_{r,vt} = x_{r,d}(v_h)$, $x_{vt} = x_{r,d}(v_h) + x_{r,0}$, $v_{vt} = v_{CC}$ and $a_{vt} = 0$

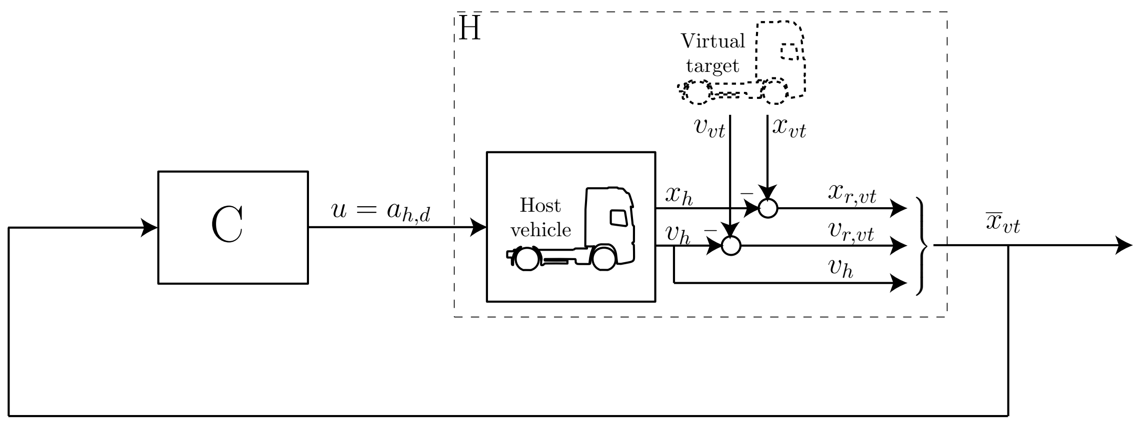


Figure 4.4: Plant 'H', containing the host vehicle and the virtual target. The input of this plant is the controller output $u = a_{h,d}$. The output is the state feedback vector $\overline{x}_{vt} = (x_{r,vt},\, v_{r,vt},\, v_h)^T$

4.4 Switching between Cruise Control and Adaptive Cruise Control

Several situations can occur in combination with the host vehicle, the virtual target (CC) and a possible real target (ACC). Examples are: no target, a real target at $x_{r,rt} < x_{r,d}$ or $x_{r,rt} \geq x_{r,d}$ both possible with $v_{r,rt} < 0$ or $v_{r,rt} \geq 0$. The velocity of the real target can also be higher or lower than the desired cruise control velocity v_{CC}. See Figure 4.5 for the host vehicle together with a virtual target and possibly with a real target.

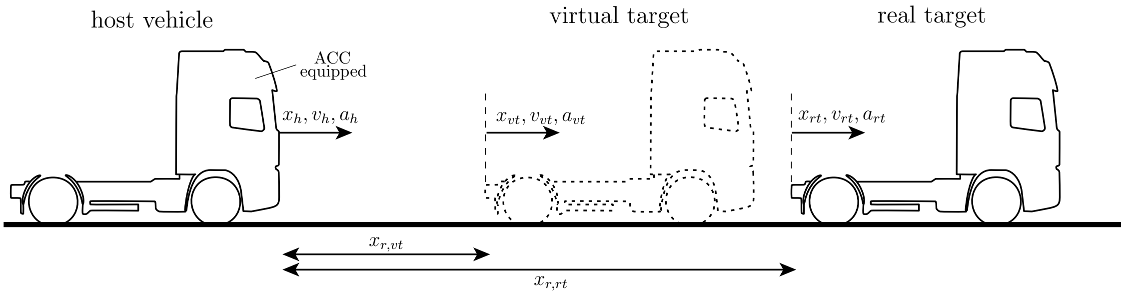


Figure 4.5: Host vehicle with target vehicle: real or virtual target

In order to check the existence of a real target, the following operational checks are necessary:

- Object detected: a relevant object is detected by the radar with the corresponding relative distance x_r and relative velocity v_r.
- Controller range validity: the off-line hybrid ACC Stop-&-Go exists for defined state boundaries of the state vector. The actual relative distance, relative velocity and the host velocity must be within these boundaries. In the ACC Stop-&-Go system, the defined state boundaries are: $0 \leq x_r \leq 180$ m, $-40 \leq v_r \leq 40\,\mathrm{m\,s^{-1}}$ and $0 \leq v_h \leq 40\,\mathrm{m\,s^{-1}}$

If not all operational checks are true, the virtual target provides the input for the controller, see Eq. (4.2). These operational checks are incorporated in the 'Logic' block of Figure 4.6.

If a real target exists, the switching between the real target and the virtual target is also induced by the 'Logic' block. Based on the lowest desired host acceleration, it is decided which target vehicle (real or virtual) should be considered as most important target. In Figure 4.7, this 'Logic' block, containing this determination of the most important target, is depicted. The calculation of these desired host accelerations is realized by using the same explicit MPC controllers of the controller part (introduced in Chapter 3), except the defined jerk constraint in Subsection 3.4.3. In this way, steps are possible in the acceleration and thus the most important object is recognized immediately. The outputs of the MPC controllers (C$_{MPC,rt}$ and C$_{MPC,vt}$) deliver the desired host accelerations, $a_{h,rt}$ and $a_{h,vt}$ respectively. After the determination of the lowest desired host acceleration, the corresponding state vector is the output of the 'Logic' block. This output is defined as $\overline{x} = (x_r\ ,\ v_r\ ,\ v_h)^T$ and this vector will be the input for the controller.
In Appendix D, the simulation results of a cut-in situation are depicted to show the need for this immediate switching.

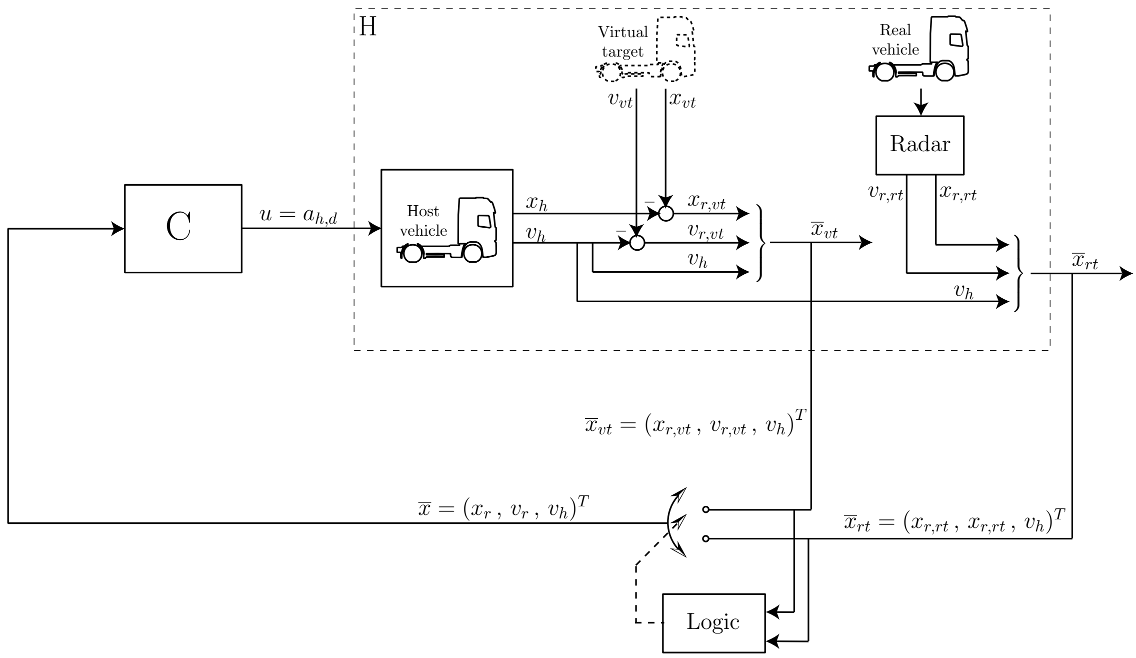


Figure 4.6: Switch between real state vector $\overline{x}_{rt}$ and virtual target state vector $\overline{x}_{vt}$

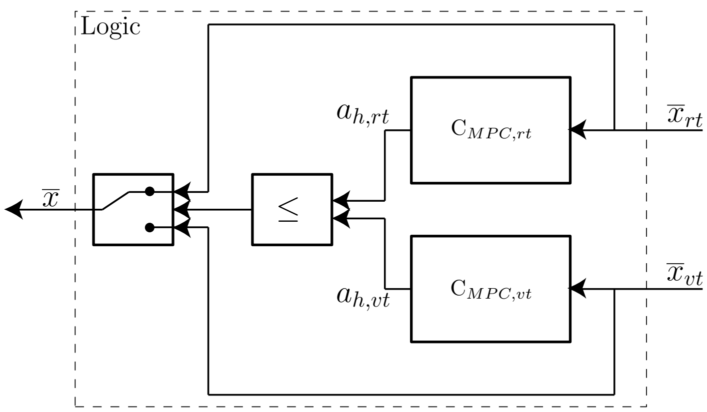


Figure 4.7: 'Logic': The desired host acceleration of the real $a_{h,rt}$ and virtual target $a_{h,vt}$ are compared. The corresponding state vector ($\overline{x}_{rt}$ or $\overline{x}_{vt}$) of the lowest desired host acceleration is the output of 'Logic'.

4.5 Actuation of throttle and brakes

The control of the host velocity requires a subdivision of the control output / desired acceleration $a_{h,d}$ into two separate signals controlling the engine and the brakes of the host vehicle. Consequently, the controller part exists of 'MPC throttle' ($C_{MPC,th}$) and 'MPC brake' ($C_{MPC,br}$), see Figure 4.8. The

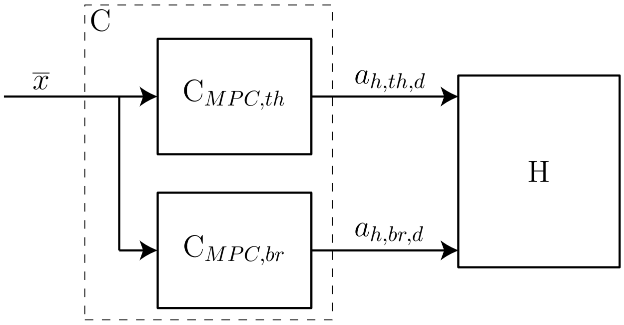


Figure 4.8: Two controllers for actuation of the throttle and brakes

output of 'MPC throttle' is a desired acceleration $a_{h,th,d}$, which should be realized by the engine of the host vehicle. The output of 'MPC brake' is a desired acceleration $a_{h,br,d}$, which should be realized by the brakes of the host vehicle. The desired 'throttle' acceleration is always smaller than the desired 'brake' acceleration ($a_{h,th,d} < a_{h,br,d}$). This is achieved by using a smaller time headway t_{hw} i.e. a smaller desired relative distance x_r as a control objective for determination of $a_{h,br,d}$. In contradiction to the MPC controllers of the previous section, a jerk constraint and a weight on the jerk are added out of comfort reasons ($u(k-1)$ is internally known in the MPC framework). These MPC controllers are already introduced in Section 3.5. By using this principle, the brakes will not be activated immediately after the throttle is released. This results in a comfortable behaviour of the vehicle [45].

In the slave control loop, these desired host accelerations are translated into real host accelerations: $a_{h,th,d}$ into a throttle signal ($a_{h,th}$) and $a_{h,br,d}$ into a brake signal ($a_{h,br}$), see Figure 4.9. Figure 4.10 illustrates combining the two realized host accelerations ($a_{h,th}$ and $a_{h,br}$, dashed lines) into one realized host acceleration a_h, represented by the solid black line. During transactions of releasing the throttle to apply the brakes, the vehicle gradually decelerates due to road friction and drag forces, this is represented by the (almost) horizontal dashed line. Also, as stated in [58, Section 3.2], switching logic with a boundary layer is necessary to avoid frequent switching between throttle and brake controls.

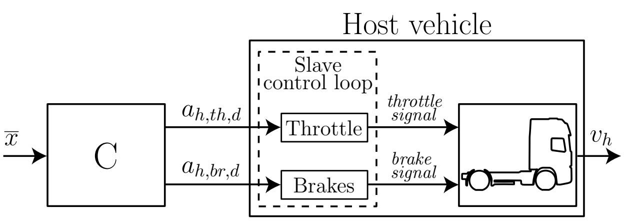


Figure 4.9: Translation of the desired accelerations $a_{h,th,d}$ and $a_{h,br,d}$ into throttle and brake signals

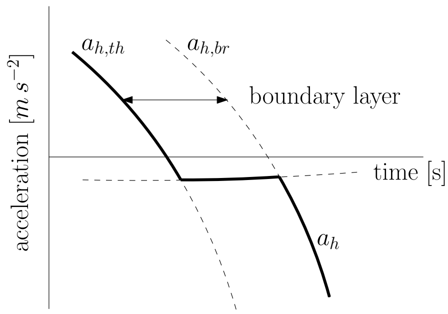


Figure 4.10: Switching between $a_{h,th}$ & $a_{h,br}$

From a control point of view, the switching between two controllers with different control objectives is undesired. This switching can degrade the system performance or even make it unstable. However, in order to create a general and a vehicle-specific part, the creation of two desired accelerations is a solution with desirable behaviour and comparable with human driving behaviour.

4.6 Operation by driver

- ACC activated/deactivated
  The driver can manually activate or deactivate the ACC system. When the ACC system is activated, the current host velocity is the CC velocity of the host vehicle $v_{CC} = v_h$.

- Changing the desired CC velocity
  The desired CC velocity v_{CC} can be manually adjusted by the driver. This desired CC velocity can be increased or decreased with steps of $1\ \mathrm{km\,h^{-1}}$. The minimum desired CC velocity is $5\ \mathrm{km\,h^{-1}}$.

- Throttle overrule
  The driver is able to overrule the ACC system with the accelerator pedal (if the vehicle velocity is too low). In case of overruling, the ACC controller is temporarily switched off. If the accelerator is released, ACC is activated again. This is called throttle overrule. In case of throttle overrule, the input of the virtual vehicle is $x_r = x_{r,d}$ & $v_r = 0$. This results in $a_x = 0$, which corresponds in a smooth transition when the throttle is released by the driver.

- Brake overrule
  When the driver applies the brake pedal the ACC system is switched off. The ACC system has to be manually activated again.

4.7 Implementation issues

4.7.1 Radar noise

When $v_h = 0$ and $v_t = 0$, the radar output of the relative velocity v_r is (slightly) negative due to radar noise. This can be interpreted as a negative velocity of the target vehicle. However, in Subsection 3.4.4 is defined $0 \leq v_t \leq v_{t,max}$, so a small negative v_t (and v_r) will result in a non-existing state feedback controller. In order to avoid this, $max(v_r, -v_h)$ is the input for v_r. See item 3 of Subsection 3.4.4.

4.7.2 Host velocity measurement

Due to the limited range of the ABS sensors, the measurement of the host velocity at low velocities is not reliable: the output of these sensors for $v_h \leq 1\ \mathrm{m\,s^{-1}} (= 3.6\ \mathrm{km\,h^{-1}})$ is zero. A work-around is needed to neglect this phenomena for a proper implementation into the hybrid control framework: the output must not be influenced by errors in the velocity measurement.
To avoid steps in the desired host acceleration, caused by steps in the velocity around $1\ \mathrm{m\,s^{-1}}$ and thus steps in the desired distance (Figure 4.11), the calculation of the desired distance is changed: the host velocity is saturated between $1\ \mathrm{m\,s^{-1}} \leq v_{h,sat} \leq v_{h,max}$. Thus, the host velocity, as part of the controller input state vector $\overline{x}$, is adjusted. The new desired distance is non-linear

$$x_{r,d} = \begin{cases} x_{r,0} & \text{for} \quad v_h \leq 1 \\ x_{r,1} + t_{hw} \cdot v_{h,sat} & \text{for} \quad v_h > 1 \end{cases} \tag{4.3}$$

In order to guarantee continuity between both desired distance calculations, the following relation must be valid (at $v_h = 1\ \mathrm{m\,s^{-2}}$): $x_{r,1} = x_{r,0} - t_{hw}$.

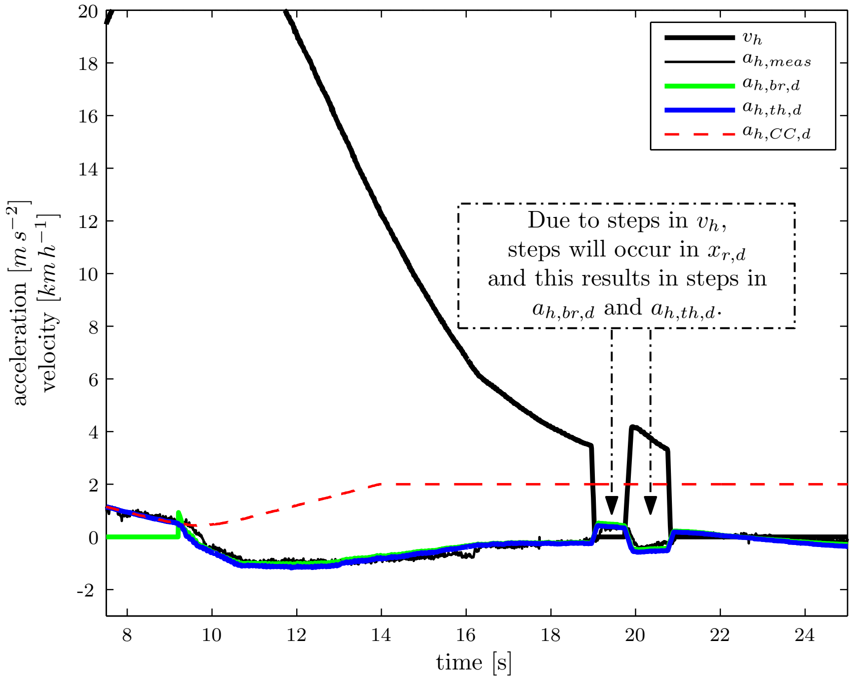


Figure 4.11: Not reliable measurement of the host velocity with resulting effects in the desired relative distance and corresponding desired host accelerations

4.8 Overall control structure

Adding the functionalities of the previous sections to the hybrid ACC Stop-&-Go controller yields the overall negative feedback control structure with switching controller inputs, depicted in Figure 4.12. Switching the input (switching between real and virtual target vehicle, based on MPC controllers without jerk constraint, see Subsection 4.4), results in outputs of these controllers that are subjected to the control objectives, requirements and constraints, defined in Section 3.4. In this manner a direct and smooth transition between the vehicles is realized.

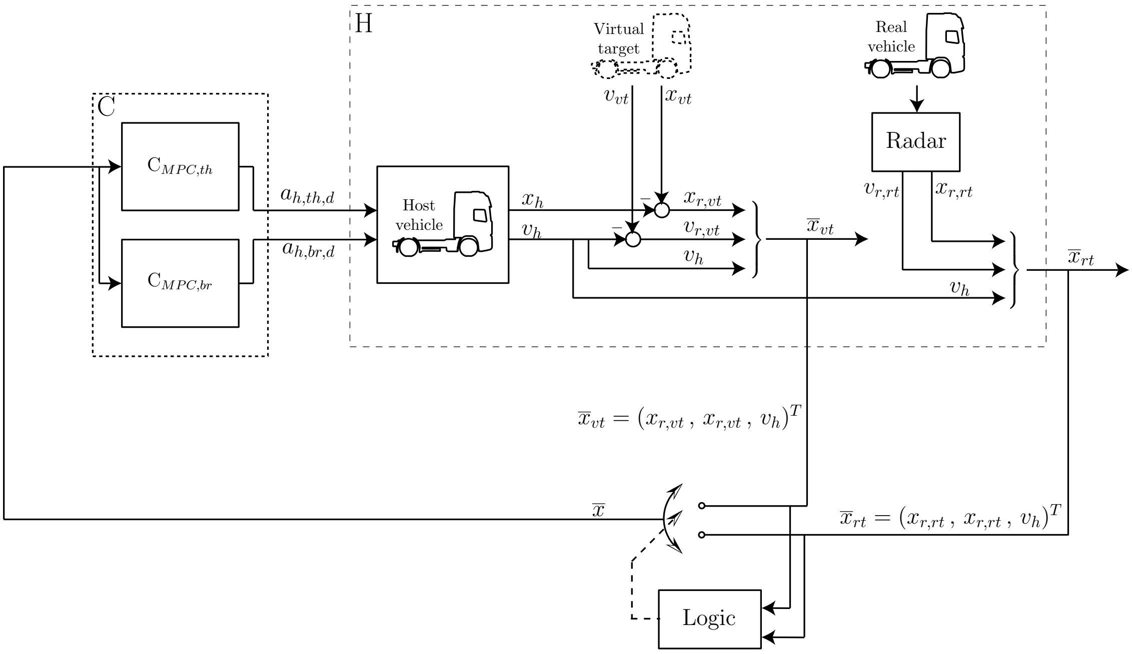


Figure 4.12: Overall control structure

In this figure, the functionalities 'throttle overrule' and 'brake overrule' are not displayed. If one of the two situations occurs, both control signals, $a_{h,th,d}$ and $a_{h,br,d}$, are overruled. For 'throttle overrule': the controller output $a_{h,th,d}$ is replaced with the accelerator pedal output and $u_{h,br,d}$ is set to zero. For 'brake overrule': the controller output $a_{h,th,d}$ will become zero and $a_{h,br,d}$ is changed to the value corresponding with the brake pedal.
Besides this, also the slave control loop is not displayed (as introduced in Subsection 4.5, Figure 4.9).

Figure E.1, in Appendix E, shows a control structure where the controller output is switched, instead of switching the controller inputs of the overall control structure (introduced in this section), see Figure 4.12 or Figure E.2. This last controller is preferable: switching between a real or a virtual target determines the input for the MPC controllers $C_{MPC,th}$ and $C_{MPC,br}$. The output of both MPC controllers will respect the defined control problem in Section 3.4, which is not the case for the 'controller output switching' control structure. In this way, the 'controller input switching' structure prevents steps in the desired host accelerations.

4.9 Feasibility and stability

Feasibility for the overall control structure is not different to both separate hybrid ACC Stop-&-Go controllers. Due to the already proven feasibility for both MPC controllers (see Subsection 3.5.5), the overall control structure is feasible.

In order to be able to perform a stability analysis of the overal control structure, this structure, depicted in Figure 4.12, is simplified to a standard control problem with a switching controller input. For this simplification, see Figure 4.13.

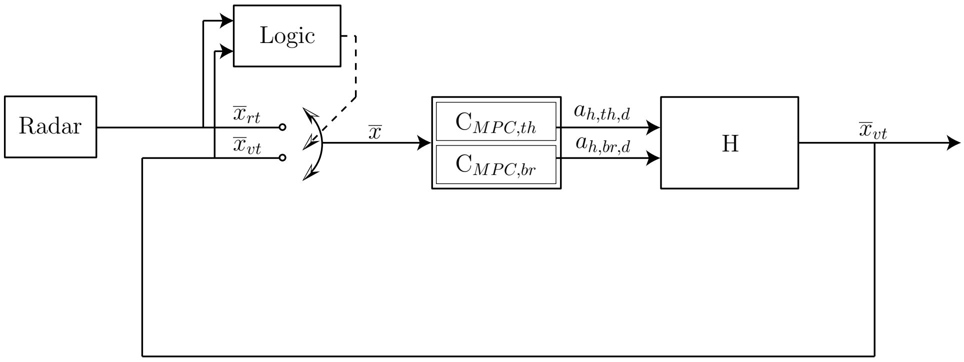


Figure 4.13: Simplified controller structure

Stability is proven for both separate MPC controllers '$C_{MPC,th}$' and '$C_{MPC,br}$' for actuation of respectively the throttle and the brakes. Switching the input of these controllers does not result in unstable behaviour, see again Subsection 3.5.5.

Both the throttle and the brakes are inputs for the host vehicle. As a result, switching occurs between these inputs. The two individual subsystems are asymptotically stable as discussed in Subsection 3.5.5. However, switching between these two subsystems can deliver a stable or unstable switched system. This switching is part of the slave control loop (depicted in Figure 1.1), which is out of scope for this project. During the road experiments (introduced in the next chapter), this switching did not result in unstable behaviour. Further investigation on 'stability under slow switching' can result in a successful stability analysis: it is well known that a switched system is stable if all individual subsystems are stable and the switching is sufficiently slow, so as to allow the transient effects to dissipate after each switch. In [34, Section 3.2], this property is formulated and justified using multiple Lyapunov techniques.

Chapter 5

Evaluation of Hybrid Adaptive Cruise Control Stop-&-Go

This chapter discusses the (performance) evaluation of the designed hybrid controller for ACC Stop-&-Go. Simulation as well as experimental results will be discussed.

5.1 Introduction

For the (performance) evaluation of the designed hybrid controller for ACC Stop-&-Go, first the goals for this evaluation are described. Secondly, several situations are determined to enable evaluation for the total envelope of working conditions of the ACC Stop-&-Go and to enable controller comparison with the existing TNO ACC Stop-&-Go. Simulation and experimental results are discussed.

5.1.1 Goals

The goals for the evaluation of the overal control structure are to

- validate applicability / functionality of the newly designed controller for a complete envelope of working conditions
- validate simulations with experimental results
- determine controller behaviour (comfort & desirability) in defined situations

5.1.2 Test program

In order to evaluate the functionality of the controller, a set of 8 distinct situations encompassing the total envelope of working conditions is determined. The following items each describe a defined situation and the performance criteria (based on Subsubsection 2.1.2) for the ACC Stop-&-Go controller.

1. Approach of a standstill or stationary driving vehicle, yielding a CC to ACC switch
   A target vehicle drives with constant velocity (zero or lower than the host vehicle velocity) at an initially larger distance than the desired distance, $x_{r,0} > x_{r,d}$ and $v_{r,0} < 0$.
   Performance criteria: a comfortable deceleration and jerk profile of the host vehicle is desired, without overshoot in relative distance (possible collision). The value of TTC_{min} and the relative velocity are important metrics for desirability to enable evaluation between ACC applications.

2. Cut-in with a negative velocity difference
   A new target vehicle appears at a smaller distance than the desired distance, with a lower velocity than the host vehicle, $x_{r,0} < x_{r,d}$ and $v_{r,0} < 0$.
   Performance criteria: a quick built-up of the maximum deceleration of the host vehicle. This serves to avoid a collision and to reach the desired distance and desired velocity. TTC_{min} must be as high as possible and the velocity difference should disappear quickly, in order to handle the situation desirable.

3. Cut-in with a positive velocity difference
   A new target vehicle appears at a smaller distance than the desired distance, with a higher velocity than the host vehicle, $x_{r,0} < x_{r,d}$ and $v_{r,0} > 0$.
   Performance criteria: due to the higher velocity of the target vehicle, both the control objectives (defined in Subsection 3.4.1) will be reached without actuating the brakes.

4. Cut-out of a target vehicle, yielding an ACC to CC switch
   A cut-out is defined as the disappearance of the target vehicle, corresponding a switch from ACC to CC.
   Performance criteria: reach the v_{CC} in a comfortable manner (low levels of acceleration and jerk).

5. Following of a decelerating (to standstill) target vehicle
   A target vehicle drives at the desired relative distance and the velocity difference between host and target vehicle is zero. At a certain time, the target vehicle applies the brakes and decelerates to standstill.
   Performance criteria: no overshoot is allowed in the relative distance between the two vehicles at standstill to avoid a collision. For desirability again the minimal value of TTC_{min} is an important metric.

6. Accelerating at a traffic light
   This situation is defined as a drive away of target vehicle from standstill with a high acceleration level.
   Performance criteria: comfortable a_h and j_h in order to reach the desired distance x_r and desired velocity $v_h = v_t$ or desired cruise control velocity $v_h = v_{CC}$.

7. Accelerating at a traffic jam
   This situation is defined as a drive away of target vehicle from standstill with a low acceleration level (positive/negative), with possible decelerations till standstill.
   Performance criteria: low levels of acceleration and jerk and avoid copying sinusoidal behaviour of the target vehicle velocity.

8. CC behaviour: accelerating and decelerating to v_{CC}
   The CC behaviour implies the behaviour of the controller in case of lower (e.g. a cut-out: ACC to CC) and higher (e.g. throttle overrule) velocities than the user-defined CC velocity v_{CC}.
   Performance criteria: No brake after throttle overrule ($v_h > v_{CC}$) or after adjusting (lowering) the cruise control velocity.

5.1.3 Performance evaluation

For performance evaluation, the Hybrid ACC controller behaviour is compared to the TNO ACC. Two different MPC controller are designed: MPC-1 with a defined weight on the jerk and MPC-2 without jerk weighting. This realized by the adaption of the defined weight of R: $R_{MPC-1} \neq 0$ and $R_{MPC-2} = 0$. Note that, both MPC-1 as well as MPC-2 have constraints on the jerk as defined in Subsection 3.4.2. The tuning of both Hybrid ACC Stop-&-Go controllers (see Subsection 3.5.3, especially the definition of Q and R) is based on simulations with TNO ACC Stop-&-Go System.

The analysis of differences and similarities between the tested control behaviour is based on levels of acceleration and jerk, the realized acceleration and jerk profiles and perceptual variables such as TTC, see Subsubsection 2.1.2.

5.2 Simulations

5.2.1 PreScan

For the simulations, MATLAB / SIMULINK is used in combination with PRESCAN, which provides a visual simulation environment [52]. PRESCAN (developed by TNO Automotive) is a simulation environment for the design and evaluation of the next generation of vehicles with ADA systems. Within PRESCAN, a vehicle can actually sense its surroundings and - based on the decision algorithms implemented - react to it. Typical detection systems that can be used range from radar, lidar and stereo vision, to car-2-car and car-2-infrastructure communication systems.
PRESCAN provides the user with a powerful development and evaluation environment for intelligent vehicle systems. Vehicles are considered to be 'intelligent' when they have (multiple) sensor(s) on board perceiving the car's surroundings and consequently have extensive data processing capabilities which either warn the driver so a potentially dangerous situation can be avoided, or which take over control once this dangerous situation is inevitable. In Figure 5.1, a screenshot is depicted of a PRESCAN simulation environment. A cut-in situation is shown: the three target vehicles perform a lane change. Three ACC Stop-&-Go equipped host vehicles react on this. The left host vehicle is equipped with the TNO ACC Stop-&-Go, the middle and the right host vehicle have Hybrid ACC Stop-&-Go: MPC-1 and MPC-2.

Figure 5.1: Screenshot of a PRESCAN simulation environment. Three target vehicles perform a cut-in situation. The three host vehicles are equipped with ACC Stop-&-Go systems: left is the TNO ACC Stop-&-Go vehicle depicted, the middle and the right vehicle are equipped with Hybrid ACC Stop-&-Go: MPC-1 (middle) and MPC-2 (right).

5.2.2 Simulation results

All test scenarios (Subsection 5.1.2) are simulated and evaluated. In this section, the results of two simulations (situations 2 and 5 of Subsection 5.1.2) are shown and explained in detail. The other results are summarized in the last subsection.

1. cut-in with a negative velocity difference (sit. 2)

In this case a cut-in with a negative velocity difference is simulated, the results are depicted in Figure 5.2. This cut-in is characterized by a small step in the relative distance (from a desired relative distance of 47 m towards 30 m) and a large step in relative velocity (from a target velocity of 60 $\mathrm{km\,h^{-1}}$ towards 20 $\mathrm{km\,h^{-1}}$) due to a new target vehicle at $t = 10$ s. Before this, the host vehicle is driving in CC mode. Here, the combination of host and virtual target vehicle meets the control objectives ($x_r = x_{r,d}$ and $v_r = 0$).

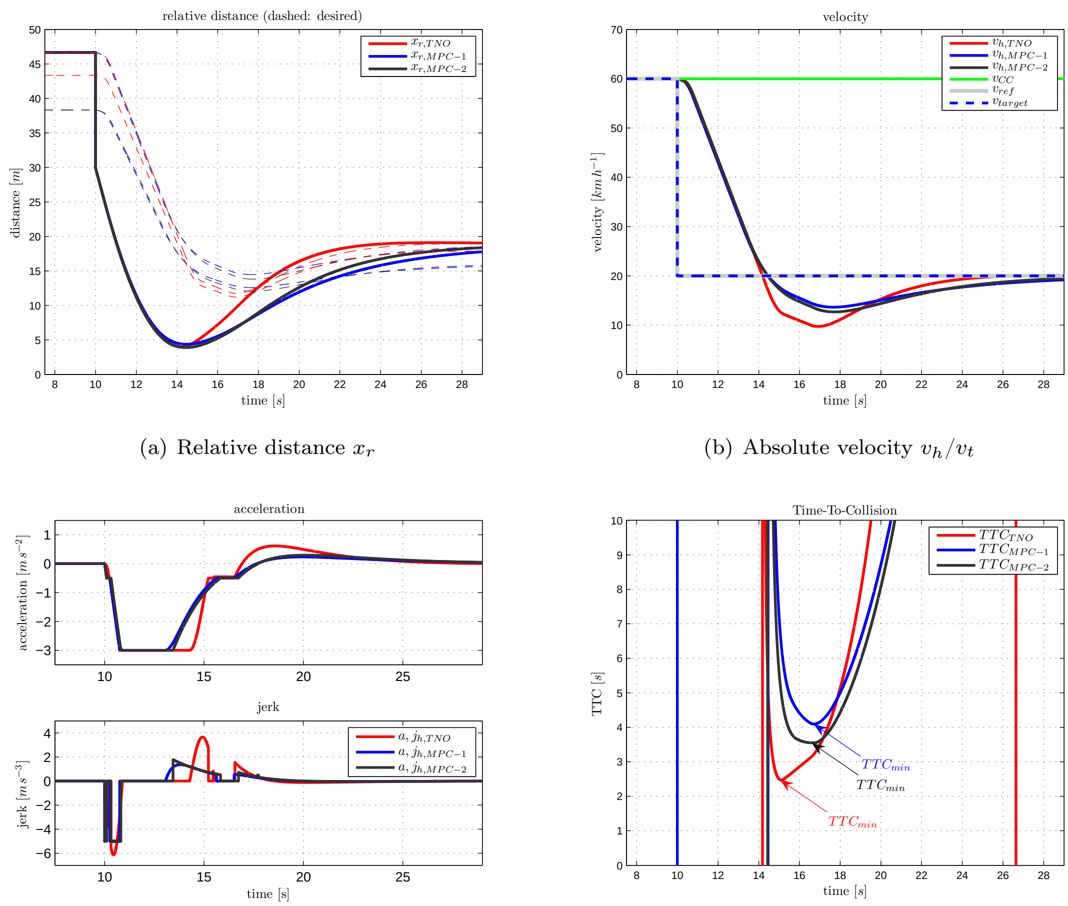
The steps in x_r and v_r, do not result in steps in the host acceleration a_h, due to the constraints on the host jerk j_h. These steps cause a transition between throttling and braking at about 10 s and maximum (constraint) acceleration levels of $a_h = -3\ \mathrm{m\,s^{-2}}$, both depicted in Figure 5.2(c). The build-up of this maximum deceleration of the TNO controller is faster due to a larger jerk level ($j_{h,TNO} = -6\ \mathrm{m\,s^{-3}}$) relative to both MPC controllers (which are constraint at $j_{h,MPC} = -5\ \mathrm{m\,s^{-3}}$). At $t = 13$ s a difference occur in the acceleration profiles: the TNO controller remains at $a_h = -3\ \mathrm{m\,s^{-2}}$, while both the MPC controllers have positive jerk levels, resulting in a smaller deceleration. At $t = 14$ s a large positive jerk of the TNO controller results in a fast changing acceleration. At $t = 15 - 16$ s a second transition between the brakes and the throttle occur, which can be seen in the acceleration profile. These differences clearly results in different (relative) velocities, relative distances and time-to-collisions.
Concerning the host vehicle velocities, between the different ACC controllers, the rise time is equal. The overshoot in v_h differences between the controllers is shown in Figure 5.2(b): the host vehicle equipped with the TNO's ACC will have a lower velocity, or a larger relative velocity difference. The settling time of $v_{h,TNO}$, however, is smaller than both $v_{h,MPC-1}$ and $v_{h,MPC-2}$. In Figure 5.2(b) also the host cruise control velocity (v_{CC}, solid green line), the target velocity (v_{target}, dashed blue line) and the reference velocity (v_{ref}, solid gray line) are depicted.

The relative distances are shown in Figure 5.2(a). The cut-in situation shows a corresponding overshoot in the relative distance x_r relative to the desired relative distances ($x_{r,th,d}$ and $x_{r,br,d}$, dashed lines), see Figure 5.2(a). Further, corresponding to the different jerk / acceleration / velocity levels there are also here deviations. Although the relative distance of the TNO ACC Stop-&-Go controller after the cut-in is larger than the Hybrid ACC Stop-&-Go controllers, the levels for the TTC$_{min}$ (Figure 5.2(d)) are lower for the TNO controller (due to large relative velocity errors).

For this cut-in situation, the higher TTC$_{min}$ levels indicate a more comfortable behaviour for the MPC ACC Stop-&-Go controllers. However, due to the larger relative distance and the lower host velocity of the TNO ACC controller, this controller results in a safer solution.

Summarizing, due to higher levels of the jerk and smaller levels for TTC$_{min}$ of the TNO ACC Stop-&-Go, the Hybrid ACC Stop-&-Go controllers have a more comfortable and desirable behaviour. For the MPC ACC Stop-&-Go controllers, the defined jerk levels can quickly be changed, because they are explicitly defined in the control problem. In case of the TNO ACC-&-Go controller, changing the jerk constraints directly is not possible.

For the controllers MPC-1 and MPC-2, this situation is almost solved similar. Differences take place in the jerk profile: due to jerk weighting of MPC-1, steps in jerk will be avoided (otherwise, higher costs in the cost function). This can affect the experience of this situation positively: a smoother jerk profile will be more comfortable. Negatively, this involves a larger settling time.



(a) Relative distance x_r

(b) Absolute velocity v_h/v_t

(c) Acceleration a and jerk j

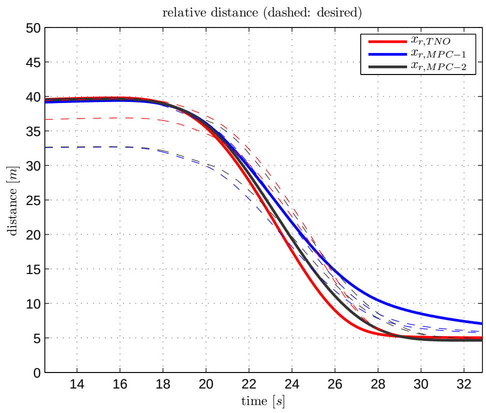
(d) TTC

Figure 5.2: Cut-in with a negative velocity difference

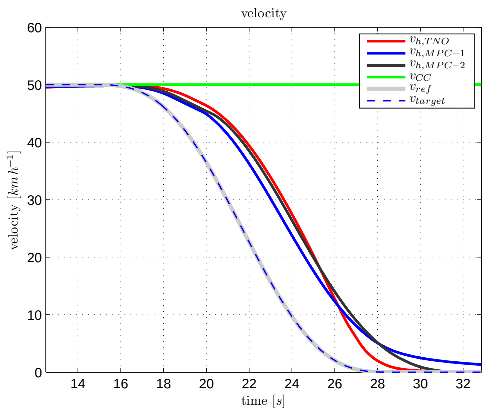
2. following of a decelerating (to standstill) target vehicle (sit. 5)

The target vehicle drives at $v_t = 50\ \mathrm{km\,h^{-1}}$. The host vehicle drives at the desired distance and with no velocity difference. At $t = 16$ s, the target vehicle starts decelerating with $a_t = -2\ \mathrm{m\,s^{-2}}$ till standstill ($v_t = 0\ \mathrm{km\,h^{-1}}$). The simulation results are depicted in Figure 5.3. As a result of the decelerating target vehicle, the host vehicle needs slow down to meet the control objectives and to avoid a collision, see Figure 5.3(b). A transition occurs between the throttle and the brakes at about $t = 18 - 20$ s. After the braking action is (almost) finished the brakes are released at $t = 29$ s.
During this simulation, comparable levels of the host acceleration and jerk are shown between the three controllers till $t = 24$ s (with the quickest build-up of deceleration of MPC-1), see the acceleration profiles in Figure 5.3(c). However, the acceleration profiles differ: the MPC controllers start with a higher desired deceleration, ending with a smaller deceleration. The TNO controller solves the situation with a different acceleration profile: smaller deceleration till $t = 22$ s and a higher deceleration after $t = 24$ s. This corresponds with human driving behaviour. On the other hand, here, this results in uncomfortable behaviour, because the driver will stay alert to apply the brakes.
Besides this, the output of the TNO controller shows steps in the jerk and thus corresponding fluctuations in the acceleration during $t = 26 - 28$ s, which is not comfortable. This is caused by the several defined situations with different settings for the TNO ACC Stop-&-Go controller. Switching between different situations can cause this behaviour. This behaviour will not appear in the MPC control framework, because of the defined control problem with constraints and a weight on the jerk.

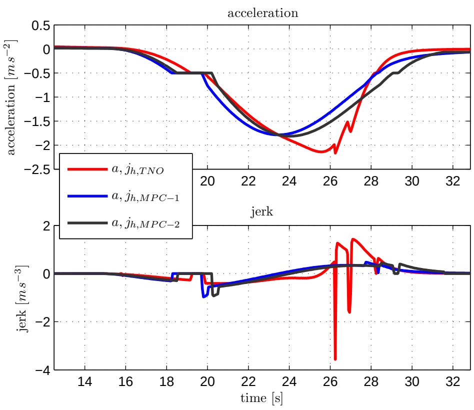
Due to the larger deceleration, the TNO host vehicle reaches a full stop as first. MPC-2 stops almost 1 s later. The settling time for the MPC-1 host vehicle is very large. Corresponding to the acceleration profiles for $t < 24$ s, the host velocities of the TNO, MPC-1 and the MPC-2 controller are comparable. After this, the host velocities differ significantly.
A little distance overshoot for the second MPC controller is shown in Figure 5.3(a). In Figure 5.3(d), the resulting TTC levels are depicted: TNO and MPC-2 have comparable levels for TTC$_{min}$. Due to the faster build-up of the deceleration of MPC-1, the related TTC$_{min}$ is around 5 s in comparison with TTC_{TNO} and TTC_{MPC-2} (around $3 - 3.5$ s).
Just like the previous simulation, the higher levels of TTC$_{min}$ of the MPC ACC-&-Go controllers, influence the comfortable experience of this situation positively.



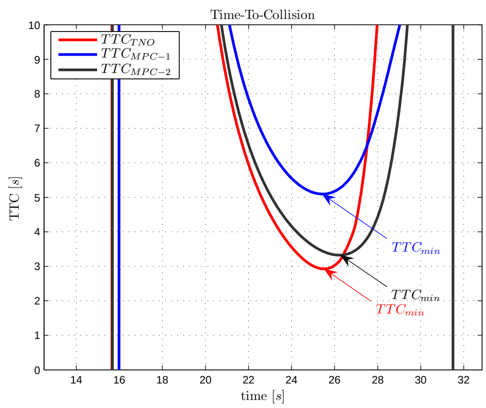
(a) Relative distance x_r



(b) Absolute velocity v_h/v_t



(c) Acceleration a and jerk j



(d) TTC

Figure 5.3: Following of a decelerating (to standstill) target vehicle

Remaining situations of the test program

In this subsection, the results of the remaining situations of the defined test program (situations 1, 3, 4, 6, 7 and 8) are presented in Table 5.1.

	ACC Stop-&-Go	situation					
		1	3	4	6	7	8
$a_{min}[\mathrm{m\,s^{-2}}]$	MPC-1	-1.0	-2.0	0	0	0	0
	MPC-2	-1.2	-2.6	0	0	0	0
	TNO	-1.6	-0.3	0	0	0	0
$a_{max}[\mathrm{m\,s^{-2}}]$	MPC-1	0	0.7	3.0	1.3	1.8	2.6
	MPC-2	0	1.0	3.0	1.2	1.8	2.0
	TNO	0	0.6	1.5	2.3	2.3	1.4
$j_{min}[\mathrm{m\,s^{-3}}]$	MPC-1	-1.2	-5.0	-1.4	-0.6	-0.3	-1.2
	MPC-2	-1.3	-5.0	-1.0	-0.4	-0.3	-0.6
	TNO	-2.6	-2.1	-0.5	-2.3	-6.8	-0.4
$j_{max}[\mathrm{m\,s^{-3}}]$	MPC-1	0.2	1.6	5.0	0.8	0.3	5.0
	MPC-2	0.2	1.7	5.0	0.7	0.3	5.0
	TNO	0.9	0.6	1.6	6.4	25	1.6
$\mathrm{TTC}_{min}[\mathrm{s}]$	MPC-1	6.3	-	-	-	-	-
	MPC-2	4.5					
	TNO	3.0					

Table 5.1: Simulation results of situation 1, 3, 4, 6, 7 and 8 of the test program

In the following list, the presented results are shortly discussed. In Appendix F, the corresponding figures are shown.

- sit. 1 During the approach of a standstill vehicle, the desired deceleration level of the TNO ACC Stop-&-Go controller is higher. This is mainly the result of a smaller time range in which the host vehicle is stopped. Here, steps in the desired host jerk, caused by non smooth switching between states, result in uncomfortable fluctuations in the desired host acceleration.
  The MPC ACC Stop-&-Go controller result in a smooth desired host acceleration profile. Different settings for the tuning parameters (MPC-1 vs. MPC-2) show the influence on the cost function and thus on e.g. different start times of the desired host deceleration, maximum levels of desired host deceleration and different levels for TTC_{min}.
- sit. 3 This situation is characterized by a cut-in ($x_r < x_{r,d}$) with a positive velocity difference $v_r > 0$. Braking is not necessary or even uncomfortable: a release of the throttle with a corresponding negative acceleration, due to energy dissipation in the vehicle, is satisfactory. Situations, where the relative velocity is larger than zero ($v_r > 0$) and with a relative distance $x_r < x_{r,d}$ or $x_r > x_{r,d}$, converges automatically to the desired distance to the target vehicle or CC without using the brakes.
  For these situations, the current MPC ACC Stop-&-Go controller brakes ($a_h < -0.5\,\mathrm{m\,s^2}$). The desired output of the MPC ACC Stop-&-Go controller is however the deceleration generated by releasing the throttle, i.e. $a_h \geq -0.5\,\mathrm{m\,s^2}$, such as the TNO ACC Stop-&-GO controller.
- sit. 4 The cut-out of a target vehicle shows different levels of desired host accelerations: the MPC ACC Stop-&-Go controllers calculates high and uncomfortable desired host accelerations. To avoid this, the TNO ACC Stop-&-Go controller has included a maximum acceleration depending on the host velocity, which is comfortable. To implement this behaviour into the MPC framework, the constraint 'Maximum acceleration depending on host velocity' in Subsection 3.4.2 is also added to the control problem.

sit. 6 Accelerating at a traffic light: TNO shows significant higher levels of jerk and acceleration, which is not comfortable. Here, the levels for the MPC ACC Stop-&-Go controller are more comfortable. Also, the acceleration and jerk profile (see Figure F.4(c)) confirm this conclusion.

sit. 7 Accelerating at a traffic jam: just like the previous item, the desired levels of acceleration and jerk of the TNO ACC Stop-&-Go controller are higher. And just like item 1, steps in the desired host jerk, caused by non smooth switching between states, result in uncomfortable fluctuations in the desired host acceleration. This results in more uncomfortable behaviour relative to the MPC ACC Stop-&-Go controller. Note that a jerk level peaks of $25\,\mathrm{m\,s^{-3}}$ will not be noticed in road experiments.

sit. 8 CC behaviour: accelerating to v_{CC}: the MPC ACC Stop-&-Go show higher levels of jerk and accelerations. For this situation, this leads to an unnecessary uncomfortable situation. To reach the desired CC velocity v_{CC}, low levels of host jerk and host acceleration are wanted. The TNO ACC Stop-&-Go controller recognizes this CC mode and reacts to this with lower and thus more comfortable levels of acceleration and jerk.

5.2.3 Conclusions

Two different situations of the test program have been simulated and evaluated in detail. Together with the short discussion of the remaining situations, proper and safe operation of the ACC Stop-&-Go system is showed.
With these simulations, differences and similarities between TNO, MPC-1 and MPC-2 have been discussed. Similarities appear in the levels of acceleration and jerk. Main differences are the jerk profiles, acceleration profiles and levels of TTC_{min}. Based on these conclusions, the MPC controllers result in a more comfortable and desirable behaviour for the defined situations.

As a result of these simulations, experiments with a real vehicle are possible for further (performance) evaluation of the designed ACC MPC Stop-&-Go controller. Due to shorter settling times and higher levels of the realized TTC$_{min}$ in comparison with MPC-1, MPC-2 is used for the road experiments. The TNO ACC controller is also tested in these road experiments.

5.3 Experiments

To validate the simulation results, the controller has been implemented on an CarLab using a DSPACE AUTOBOX. This CarLab is depicted in Appendix G. In contrast to the simulations, where a vehicle model was not present, here the controller is tested in real road experiments.
The functionality of the controller was first tested in the TNO VeHIL test facility before the tests in actual traffic have been performed. VeHIL is a hardware-in-the-loop test facility for the development of intelligent road vehicles. The principle of VeHIL is to simulate only the relative motion of other vehicles with respect to the test vehicle. This allows efficient, safe and reproducible testing [20]. Here, the car is tested on a chassis dynamometer and with the so-called Moving Base (MB). The MB is a Automated Guided Vehicle (AGV): a four-wheel steered and four-wheel driven (4ws4wd) vehicle. With this setup, simple functionality checks have been performed. In Section 4.7, two issues resulting from these first experiments are already introduced. The results of these tests will not be discussed here. Afterwards, the road experiments have been carried out. The results of two situations (the same situations as used in the simulations) will be reported in the next subsections.

5.3.1 Experiments results

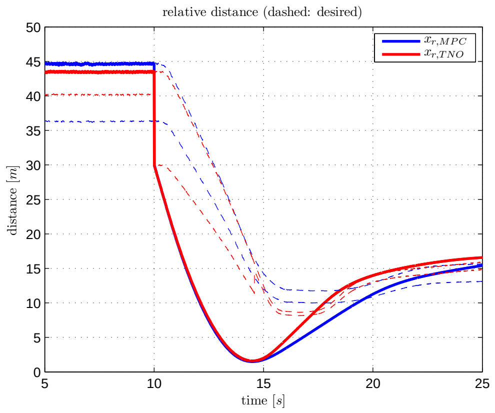
Again, the defined test program (Subsection 5.1.2) is used to validate the controllers and the same defined situations used for the simulations are used here: 1. cut-in with a negative velocity difference and 2. following of a decelerating (to standstill) target vehicle.

1. cut-in with a negative velocity difference

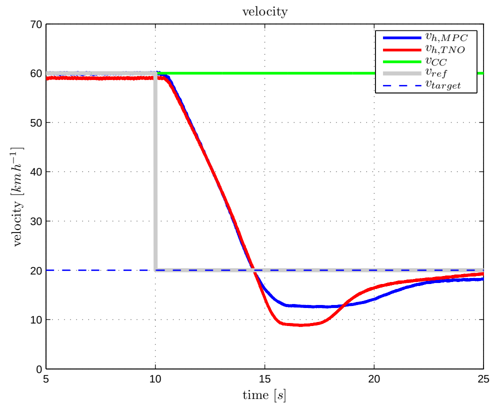
The experiment set-up is almost identical to the simulation of a cut-in with a negative velocity difference (almost the same steps in x_r and v_r at $t = 10$ s), depicted in Figure 5.4. The profiles of the acceleration show similarities: resulting $a_{h,min}$ of (constrained) -3 m s^{-2}, transitions between throttle ($a_{h,th,d}$) and brake ($a_{h,br,d}$) at 10 and $17 - 18$ s and the same acceleration inequalities at $t = 14 - 15$ s. The resulting overshoot of the host velocities are of the same order and the difference of the levels for TTC$_{min}$ are the same: $TTC_{min,\,TNO} < TTC_{min,\,MPC}$. Differences are shown in Figure 5.4(c): the desired deceleration is not met by the actual deceleration. These errors are caused by the actuation of the brakes in the slave control loop (see Figure 1.1) and by vehicle dynamics. Due to these errors, the host velocity is negatively influenced (Figure 5.4(b)) and relative distance becomes smaller than necessary, see Figure 5.4(a). Also, the TTC (Figure 5.4(d)) is affected: both the levels of TTC_{min} are smaller (simulations: $TTC_{min} = 2.5 - 4$ s, experiments $TTC_{min} = 1 - 1.5$ s). As depicted in Figure 5.4(a), it can be seen that a step occurs in the desired distance of the TNO ACC Stop-&-Go controller. In this case, this does not affect the desired host acceleration (because it remains at -3 m s^{-2}). This is caused by non smooth switching between states, which already appeared at the simulations.

Due to this cut-in situation, this manoeuvre is not comfortable (high levels of the host acceleration and the host jerk). Desirable behaviour for this situation is to avoid a collision. To determine if this situation is desirable, [54] states that a level of 1.5 s for TTC$_{min}$ distinguishes critical from normal behaviour. Here, this level for TTC$_{min}$ is violated by the TNO ACC Stop-&-Go controller.

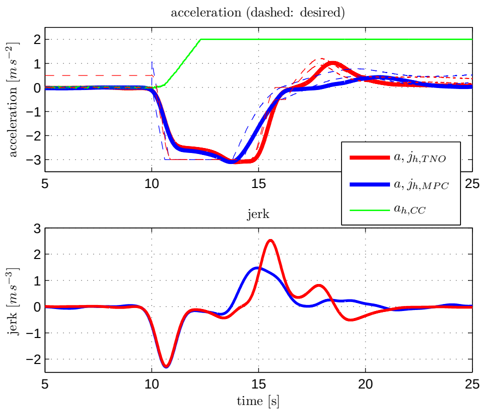
Analysis of the vibrational comfort of this situation, experienced by a human being (Subsubsection 2.1.2), is depicted in Figure 5.5. The PSD plots of the realized host accelerations of both ACC Stop-&-Go controllers for this situation are shown. Instead of the filtered signals used in Figure 5.4, this PSD analysis is based on the raw measurement data. The magnitude corresponds to the involved energy at the different frequencies. This gives information about the vibrational comfort.
For the whole frequency range depicted in the figure, the magnitude of the MPC ACC Stop-&-Go is slightly larger than the magnitude corresponding to the TNO ACC Stop-&-Go controller. For example: at 4 Hz the difference is almost a factor 1.1 (MPC: $1.9e^{-1}$, TNO: $1.7e^{-1}$). Based on these results, the TNO ACC Stop-&-Go controller has a slightly higher vibrational comfort for this situation.



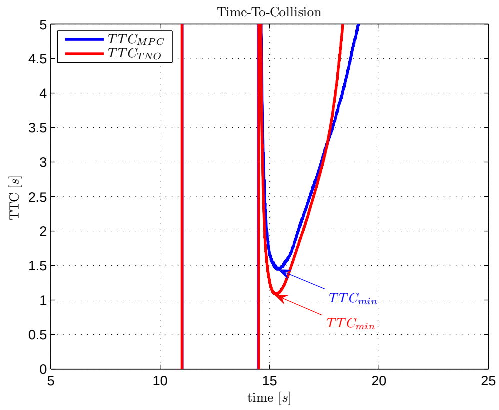
(a) Relative distance x_r



(b) Absolute velocity v_h/v_t



(c) Acceleration a and jerk j



(d) TTC

Figure 5.4: Cut-in with a negative velocity difference

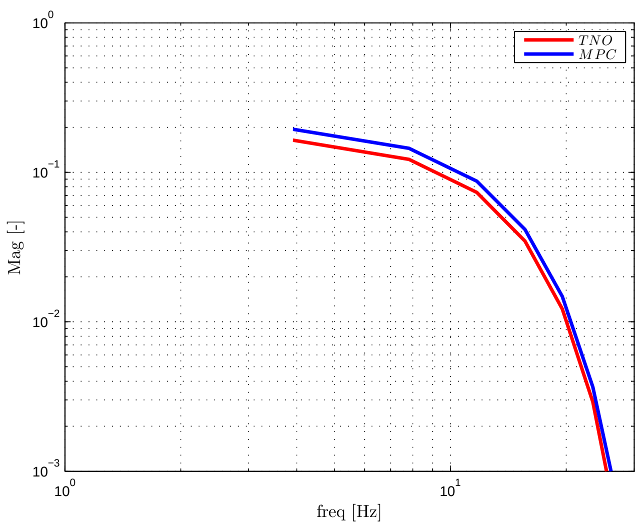


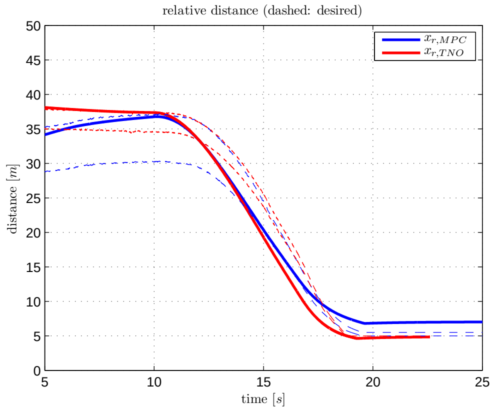
Figure 5.5: PSD

2. following of a decelerating (to standstill) target vehicle

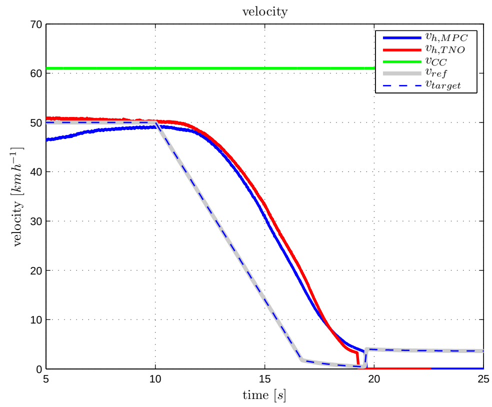
For this second experiment 'following of a decelerating (to standstill) target vehicle', the set-up is also almost exactly the same as the corresponding simulation, see Figure 5.6. The initial velocity of both host vehicles (TNO: solid red line and MPC: solid blue line) and the target vehicle (dashed blue line) is $v_h = v_t = 50\,\mathrm{m\,s^{-1}}$, see Figure 5.6(b). In Figure 5.6(a) the relative distance is depicted: again the solid red line belongs to the relative distance achieved by the TNO controller. The solid blue line is related to the MPC controller. Both the dashed lines corresponds to desired relative distances.
At $t = 10\,\mathrm{s}$ the target vehicle starts decelerating till standstill ($v_t = 0\,\mathrm{km\,h^{-1}}$). Both the TNO and MPC controller starts decelerating in order to meet the objectives of a smaller desired relative distance and to attain a zero velocity difference. See Figure 5.6(c) for the resulting (desired) acceleration profiles. Both controllers have a comparable acceleration profile for $t = 5 - 15\,\mathrm{s}$. After $t = 15\,\mathrm{s}$, the TNO desired and actual acceleration show remarkable jumps in jerk as well as acceleration profile. This phenomena is identical to the simulation and is not comfortable.
Due to unsolved host velocity measurement issues (Subsection 4.7.2), the velocity jumps at $t = 19\,\mathrm{s}$ from 3.6 to $0\,\mathrm{km\,h^{-1}}$. An error in the actual and desired distance is the result.
Both the levels of TTC_{min} are smaller of the experiments in comparison with the simulations (Figure 5.6(d)).

For both ACC Stop-&-Go controllers, this experiment was experienced as comfortable (for TNO till $t = 15\,\mathrm{s}$), which is confirmed by the levels and profiles of the host accelerations and the host jerk. Based on the TTC$_{min}$ levels, this manoeuvre is not critical for both ACC Stop-&-Go controllers .

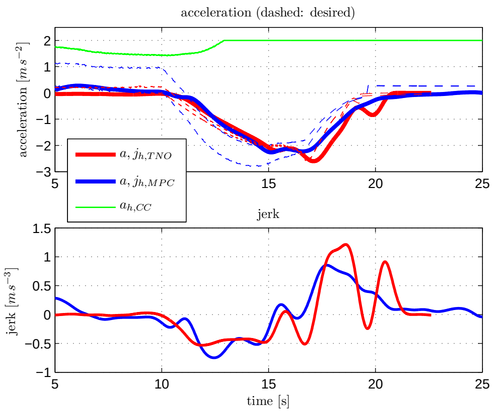
The PSD analysis for this situation (see Figure 5.7) shows that the energy for the TNO ACC Stop-&-Go is larger than the energy involved by the MPC ACC Stop-&-Go controller. Also for this situation, the PSD analysis is based on the raw measurement data. Analyzing Figure 5.7 delivers a factor 2.3 difference at $4\,\mathrm{Hz}$ (MPC: $3.8e^{-2}$, TNO: $8.7e^{-2}$). Here, the conclusion is that the MPC ACC Stop-&-Go controller has a higher vibrational comfort.
In comparison with the results of the previous situation, depicted in Figure 5.5, this situations results in lower magnitudes around the relevant frequency range. This means that this situation is more comfortable concerning vibrational comfort.



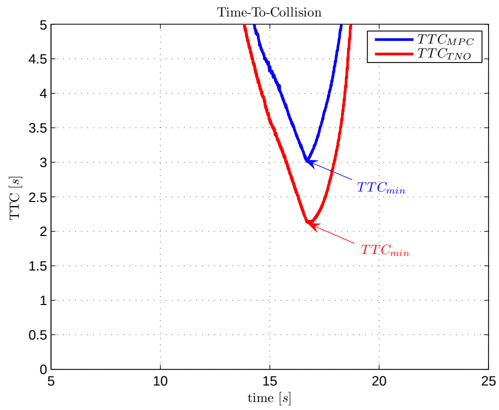
(a) Relative distance x_r



(b) Absolute velocity v_h/v_t



(c) Acceleration a and jerk j



(d) TTC

Figure 5.6: Following of an accelerating, decelerating (to standstill) vehicle

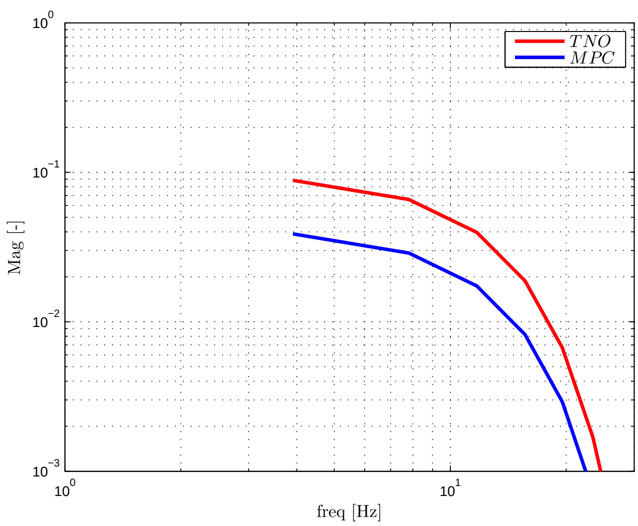


Figure 5.7: PSD

5.3.2 Conclusions

The conclusions after the simulations, stated in Subsection 5.2.3, also hold for the experiments: based on the defined performance metrics in Subsubsection 2.1.2, the performance of the Hybrid ACC Stop-&-Go controller is more comfortable and desirable. In comparison with the simulations, the differences in acceleration profile and levels of TTC between the TNO ACC Stop-&-Go and Hybrid ACC Stop-&-Go controller are more leveled. A possible reason for this is the difference of the desired and the actual deceleration.

The overall conclusions, after the performed simulations and road experiments with respect to the stated goals in Subsection 5.1.1, are:

- Simulations are useful for checking proper and safe operation of the ACC MPC controller. The levels of acceleration and jerk are similar to the experiments. The levels of TTC_{min} are higher for the simulations.
- Experiments are necessary to design an ACC controller, that is robust for not ideal sensors, such as radar noise and measurement errors (in this case the velocity measurement with ABS sensors).
- No overall level of TTC_{min} for desirability can be defined, because of the situation dependency.
- For the designed MPC controller, the used constraint levels of jerk ($-5 \leq j_h \leq 5\,\mathrm{m\,s^{-3}}$), levels of acceleration ($-3 \leq a_h \leq 3(v_h)\,\mathrm{m\,s^{-2}}$) and the tuning parameters (weighting matrices Q on the state vector and R on the input, prediction horizon N_y and control horizon N_u) result in comfortable and desirable behaviour.
  The tuning proces of the Hybrid ACC Stop-&-Go controller is easy by changing the parameters in Q and R. Further investigation on tuning can improve the controller output based on the performance criteria.

Further investigation enables the definition of 1. minimum and maximum levels of acceleration and jerk as comfortable levels and 2. several levels of TTC$_{min}$ as being desirable for different situations.

Chapter 6

Conclusions and recommendations

6.1 Conclusions

The main conclusion of this thesis is that an Adaptive Cruise Control Stop-&-Go system is successfully designed with Hybrid control, in simulations as well as road experiments.

The contribution of this master's thesis is the design of an Adaptive Cruise Control Stop-&-Go (ACC Stop-&-Go) with a Model Predictive Controller (MPC) structure.

For this ACC Stop-&-Go control problem, a MPC controller has been designed. MPC can deal with MIMO systems and is able to handle constraints on input, states and/or outputs. Besides this, MPC solves an optimal control by minimizing a performance criterion on-line. The weightings Q and R enable tuning of this optimization problem. By moving the computational effort off-line (which delivers an explicit MPC , also called a Hybrid controller), the applicability of MPC is extended to small-size/fast-sampling applications. Furthermore, stability of the resulting hybrid controller can be proven and a feasibility analysis delivers a subset of the controller for which constraint satisfaction is guaranteed.
As a result, this control framework enables the design of an easy-to-tune ACC application, ensuring comfortable and desirable behaviour.

Disadvantages of explicit MPC are the rapid growth of size in the explicit solution as the problem size increases. Furthermore, the look-up table is only valid for a specific configuration of the system.

For the actual implementation in a vehicle, several functionalities are added ensuring a proper and safe operation for a complete envelope of working conditions. The validation and performance evaluation of the designed hybrid controller for ACC has been carried out by simulations with PreScan and by road experiments with an instrumented vehicle.

6.2 Recommendations

Tuning of the hybrid ACC controller

Close the loop: based on the simulation and experimental results, it is possible to tune the Hybrid ACC Stop-&-Go controller. This can result in higher controller performance regarding comfort (low levels for the host acceleration and the host jerk) and desirability (high levels for TTC$_{min}$).
One step further is the incorporation of the performance metrics into the MPC framework. Based on this, automated optimal tuning of the controller, regarding comfort as well as desirability, is possible. Automated tuning involves the automatically adaption of the tuning parameters of the MPC controller to improve the performance metrics. These metrics are driver-specific: e.g. the time headway between the vehicle equipped with ACC and its predecessor. Based on logging of driver-specific behaviour for a certain time period, these parameters can be adjusted to level the performance of the ACC system with the driver.

Model with vehicle dynamics

The used model in this thesis are, in fact, two integral actions. By including the vehicle dynamics into the model, the division of the desired host acceleration into the desired host accelerations for throttle and brakes in not necessary anymore. This improves the switching between the actuation of the throttle and the actuation of the brakes.
Besides this, the implementation of CC and ACC into the model will result in an easier control structure: the switching between the real and virtual target is included into the control framework. Moreover, different maximum levels of acceleration and/or jerk for ACC and CC can be formulated to improve the comfortable behaviour.

Estimation target acceleration

If the target vehicle acceleration (a_t) is known, improvement of the controller performance is possible: the earlier detection of target vehicle behaviour can result in an earlier, appropriate reaction with corresponding higher performance. The target acceleration is unknown due to physical limitations of the used FMCW radar. A recommendation for estimating the target acceleration is to study a so-called Kalman filter. The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error [56].

Extension functionality ACC Stop-&-Go

- MIO loss due to driving in curves
  Missed target detections (MIO loss) can occur when driving in curves. The radar isn't able to track objects in sharp curves due to the limited opening angle. Acceleration to the desired CC velocity in a curve, due to this MIO loss, should be avoided. This is not yet incorporated in the current ACC Stop-&-Go model. To implement this into the control framework, the yaw rate (in $\mathrm{rad\,s^{-1}}$) of the host vehicle can be used. MIO loss will occur above a certain level of the yaw rate.

- Positive velocity difference involves no braking
  Situations, where the relative velocity is larger than zero ($v_r > 0$) and with a relative distance $x_r < x_{r,d}$ or $x_r > x_{r,d}$, converges automatically to the desired distance to the target vehicle or CC without using the brakes. For these situations, the current MPC ACC Stop-&-Go controller brakes ($a_h < -0.5\,\mathrm{m\,s}^2$). The desired output of the MPC ACC Stop-&-Go controller is however the deceleration generated by releasing the throttle, i.e. $a_h \geq -0.5\,\mathrm{m\,s}^2$. First example: a cut-in ($x_r < x_{r,d}$) with a positive velocity difference $v_r > 0$. Braking is not necessary or even uncomfortable: a release of the throttle with a corresponding negative acceleration, due to energy dissipation in the vehicle, is satisfactory. Second example: if a target vehicle at $x_r > x_{r,d}$ and with $v_r > 0$ is present, the host vehicle will, or accelerate to meet the control objectives, or the host vehicle will be in CC ($v_t > v_C C$).

Slave control loop

During the road experiments the desired acceleration levels are not completely achieved by the throttle/brakes. In the slave control loop, the desired acceleration is translated into the actuation of the throttle or the brakes. Improvement of this will result in a performance improvement. E.g. the level of TTC$_{min}$, in case of a cut-in with a negative velocity difference, will be higher because of the faster achieved maximum deceleration. Perfect tracking of the desired host acceleration is not possible, because of the vehicle dynamics.
The current slave control loop is an open-loop system. By the implementation of a feedback/closed-loop controller (e.g. a PI controller), improved performance of the slave controller is expected.

Stability switching between throttle and brakes

In this project, the system, containing the MPC controllers to actuate the throttle and the brakes of the host vehicle, are stable. However, both the throttle and the brakes deliver the input for the host velocity. Switching occurs between the actuation of the throttle and brakes. For this switching, no stability guarantee for the whole closed loop system can be given.

Sensor fusion

Sensor fusion techniques can be applied to merge the information provided by different sensors. A typical approach is to combine radar with vision, since radar has high distance accuracy, but low lateral resolution, whereas video has a low distance accuracy with high lateral resolution. Another improvement could be achieved by implementing a car-2-car communication system. ACC systems can be extended to Cooperative Adaptive Cruise Control (CACC) systems, where the relative motion is accurately estimated by a combination of an environmental sensor (for example the used FMCW radar) and car-2-car communication.

Bibliography

[1] Balluchi, A., Benvenuti, L., Ferrari, A., and Sangiovanni-Vincentelli, A. Hybrid systems in automotive electronics design. *International Journal of Control Vol. 79*, No. 5 (2006), pp. 375–394.

[2] Baotic, M. *Optimal Control of Piecewise Affine Systems – a Multi-parametric Approach*. PhD thesis, Swiss Federal Institute of Technology (ETH), (2005).

[3] Baotic, M., Vasak, M., Peric, N., and Morari, M. Hybrid System Theory Based Optimal Control of an Electronic Throttle. *American Control Conference* (2003), pp. 5209–5214.

[4] Becker, S. Panel discussion on introduction of intelligent vehicles into society: technical, mental and legal aspects. mental models, expectable consumer behaviour and consequences for system design and testing. *Proceedings of the Intelligent Vehicles Symposium IEEE* (1996), pp. 313–318.

[5] Bemporad, A., Borodani, P., and Mannelli, M. Hybrid control of an automotive robotized gearbox for reduction of consumptions and emissions. *HSCC, LNCS 2623* (2003), pp. 81–96.

[6] Bemporad, A., Borrelli, F., Glielmoy, L., and Vascay, F. Hybrid control of dry clutch engagement. *Proceedings of the European Control Conference* (2001), pp. 635–639.

[7] Bemporad, A., Heemels, W., and Schutter, B. D. Equivalance of hybrid dynamical models. *Automatica Vol. 37*, No. 7 (2001), pp. 1085–1091.

[8] Bemporad, A., Heemels, W., and Schutter, B. D. On hybrid systems and closed-loop MPC systems. *IEEE Transactions on Automatic Control Vol. 47*, No. 5 (2002), pp. 863–869.

[9] Bemporad, A., and Morari, M. Control of systems integrating logic, dynamics, and constraints. *Automatica Vol. 35*, No. 3 (1999), pp. 407–427.

[10] Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. The explicit linear quadratic regulator for constrained systems. *Automatica Vol. 38*, No. 1 (2002), pp. 3–20.

[11] Boer, E. Car following from the driver's perspective. *Journal of Transportation Research* (2000), pp. 201–206.

[12] Borrelli, F. *Discrete Time Constrained Optimal Control*. PhD thesis, Swiss Federal Institute of Technology (ETH), (2002).

[13] Borrelli, F., Bemporad, A., Fodor, M., and Hrovat, D. An mpc/hybrid system approach to traction control. *IEEE Transactions on Control Systems Technology Vol. 14*, No. 3 (2006), pp. 541–552.

[14] Boyd, S., and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, (2004).

[15] Corona, D., Lazar, M., Schutter, B. D., and Heemels, M. A hybrid mpc approach to the design of a smart adaptive cruise controller. *IEEE International Conference on Control Applications* (2006), pp. 231–236.

[16] de Schutter, B., and Heemels, W. *Modeling and Control of Hybrid Systems - Lecture Notes of the DISC Course,* (2004).

[17] Di Cairano, S., Bemporad, A., Kolmanovsky, I., and Hrovat, D. Model predictive control of magnetic automotive actuators. *American Control Conference* (2007), pp. 5082–5087.

[18] García, C. E., Prett, D. M., and Morari, M. Model predictive control: Theory and practice - a survey. *Automatica Vol. 25,* No. 3 (1989), pp. 335–348.

[19] Gietelink, O. *Design and Validation of Advanced Driver Assistance Systems.* PhD thesis, Technische Universiteit Delft, (2007).

[20] Gietelink, O., Ploeg, J., de Schutter, B., and Verhaegen, M. Vehil: A test facility for validation of fault management systems for advanced driver assistance systems. *Proceedings of the 1st IFAC Symposium on Advance in Automotive Control* (2004).

[21] Giorgetti, N., Bemporad, A., Kolmanovsky, I., and Hrovat, D. Explicit hybrid optimal control of direct injection stratified charge engines. *IEEE ISIE* (2005), pp. 247–252.

[22] Giorgetti, N., Bemporad, A., Tseng, H., and Hrovat, D. Hybrid model predictive control application towards optimal semi-active suspension. *IEEE ISIE* (2005).

[23] Goodrich, M. A., Boer, E. R., and Inoue, H. A model of human brake initiation behavior with implications for ACC design. *Proceedings of IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems* (1999), 86–91.

[24] Grant, P., and Haycock, B. The effect of jerk and acceleration on the perception of motion strength. *AIAA Modeling and Simulation Conference* (2006), pp. 303–313.

[25] Grieder, P. *Efficient Computation of Feedback Controllers for Constrained Systems.* PhD thesis, Swiss Federal Institute of Technology (ETH), (2004).

[26] Grieder, P., Lühti, M., Parrilo, P., and Morari, M. Stability & Feasibility of Constrained Receding Horizon Control. *European Control Conference* (2003).

[27] Han, D., and Yi, K. A driver-adaptive range policy for adaptive cruise control. *Proceedings of IMechE Vol. 220,* Part D: J. Automobile Engineering (2006), pp. 321–334.

[28] Hoedemaker, M. *Driving with intelligent vehicles.* PhD thesis, TU Delft, (1999).

[29] Hoetink, A. Advanced cruise control en verkeersveiligheid. Stichting Wetenschappelijk Onderzoek Verkeersveiligheid, SWOV, (2003).

[30] Khalil, H. K. *Nonlinear systems.* Prentice Hall, (2002).

[31] Kusters, J. Modeling and coordinated control of advanced automotive transmissions. Master's thesis, Technische Universiteit Eindhoven, DCT 2005.058, (2005).

[32] Kvasnica, M., Grieder, P., Baotic, M., and Christophersen, F. Multi-parametric toolbox (mpt). http://control.ee.ethz.ch/~mpt/, (2006).

[33] Löfberg, J. YALMIP : A toolbox for modeling and optimization in MATLAB. *Proceedings of the CACSD Conference* (2004), pp. 284 – 289.

[34] Liberzon, D. *Switching in Systems and Control.* Birkhauser Boston, (2003).

[35] Loos, S., Kennes, N., and Wallentowitz, H. Comfort assessment of vehicles. *Institut für Kraftfahrwesen Aachen* (2007).

[36] Lygeros, J. Lecture notes on hybrid systems. http://robotics.eecs.berkeley.edu/~sastry/ee291e/lygeros.pdf, (2004).

[37] Marsden, G., McDonald, M., and Brackstone, M. Toward an understanding of adaptive cruise control. *Journal of Transportation Research Vol. 9*, pt. C (2001), pp. 33–51.

[38] MathWorks, T. Signal processing toolbox 6 - user's guide, (2007).

[39] Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. Constrained model predictive control: Stability and optimality. *Automatica Vol. 36*, No. 6 (2000), pp. 789–814.

[40] Mignone, D., Ferrari-Trecate, G., and Morari, M. Stability and stabilization of piecewise affine and hybrid systems: an lmi approach. *Proceedings of the 39th IEEE Conference on Decision and Control Vol. 1*, No. 7 (2000), pp. 504–509.

[41] Möbus, R., Baotic, M., and Morari, M. Multi-object adaptive cruise control. *HSCC* (2003), pp. 359–374.

[42] Mosterman, P. J. An overview of hybrid simulation phenomena and their support by simulation packages. *HYbrid Systems: COmputation and COntrol Vol. LNCS-1569* (1999), pp. 178–192.

[43] Naus, G., van den Bleek, R., Ploeg, J., Scheepers, B., van de Molengraft, R., and Steinbuch, M. Explicit MPC design and performance evaluation of an ACC Stop-&-Go. *submitted* (2007).

[44] Prestl, W., Sauer, T., Steinle, J., and Tschernoster, O. The bmw active cruise control acc. *SAE Technical Paper Series*, SAE 200010344 (2000), pp. 1–7.

[45] Scheepers, B. Implementation of a stop&go adaptive cruise control in a man truck, based on the groeneveld flr radar. *TNO Advanced Chassis and Transport Systems* (2005).

[46] Skogestad, S., and Postlethwaite, I. *Multivariable Feedback Control: Analysis and Design*. John Wiley & Sons, (2005).

[47] Stanton, N., Young, M., and McCaulder, B. Drive-by-wire: The case of driver workload and reclaiming control with adaptive cruise control. *Safety Science Vol. 27*, No. 2/3 (1997), pp. 149–159.

[48] Strandemar, K. On objective measures for ride comfort evaluation. Master's thesis, Royal institute of technology (KTH), Department of Signals, Sensors and Systems, (2005).

[49] Tøndel, P. *Constrained Optimal Control via Multiparametric Quadratic Programming*. PhD thesis, Norwegian University of Science and Technology, Department of Engineering Cybernetics, (2003).

[50] Tøndel, P., Johansen, T., and Bemporad, A. An algorithm for multi-parametric quadratic programming and explicit mpc solutions. *Automatica Vol. 39*, No. 3 (2003), pp. 489–497.

[51] TNO. Netherlands organisation for applied scientific research. http://www.tno.nl/.

[52] TNO Automotive. *PreScan R 2.2 Help Documentation*, (2007).

[53] Vahidi, A., and Eskandarian, A. Research advances in intelligent collision avoidance and adaptive cruise control. *IEEE Transactions on Intelligent Transportation Systems Vol. 4*, No. 3 (2003), pp. 143–153.

[54] Van der Horst, R., and Hogema, J. Time-to-collision and collision avoidance systems. *The 6th Workshop of the International Cooperation on Theories and Concepts in Traffic Safety* (1993), p. 15–22.

[55] Venhovens, P., Naab, K., and Adiprasito, B. Stop and go cruise control. *International Journal of Automotive Technology Vol. 1*, No. 2 (2000), pp. 61–69.

[56] Welch, G., and Bishop, G. An introduction to the kalman filter. http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html/, (2006).

[57] Yamamura, Y., Tabe, M., Kanehira, M., and Murakami, T. Development of an adaptive cruise control system with stop-and-go capability. *SAE Technical Paper Series Vol. 110*, No. 7 (2001), pp. 413–419.

[58] Yi, K., Hong, J., and Kwon, Y. A vehicle control algorithm for stop-and-go cruise control. *Proceedings of the I MECH E Part D Journal of Automobile Engineering Vol. 215*, No. 10 (2001), pp. 1099–1115.

[59] Zheng, A., and Morari, M. Stability of model predictive control with soft constraints. *IEEE Transactions on automatic control Vol. 40*, No. 10 (1995), pp. 1818–1823.

Appendix A

Hybrid Systems

A.1 Introduction

In [36] an useful characterization has been made of dynamical systems. Roughly speaking a dynamical systems describes the evolution of a state over time. As a well-known illustration the following differential equation is considered

$$\dot{x} = f(x(t)), u(t) \tag{A.1}$$

The state variable at time t in this case is given by $x(t)$ (e.g. position and velocity of a vehicle) and typically takes values in $\mathcal{X} \subseteq \mathbb{R}_n$ and evolves over time $t \in \mathcal{T} \subseteq \mathbb{R}$ according to Eq. ($A.1$). The variable $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ at time t denotes either control inputs or disturbances.
Based on the type of the state the following classification can be made of systems [16]:

1. continuous state: The state takes values in a continuous set, e.g. $\mathbb{R}^n$ for some $n \geq 1$. $x \in \mathbb{R}^n$ denotes the continuous state.
2. discrete state: The state takes values in a countable or finite discrete set $\{q_1, q_2, \cdots\}$. q denotes the discrete state.

Based on the set of times $\mathcal{T}$ over which the states evolves, dynamical systems can be categorized in:

1. continuous time: $\mathcal{T}$ is a continuous set, e.g., a subset of $\mathbb{R}$. $t \in \mathbb{R}$ denotes continuous time. Continuous system dynamics can be described by possibly large systems of differential equations. These can be either ordinary differential equations (ODEs) or contain algebraic constraints as well to form differential and algebraic equations (DAEs).
2. discrete time: $\mathcal{T}$ is a discrete set, e.g., a subset of the integers $\mathbb{Z}$. $t \in \mathbb{Z}$ denotes discrete time (and t_k denotes the corresponding real time instant).

Finally, we make distinction between systems based on the mechanism that drives their evolution, which can be:

1. time-driven: The state of the system changes as time progresses, i.e., continuously (for continuous-time systems), or at every tick of the clock (for discrete-time systems).
2. event-driven: The state of the system changes due to the occurrence of an event. An event corresponds to the start or the end of an activity. In general, event-driven systems are asynchronous

and the event occurrence times are not equidistant. Typical examples of event-driven systems are manufacturing systems, telecommunication networks, parallel processing systems and logistic systems. For a manufacturing system possible events are: the completion of a part on a machine, a machine breakdown or a buffer becoming empty.

A.1.1 Models for time-driven systems

The behaviour of continuous-time time-driven systems can be described by a system of differential equations of the form

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= g(x(t), u(t)) \end{aligned} \tag{A.2}$$

where $x(t)$, $u(t)$ and $y(t)$ are the state, the input, and the output of the system at time t. Note that these descriptions are typically used for continuous-time systems.
Time-driven discrete-time (or sampled) systems can be modeled by a system of difference equations of the form

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= g(x(k), u(k)) \end{aligned} \tag{A.3}$$

where $x(k)$, $u(k)$ and $y(k)$ are the state, the input, and the output of the system at time t_k.

A.1.2 Models for event-driven systems

Automata or finite state machines are the most common models for discrete-state event-driven systems.

A.1.3 Hybrid systems

We can also have combinations of continuous and discrete states, of continuous and discrete time, or of time-driven and event-driven dynamics. The resulting systems are called hybrid systems. Hybrid systems combine the continuous behaviour evolution specified by differential equations with discontinuous changes specified by discrete event switching logic [42]. In other words: hybrid dynamical models describe systems where both analog (continuous) and logical (discrete) components are relevant and interacting [8], see Figure A.1.
A first criterion for the validity of a mathematical model involves the existence and uniqueness of solutions (given initial conditions). This property is referred to as wellposedness and is a fundamental issue for every class of dynamical systems.

A.2 Classes of Hybrid Dynamical Models

Discrete-time models can formulated as

$$x(k+1) = f(x(k), u(k), w(k)) \tag{A.4a}$$

$$y(t) = g(x(k), u(k), w(k)) \tag{A.4b}$$

$$0 \leq h(x(k), u(k), w(k)) \tag{A.4c}$$

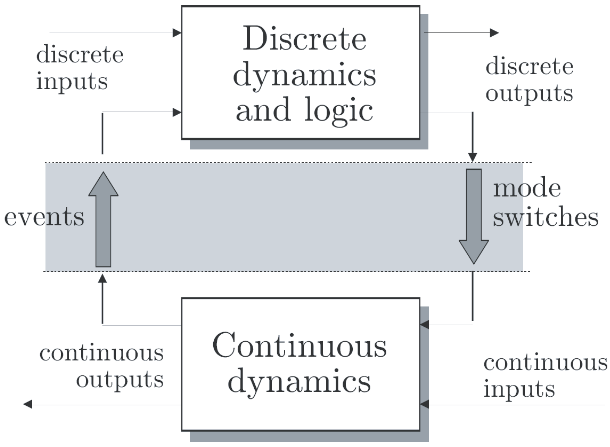


Figure A.1: Hybrid systems. Logic-based discrete dynamics and continuous dynamics interact through events and mode switches

where the variables $u(k) \in \mathbb{R}^m$, $x(k) \in \mathbb{R}^n$ and $y(k) \in \mathbb{R}^l$ denote the input, state and output, respectively, at time k, and $w(k) \in \mathbb{R}^r$ is a vector of auxiliary variables (this notation also holds for all hybrid models introduced later), $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^r \mapsto \mathbb{R}^n$, $g : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^r \mapsto \mathbb{R}^p$, $h : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^r \mapsto \mathbb{R}^q$, and the inequality Eq. (A.4a) should be interpreted componentwise. The evolution of system Eq. (A.4) is determined as follows: given the current state $x(k)$ and input $u(k)$ the collection of inequalities Eq. (A.4c) is solved for $w(k)$. By substitution of $w(k)$ in Eq. (A.4a) - Eq. (A.4b), the state update $x(k+1)$ and the current output $y(k)$ are obtained. Specific choices of the form of the functions f, g, h will determine different classes of hybrid systems, as we will detail in the rest of this section.

A.2.1 Piecewise Affine (PWA) Systems

Piecewise affine (PWA) systems are described by

$$\begin{array}{rcl} x(k+1) & = & A^i x(k) + B^i u(k) + f^i \\ y(t) & = & C^i x(k) + D^i u(k) + g^i \end{array} \quad \text{for} \quad \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \Omega_i \tag{A.5}$$

for $i = 1, \cdots, N$ where $\Omega_1, \cdots, \Omega_N$ are convex polyhedra (i.e. given by a finite number of linear inequalities) in the input/state space with non-overlapping interiors. The variables $u(k) \in \mathbb{R}^m$, $x(k) \in \mathbb{R}^n$ and $y(k) \in \mathbb{R}^l$ denote the input, state and output, respectively, at time k. PWA systems have been studied by several authors (see [16], including references herein) as they form the “simplest” extension of linear systems that can still model non-linear and non-smooth processes with arbitrary accuracy and are capable of handling hybrid phenomena.

A.2.2 Mixed Logical Dynamical (MLD) Systems

The MLD framework allows specifying the evolution of continuous variables through linear dynamic equations, of discrete variables through propositional logic statements and automata, and the mutual interaction between the two worlds [9]. The key idea of the approach consists of embedding the logic part in the state equations by transforming Boolean variables into 0-1 integers, and by expressing the relations as mixed-integer linear inequalities. We assume that system in Equations Eq. (A.6) is completely well-posed, which means in words that for all x, u within a bounded set the variables δ, z

are uniquely determined.
The general MLD form of a hybrid system is

$$x(t+1) = Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \tag{A.6a}$$
$$y(t) = Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \tag{A.6b}$$
$$E_2\delta(t) + E_3z(t) \leq E_1u(t) + E_4x(t) + E_5 \tag{A.6c}$$

A.2.3 Linear Complementarity (LC) Systems

Linear Complementarity (LC) systems are given by the equations (in discrete time):

$$x(k+1) = Ax(k) + B_1u(k) + B_2w(k) \tag{A.7a}$$
$$y(k) = Cx(k) + D_1u(k) + D_2w(k) \tag{A.7b}$$
$$v(k) = E_1x(k) + E_2u(k) + E_3w(k) + g_4 \tag{A.7c}$$
$$0 \leq v(k) \perp w(k) \geq 0 \tag{A.7d}$$

with $v(k), w(k) \in (R)^s$ and where $\perp$ denotes the orthogonality of vectors. $v(k)$ and $w(k)$ are complementarity variables.

A.2.4 Equivalence of hybrid model classes

In [7] it is shown that the above subclasses of hybrid systems are equivalent. Some of the equivalences were obtained under additional assumptions related to well-posedness (i.e., existence and uniqueness of solution trajectories) and boundedness of (some) system variables. These results are important, as they allow to transfer all the analysis and synthesis tools developed for one particular class to any other equivalent subclasses of hybrid systems.
In [8] it is also shown that linear or hybrid plants in closed-loop with a model predictive control (MPC) controller based on a linear model, fulfilling linear constraints on input and state variables, and utilizing a quadratic cost criterion, is a subclass of any of the five classes of hybrid systems.

Appendix B

Model Predictive Control

B.1 Introduction

Model Predictive Control (MPC) or Receding Horizon Control (RHC) is widely adopted in industry as an effective means to deal with multivariable constrained control problems [18, 39]. The main idea of MPC is to choose the control action by repeatedly solving on-line an optimal control problem. This aims at minimizing a performance criterion over a future horizon and yields an optimal control sequence, possibly subject to constraints on the manipulated inputs and outputs, where the future behaviour over a specified time horizon, is computed according to a model of the plant. This future behaviour is usually called the prediction horizon and is denoted by N_y as depicted in Figure B.1. At each discrete-time instant k, the measured variables and the process model (linear, nonlinear or hybrid) are used to (predict) calculate the future behaviour of the controlled plant [8]. This is achieved by considering a future control scenario, which is usually called control horizon and is denoted by N_c (see Figure B.1), as the input sequence applied to the process model, which must be calculated such that certain desired constraints and objectives are fulfilled. The first control in this sequence is applied to the plant. At the next time step the computation of the optimization is repeated starting form the new state and over a shifted horizon, leading to a moving horizon policy.

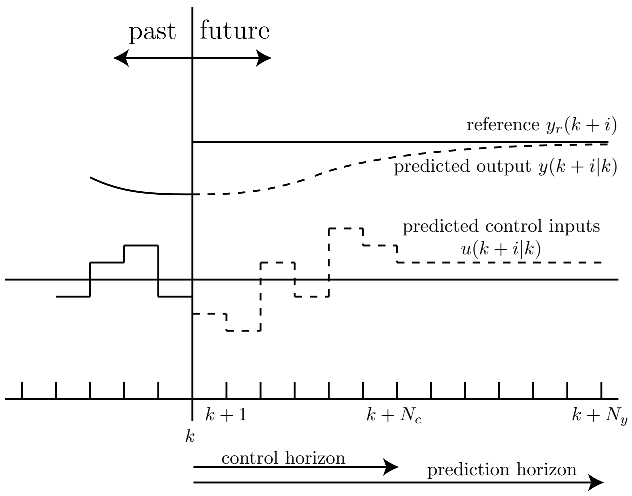


Figure B.1: A graphical illustration of Model Predictive Control

This is the main difference from conventional control which uses a pre-computed control law. An important advantage of this type of control is its ability to cope with hard constraints on controls and states. Nearly every application imposes constraints; actuators are naturally limited in the force (or equivalent) they can apply, safety limits states such as temperature, pressure and velocity and efficiency often dictates steady-state operation close to the boundary of the set of permissible states. The prevalence of hard constraints is accompanied by a dearth of control methods for handling them, despite a continuous demand from industry that has had, in their absence, to resort often to ad hoc methods. Model predictive control is one of few suitable methods, and this fact makes it an important tool for the control engineer [39].
In Figure B.2 a schematic overview of the algorithm is depicted. In the following sections the different parts will be described in detail.

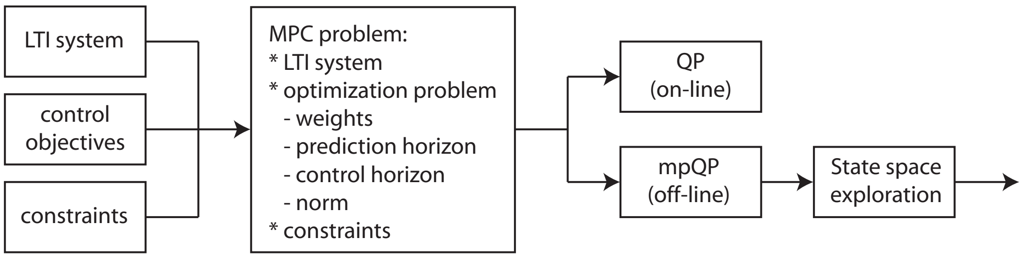


Figure B.2: Schematic overview of the MPC algorithm

For linear discrete-time systems and with linear constraints only, the MPC problems boils down to a convex Quadratic Programming (QP) problem, which can be solved efficiently. In this case the solution can be computed off-line and reduces to the simple evaluation of an explicitly defined piecewise affine function, see Section B.4.

B.2 Linear time invariant system

Consider the problem of regulating the discrete-time Linear Time Invariant (LTI) system

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{cases} \qquad k \geq 0 \tag{B.1}$$

while fulfilling the constraints

$$\begin{aligned} y_{min} &\leq y(k) \leq y_{max} \\ u_{min} &\leq u(k) \leq u_{max} \end{aligned} \tag{B.2}$$

at all time instants $k \geq 0$. In Eq. ($B.1$) - Eq. ($B.2$), $x(k) \in \mathbb{R}^{n\times 1}$, $u(k) \in \mathbb{R}^{m\times 1}$, and $y(t) \in \mathbb{R}^{p\times 1}$ are the state, input and output vectors. $A \in \mathbb{R}^{n\times n}$ is the system matrix, $B \in \mathbb{R}^{n\times m}$ the input matrix and $C \in \mathbb{R}^{p\times n}$ the output matrix. $y_{min} \leq y_{max}$ ($u_{min} \leq u_{max}$) are $p(m)$-dimensional vectors and the pair (A, B) is stabilizable.

The resulting system is a constrained discrete-time linear time-invariant system. Based on [8] the (robust) stability analysis, well-posedness results and safety analysis tools available for any of the mentioned classes of hybrid systems (see Section A.2) can be applied to any combination of a linear MPC controller and a linear system (possibly including disturbances and model uncertainties).

B.3 Model predictive control

B.3.1 Cost criterion and constraints

Model predictive control solves a constrained regulation problem in the following way. Assume that a full measurement of the state $x(k)$ is available at the current time k. In a Quadratic Program (QP), a convex quadratic function is minimized over a polyhedron, as illustrated in Figure B.3.

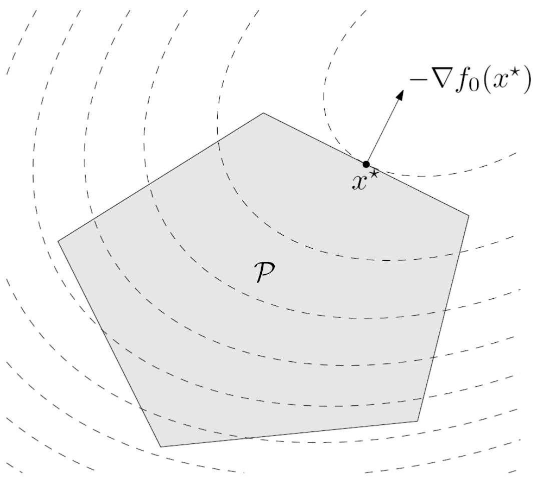


Figure B.3: Geometric illustration of QP. The feasible set $\mathcal{P}$, which is a polyhedron, is shown shaded. The contour lines of the objective function, which is convex quadratic, are shown as dashed curves. The point x^* is optimal. [14]

Then, the optimization problem consisting of the QP cost criterion with constraints is defined as

$$\begin{aligned}
\min_{U \triangleq \{u_k,\dots,u_{k+N_u-1}\}} & \left\{ J(U, x(k)) = x^T_{k+N_y|k} Q_f x_{k+N_y|k} + \sum_{k=0}^{N_y-1} \left[x^T_{k+t|k} Q x_{k+t|k} + u^T_{k+t} R u_{k+t} \right] \right\} \\
\text{subject to} \quad & y_{min} \leq y_{k+t|k} \leq y_{max}, \quad t = 1, \cdots, N_c, \\
& u_{min} \leq u_{k+t|k} \leq u_{max}, \quad t = 1, \cdots, N_c, \\
& x_{k|k} = x(k), \\
& x_{k+t+1|k} = A x_{k+t|k} + B u_{k+t}, \quad t \geq 0 \\
& y_{k+t|k} = C x_{k+t|k}, \quad t \geq 0 \\
& u_{k+t} = K x_{k+t|k}, \quad N_u \leq t < N_y
\end{aligned} \tag{B.3}$$

and is solved at each time k, where $x_{k+t|k}$ denotes the predicted state vector at time $k+t$, obtained by applying the input sequence $u_k, \cdots, u_{k+t-1}$ to model Eq. ($B.1$) starting form the state $x(k)$. In Eq. ($B.3$), it is assumed that $Q = Q' \succcurlyeq 0$, $R = R' \succcurlyeq 0$, $Q_f \succcurlyeq 0$, $(Q^{1/2}, A)$ detectable (for instance, $Q = C'C$ with (C, A) detectable), K is some feedback gain, N_y, N_u, N_c are the output, input and constraint horizons, respectively, with $N_u \leq N_y$ and $N_c \leq N_y - 1$. With a fixed prediction horizon N this problem is called a Constrained Finite Time Optimal Control (CFTOC) problem. Note that Q_f weights the final term of the horizon, $x(t + N_y)$, and is usually referred to as a terminal weight.

B.3.2 Tuning

The parameters N_y, N_u, N_c, Q and R are the basic MPC tuning parameters. The prediction horizon N_p is related to the length of the step response of the process. The time interval (1,N_p) should contain the crucial dynamics of the process. An important effect of a small control horizon N_c is the smoothing of the control signal. The control signal is then rapidly forced towards its steady-state value, which is important for stability properties. Another important consequence of decreasing N_c is the reduction in computational effort, because the number of optimization parameters is reduced. A large control horizon N_c results in faster disturbance rejection.
The weight matrices Q and R can be freely chosen as tuning parameters to give a trade off between performance and ride comfort and to obtain smooth naturalistic vehicle behaviour. Q is the state weighting matrix and R is the control weighting matrix.

B.3.3 Feasibility and stability

The main problem of MPC is that is does not, in general, a priori guarantee stability. Furthermore, MPC might drive the state to a part of the state space where no solution to the finite time optimal control problem satisfying the constraints exists. In order for MPC to be applicable, stability and feasibility must be proven.

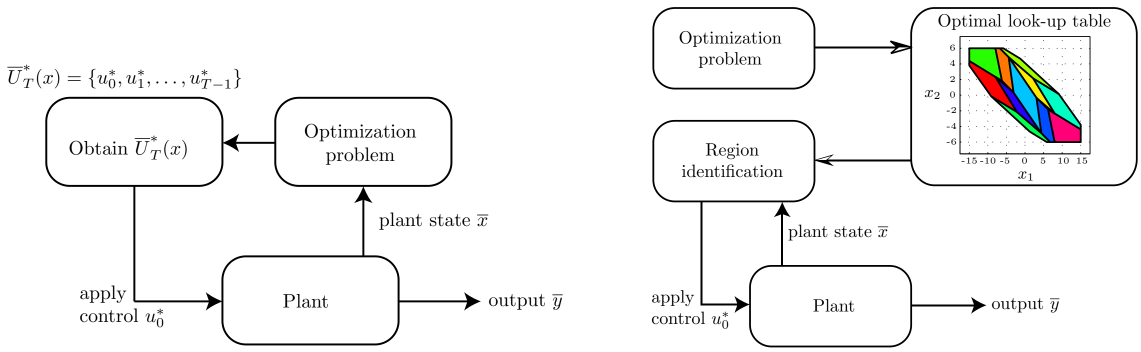
When $N_c < \infty$, there is no guarantee that the optimization problem will remain feasible at all future time steps t. For controllers for which no feasibility guarantee can be given a priori, an invariant subset of the controller must be calculated, such that constraints satisfaction is guaranteed for all time.

To show stability, setting $N_c = \infty$ leads to an optimization problem with an infinite number of constraints, which is impossible to handle because of computational complexity. Another way which results in a stable control problem is the addition of an end constraint. This constraint tend to result in in small terminal sets. This inadvertently leads to large horizons if the distance between the initial state and the terminal state is large and the input is bounded [26].
The main method of dealing with stability, after optimization with no a priori stability guarantee, in MPC is recognized to be a Lyapunov analysis. This analysis is necessitated as the underlying systems are constrained, and therefore nonlinear.

The closed-loop system with MPC Controller is globally asymptotically stable if and only if the optimization problem is feasible [59, Theorem 1].

B.4 Hybrid control / explicit MPC

Although control based on on-line optimization model predictive control has long been recognized as the winning alternative for constrained multivariable systems, its applicability has been limited to slow systems, such as chemical processes, where large sampling times allow to solve large optimization problems each time new measurements are collected from the plant (see Figure B.4(a)). The optimization problem can be solved offline (see Figure B.4(b)) for all the expected measurement values through multiparametric solvers, see [2], [12], [25] and [50]. This is due to the possibility of stating constrained MPC problems as multi-parametric Quadratic Program (mpQP), which has allowed computationally efficient explicit solutions to problems which previously required computationally demanding real-time optimization [49]. For discrete time linear time invariant systems with constraints



(a) Online MPC

(b) Offline MPC

Figure B.4: Online MPC versus offline MPC (explicit solution)

on inputs and states, the mpQP solver determines explicitly the state feedback control law which minimizes a quadratic performance criterion. The resulting feedback controller inherits all the stability and performance properties of MPC. Moreover, such an explicit form of the controller provides additional insight for better understanding of the control policy of MPC. It is shown that the form of this quadratic program is maintained for various practical extensions of the basic setup, for example, trajectory following, suppression of disturbances, time-varying constraints and also the output feedback problem [10]. On-line computation is reduced to a simple function evaluation, instead of the on-line calculation of the extensive quadratic program. The implementation of off-line MPC enlarge the scope of applicability of MPC to small-size/fast-sampling applications. The optimal solution is a piecewise affine function of the state.

As the coefficients of the linear term in the cost function and the right-hand side of the constraints depend linearly on the current state, the quadratic program can be viewed as a multi-parametric quadratic program (mpQP). Parametric programming is defined as the characterization of the solution for a full range of parameter values is sought. In the jargon of operations research, programs which depend only on a scalar parameter are referred to as parametric programs, while problems depending on a vector of parameters as multi-parametric programs [10].

There are also some disadvantages of using explicit solutions to MPC problems compared to using the more conventional method with on-line solution of an optimization problem. The most obvious disadvantage is the rapid growth of size in the explicit solution as the problem size increases. This limits the use of these solutions to small problems. This limitation is primarily due to the on-line memory requirements becoming too high [49].
Another limitation of off-line methods is that the look-up table is only valid for a specific configuration of the system. Thus these methods are not able to handle possible variation of the system parameters values, and a recomputation of the table may be required.

In the next subsections, the mpQP-algorithm will be treated to implement an explicit MPC controller. In Appendix C more information can be found on QP as well as Linear Programming (LP) and the introduction of integers, leading to a Mixed Integer Linear Program (MILP) or Mixed Integer Quadratic Program (MIQP) (both online). The explicitly designed variants are called: multi-parametric Mixed Integer Linear Program (mpMILP) and multi-parametric Mixed Integer Quadratic Program (mpMIQP).

B.4.1 Quadratic program (rewritten)

Substitute Eq. ($B.4$) in Eq. ($B.3$).

$$x_{k+t|k} = A^k x(k) + \sum_{j=0}^{t-1} A^j B u_{k+t-1-j} \tag{B.4}$$

The result is an optimization problem Eq. ($B.6$) which is a Quadratic Program (QP). This program depends on the current state $x(t)$ and past input $u(t) = u_{t-1}$.

$$\begin{aligned} V(x(t)) \quad = \quad & \min_U \frac{1}{2}U'HU + x(t)'FU \\ & \text{subjected to} \quad GU \leq W + Sx(t) \end{aligned} \tag{B.5}$$

where $x(t) \triangleq [x'(t)]'$, $H = H' \succcurlyeq 0$, and H, F, Y, G, W, S are obtained from Eq. ($B.3$). Solve mp-QP offline to find optimal solution $U_t^* = U^*(x(t))$. Optimal input given by

$$u(t) = [I\ 0\ \ldots\ 0]U^*(x(t)) \tag{B.6}$$

Transforming to an equivalent problem by defining

$$z \triangleq U + H^{-1}F'x(t) \tag{B.7}$$

delivers

$$\begin{aligned} V_z(x) \quad = \quad & \min_z \frac{1}{2}z'Hz \\ & \text{subjected to} \quad Gz \leq W + Sx(t) \end{aligned} \tag{B.8}$$

where $S \triangleq E + GH^{-1}F'$ and $V_z(x) = V(x) - \frac{1}{2}x'(Y - FH^{-1}F')x$.
In order to start solving the mpQP problem, an initial vector x_0 inside the polyhedral set $\mathcal{X} = \{x : Tx \leq Z\}$ of parameters is needed over which the problem can be solved such that the QP problem Eq. ($B.8$) is feasible for $x = x_0$. A good choice for x_0 is the center of the largest ball contained in $\mathcal{X}$ (x_0 will be a projection of the Chebychev center) for which a feasible z exists, determined by solving the Linear Program (LP)

$$\begin{aligned} & \max_{x,z,\epsilon} && \epsilon, \\ & \text{subjected to} && T^i x + \epsilon\|T^i\| \leq Z^i, \\ & && Gz - Sx \leq W \end{aligned} \tag{B.9}$$

if $\epsilon \leq 0$, then the QP problem Eq. ($B.8$) is infeasible for all x in the interior of $\mathcal{X}$. Otherwise, $x = x_0$ and the QP problem Eq. ($B.8$) can be solved in order to obtain the corresponding optimal solution z_0. Such a solution is unique, because $H \succcurlyeq 0$, and therefore uniquely determines a set of active constraints $\tilde{G}z_0 = \tilde{S}x_0 + \tilde{W}$ out of the constraints in Eq. ($B.8$).

B.4.2 Constrained optimization: Karush-Kuhn-Tucker conditions

It is possible to use the Karush-Kuhn-Tucker conditions to obtain an explicit representation of the optimizer $U^*(k)$ for QP Eq. ($B.4$) which are necessary and sufficient for optimality of $U^*(k)$ and assuming linearly independent active constraints. This optimizer $U^*(k)$ is valid in some neighborhood of x_0.

$$Hz + G^T\lambda = 0, \quad \lambda \in \mathbb{R}^q \tag{B.10a}$$
$$\lambda_i(Gz - W^i - S^i x) = 0, \quad i = 1, \dots, q, \tag{B.10b}$$
$$\lambda \geq 0, \tag{B.10c}$$
$$Gz \leq W + Sx \tag{B.10d}$$

where the superscript i denotes the ith row. Solving Eq. ($B.10a$) for z,

$$z = -H^{-1}G^T\lambda \tag{B.11}$$

and substitute the result into Eq. ($B.10b$) to obtain the complementary slackness condition $\lambda(-GH^{-1}G^T\lambda - W - Sx) = 0$.

Let $z^*(x)$ be the optimal solution to Eq. ($B.8$) for a given x. We define 'active constraints' the constraints with $G^i z^*(x) - W^i - S^i x = 0$, and 'inactive constraints' the constraints with $G^i z^*(x) - W^i - S^i x < 0$. The 'optimal active set $\mathcal{A}^*(x)$' is the set of indices of active constraints at the optimum, $\mathcal{A}^*(x) = \{i | G^i z^*(x) = W^i + S^i x\}$.

Assume for the moment that the set $\mathcal{A}$ of constraints that are active at the optimum for a given x is known. Now, the matrices $G^{\mathcal{A}}$, $W^{\mathcal{A}}$, $S^{\mathcal{A}}$ and the Lagrange multipliers $\lambda^{\mathcal{A}} \geq 0$, corresponding to the optimal active set $\mathcal{A}$ can be formed.

For active constraints, $-G^{\mathcal{A}}H^{-1}(G^{\mathcal{A}})^T\lambda^{\mathcal{A}} - W^{\mathcal{A}} - S^{\mathcal{A}}z = 0$, and therefore,

$$\lambda^{\mathcal{A}} = -(G^{\mathcal{A}}H^{-1}(G^{\mathcal{A}})^T)^{-1}(W^{\mathcal{A}} + S^{\mathcal{A}}x) \tag{B.12}$$

$(G^{\mathcal{A}}H^{-1}(G^{\mathcal{A}})^T)^{-1}$ exists because the rows of $G^{\mathcal{A}}$ are linearly independent. Thus, $\lambda^{\mathcal{A}}$ is an affine function of x. Substitution of $\lambda^{\mathcal{A}}$ from Eq. ($B.12$) into Eq. ($B.11$) delivers

$$z = H^{-1}(G^{\mathcal{A}})^T(G^{\mathcal{A}}H^{-1}(G^{\mathcal{A}})^T)^{-1}(W^{\mathcal{A}} + S^{\mathcal{A}}x) \tag{B.13}$$

and note that z is also an affine function of x.

For a given optimal active set $\mathcal{A}$ and a fixed x, the solution to Eq. ($B.8$) is characterized. However, as long as $\mathcal{A}$ remains the optimal active set in a neighborhood of x, the solution Eq. ($B.13$) remains optimal, when z is viewed as a function of x. Such a neighborhood where $\mathcal{A}$ is optimal is determined by imposing that z must remain feasible [10, Theorem 2]:

$$GH^{-1}(G^{\mathcal{A}})^T(G^{\mathcal{A}}H^{-1}(G^{\mathcal{A}})^T)^{-1}(W^{\mathcal{A}}+S^{\mathcal{A}}x)\leq W+Sx \tag{B.14}$$

and by Eq. ($B.10c$), the Lagrange multipliers in Eq. ($B.12$) must remain non-negative

$$-(G^{\mathcal{A}}H^{-1}(G^{\mathcal{A}})^T)^{-1}(W^{\mathcal{A}}+S^{\mathcal{A}}x)\geq 0 \tag{B.15}$$

as x is varied. After removing the redundant inequalities from Eq. ($B.14$) and Eq. ($B.15$), a compact representation of CR_0 is obtained. CR_0 is a polyhedron in the x-space and represents the largest set of $x \in \mathcal{X}$ such that the combination of active constraints at the minimizer remains unchanged. Once the the critical region is defined, the rest of the space $CR^{rest} = \mathcal{X} - CR_0$ has to be explored and new critical regions generated. The argument Eq. ($B.12$)-Eq. ($B.15$) is repeated in each new region, until the whole x-space has been covered.

The recursive algorithm of [10] can be briefly summarized as follows: Solve the linear program (Eq. ($B.9$)) to find a feasible parameter $x_0 \in \mathcal{X}$, where $\mathcal{X}$ is the range of parameters for which the mpQP is to be solved. Solve the QP Eq. ($B.8$) with $x = x_0$, to find the optimal active set $\mathcal{A}$ for x_0, and then use Eq. ($B.12$)-Eq. ($B.15$) to characterize the solution and critical region CR_0 corresponding to $\mathcal{A}$. Then divide the parameter space as in Figure B.5b-c by reversing one by one the hyperplanes defining the critical region. Iteratively subdivide each new region R_i in a similar way as was done with $\mathcal{X}$.

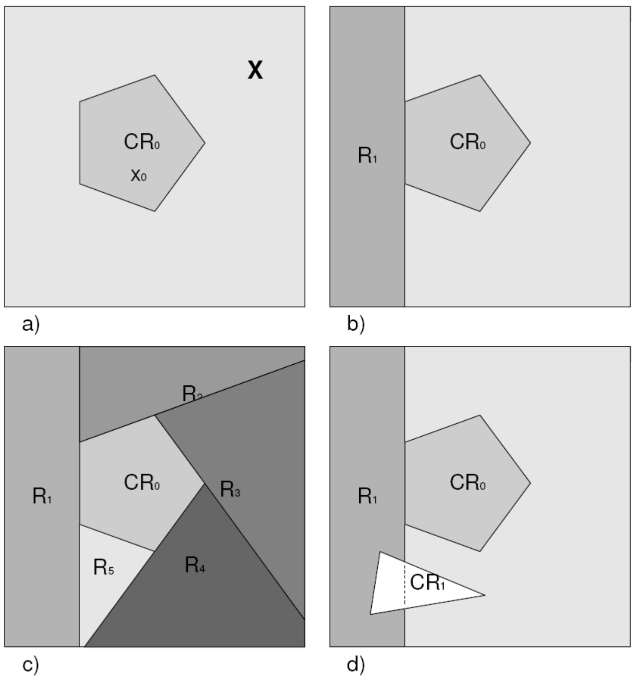


Figure B.5: State space exploration strategy of [10]

Thus, this algorithm gives partitioning of feasible regions into polyhedra and inherits properties of optimization problem.

B.4.3 Switching linear state feedback solution

This approach to handle constraints leads to a switching controller, i.e., a controller that uses different feedback laws in different domains of the state-space. By tuning the feedback laws so that the constraints are guaranteed to hold in each domain, constraints are handled in a simple and attracting way. The problem is to determine how to partition the state-space and select the feedback law in order to guarantee stability and obtain good performance. A simple choice is to use linear feedback laws in each of the domains.

Depending on the dimension of the state x and the size of N_u, i.e. number of degrees of freedom in the MPC optimization, the state-space is divided into R state-dependent regions $R_i(x(t)), i = 1, ..., R$ for which an explicit optimal control law is computed. The result is stored in a look-up table, comprising a state dependent part $F(R_i, x(t)) \in \mathbb{R}^{R \times \text{entries of } x}$ and a constant part $g(R_i) \in \mathbb{R}^{R \times 1}, i = 1, ..., R$. With this setup, a state feedback solution is obtained which is continuous and piecewise affine, i.e. a partition of neighbouring polytopes in the $\mathcal{X}$-space.

$$\begin{aligned} u(t, R_i) &= F_i\, x(t) + g_i, \quad \text{for} \\ x(t) &\triangleq \{x : H_i\, x \le S_i\}, \quad i = 1, \ldots, R \end{aligned} \tag{B.16}$$

The state space is divided in: 1. unconstrained areas 2. constrained areas. The control output of the unconstrained areas depends on the state vector with different gains for each area and a constant. The right part in Eq. ($B.17$) F_i multiplied with the state vector x and the constant g_i represents this behaviour.
The output of the constrained areas is independent of the state vector: only the constraints on the control output influences the control output: g_i in Eq. ($B.17$) will be nonzero (F_i is zero).

B.5 Multi-parametric Toolbox

The algorithm presented in the preceding sections is implemented in the Multi-parametric Toolbox (MPT), created by the Swiss Federal Institute of Technology (ETH Zurich).
MPT is a Matlab toolbox for multi-parametric optimization and computational geometry. MPT is an easy-to-use software tool for computation, analysis and visualization of explicit control laws for linear and Piecewise Affine (PWA) systems. It provides a large variety of different optimization procedures, ranging from cost-optimal solutions, through time-optimal strategies to computation of controllers of low complexity. The toolbox can easily deal with uncertain dynamical systems (parametric uncertainties and additive disturbances) and provide robustly stabilizing feedback control laws [32].

Appendix C

Mathematical programming

C.1 Overview

The term 'mathematical programming' refers to the study of problems in which one seeks to minimize or maximize a real function by systematically choosing the values of variables from within an allowed set, i.e. optimization problems. The term 'programming' predates computers and means 'preparing a schedule of activities' in this context. An overview of the most important types of mathematical programming is given next.

- Linear/Quadratic Programming (LP/QP)
  Refers to the minimization of a linear (1- or ∞-norm) or quadratic (2-norm) cost function subject to linear constraints, where all variables are continuous. The standard linear program is formulated as follows

$$\begin{aligned} \min_x \quad & z = c^T x \\ \text{subjected to} \quad & Ax \leq b \end{aligned} \tag{C.1}$$

  with A the coefficient matrix for the linear constraints and b the upper bounds on the constraints. The standard quadratic program is formulated as follows

$$\begin{aligned} \min_x \quad & z = \frac{1}{2} x^T H x \\ \text{subjected to} \quad & Ax \leq b \end{aligned} \tag{C.2}$$

- Mixed-Integer Linear/Quadratic Programming (MILP/MIQP)
  Refers to the minimization of a linear/quadratic cost function subject to linear constraints, where some or all variables are restricted to be binary (integers), e.g. hybrid models. This requires special solving algorithms such as *branch-and-bound*. The standard MILP is formulated as follows

$$\begin{aligned} \min_{x,y} \quad & z = c^T x + d^T y \\ \text{subjected to} \quad & Ax + Ey \leq b \\ & x \in \mathbb{R}^{n \times 1} \\ & y \in \{0,1\}^m \end{aligned} \tag{C.3}$$

Sometimes also the abbreviation MIP is used. This refers to both MILP and MIQP.

- Multi-Parametric Linear/Quadratic Programming (mpLP/mpQP)
  An optimization problem becomes a multi-parametric problem if one or more parameters vary over a defined set. That is, the objective is to find the solution for a full range of parameter values. The standard mpLP is formulated as follows

$$\begin{aligned} \min_x \quad & z = c^T x \\ \text{subjected to} \quad & Gx \leq b + S\theta \\ & \theta_{min} \leq \theta \leq \theta_{max} \end{aligned} \tag{C.4}$$

  where x are the variables and θ the parameters.

- Multi-Parametric Mixed-Integer Linear Programming (mpMILP)
  An mpMILP problem is exactly the same as an mpLP problem except that the integer variables are present. The standard formulation is as follows

$$\begin{aligned} \min_x \quad & z = c^T x + d^T y \\ \text{subjected to} \quad & Gx \leq b + S\theta \\ & x \in \mathbb{R}^{n\times 1} \\ & y \in \{0,1\}^m \\ & \theta \in K \end{aligned} \tag{C.5}$$

  where K is the parameter space.

C.2 mpLP versus mpQP

The most intuitive and common way to implement an MPC algorithm is to use a quadratic performance index. The mpQP approach also has several advantages over the mpLP scheme. Most notably, the LP formulation of the MPC law introduces auxiliary variables which increases the complexity of the problem. As a result, the dimension of the corresponding QP problem is smaller. Moreover, it is easier to implement an iterative and more analytical exploration of the parameter space when a quadratic objective function is used. For the purpose of model predictive control of linear systems, the mpQP approach is thus more suitable.

C.3 mpMILP versus mpMIQP

In general, solving mpMIQPs is more complex than solving mpMILPs, as it involves solving Mixed-Integer Non-Linear Programs (MINLPs). This is more difficult and expensive compared to the MILPs in the mpMILP algorithms. As a result, for the control of hybrid systems the mpMILP approach is considered more suitable.

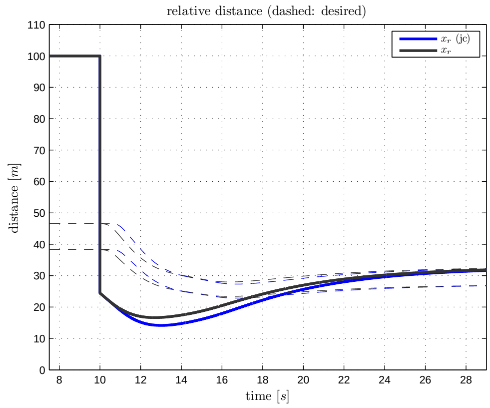
Appendix D

Switching with jerk/without jerk constraint

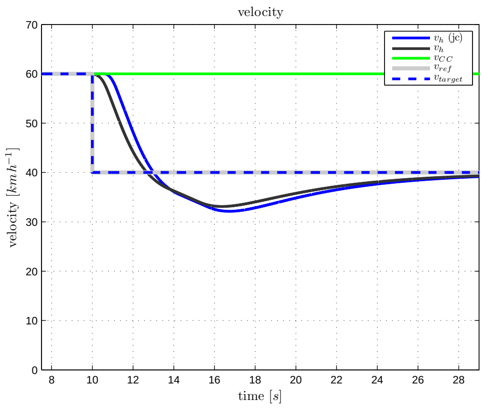
Based on a simulation of a specific situation (cut-in), comparison between switching with or without jerk constraint is possible. As a result, the controller which uses 'input switching' is the preferred controller for this application.

The reason for the existing difference between the controllers present in 'Logic' (depicted in Figure 4.6 and Figure 4.12) and the controller part (depicted in Figure 4.8) is the determination of the most important vehicle and the actual calculation of the desired accelerations. In the next example (see Figure D.1 for the simulation results) a possible situation is explained, where the distinction between the controllers is present: the desired distance is smaller than the desired distance at $t = 0$ s: $x_{r,1} > x_{r,d,1}$ and the relative velocity is zero $v_{r,1} = 0$ and equal to the desired cruise control velocity $v_h = v_{CC}$. The corresponding accelerations: $a_{h,rt,d} =$ some positive value to meet the control objective $x_r = x_{r,d}$, $a_{h,vt,d} = 0$ because $v_h = v_{CC}$.

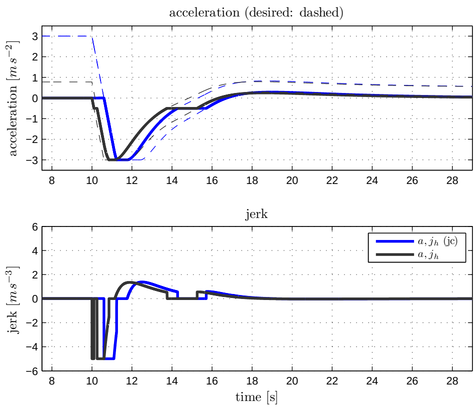
At time $t = 10$ s a cut-in occurs: $x_{r,2} < x_{r,d,2}$ & $v_{r,2} < 0$. A jerk constraint in the controllers present in 'Logic' delays the switching between the virtual and real target: $a_{h,rt,d} < a_{h,vt,d}(= 0)$: because of the jerk constraint it will take some time for $a_{h,rt,d}$ to become smaller than $a_{h,vt,d}$.
With jerk constraint: ACC accelerations (jerk constraint (jc), blue dashed lines) $a_{h,th,d}$ & $a_{h,br,d} = 3\,\mathrm{m\,s^{-2}}$ (no difference in controlling the time headway difference). Without jerk constraint: ACC acceleration: (black dashed lines) $a_{h,th,d} = .8\,\mathrm{m\,s^{-2}}$ (due to smaller t_{hw}) & $a_{h,br,d} = 0\,\mathrm{m\,s^{-2}}$.
Both systems react at the same time, however the resulting acceleration of 'Input switching' becomes earlier smaller than zero due to missing jerk constraints, resulting in a larger $x_{r,min}$ and a larger $TTC_{r,min}$, both are desired.



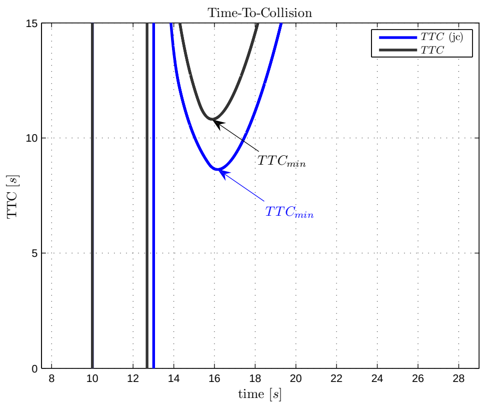
(a) Relative distance x_r



(b) Absolute velocity v_h/v_t



(c) Acceleration a and jerk j



(d) TTC

Figure D.1: $0 < t < 10$: $x_r > x_{r,d}$ & $v_r = 0$ $(v_h = v_{CC})$ $\quad t \geq 10$: $x_r < x_{r,d}$ & $v_r < 0$

Appendix E

Controller design: output versus input switching

This appendix treats the overall controller design for the ACC Stop-&-Go application. Two different structures are presented: 1. *overall control structure with controller output switching* 2. *overall control structure with controller input switching.*

E.1 Controller output switching

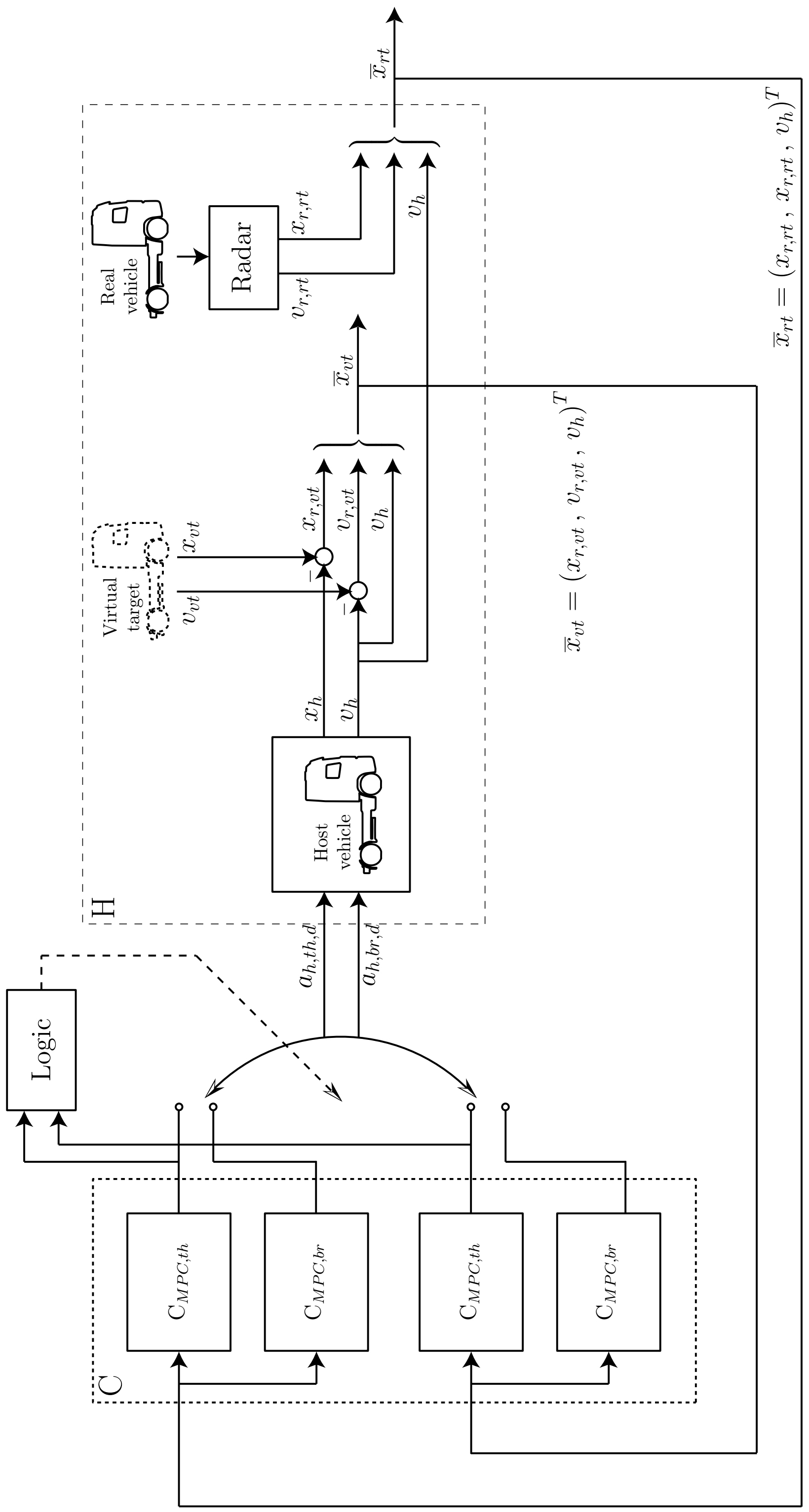


Figure E.1: Overall control structure with controller output switching

E.2 Controller input switching

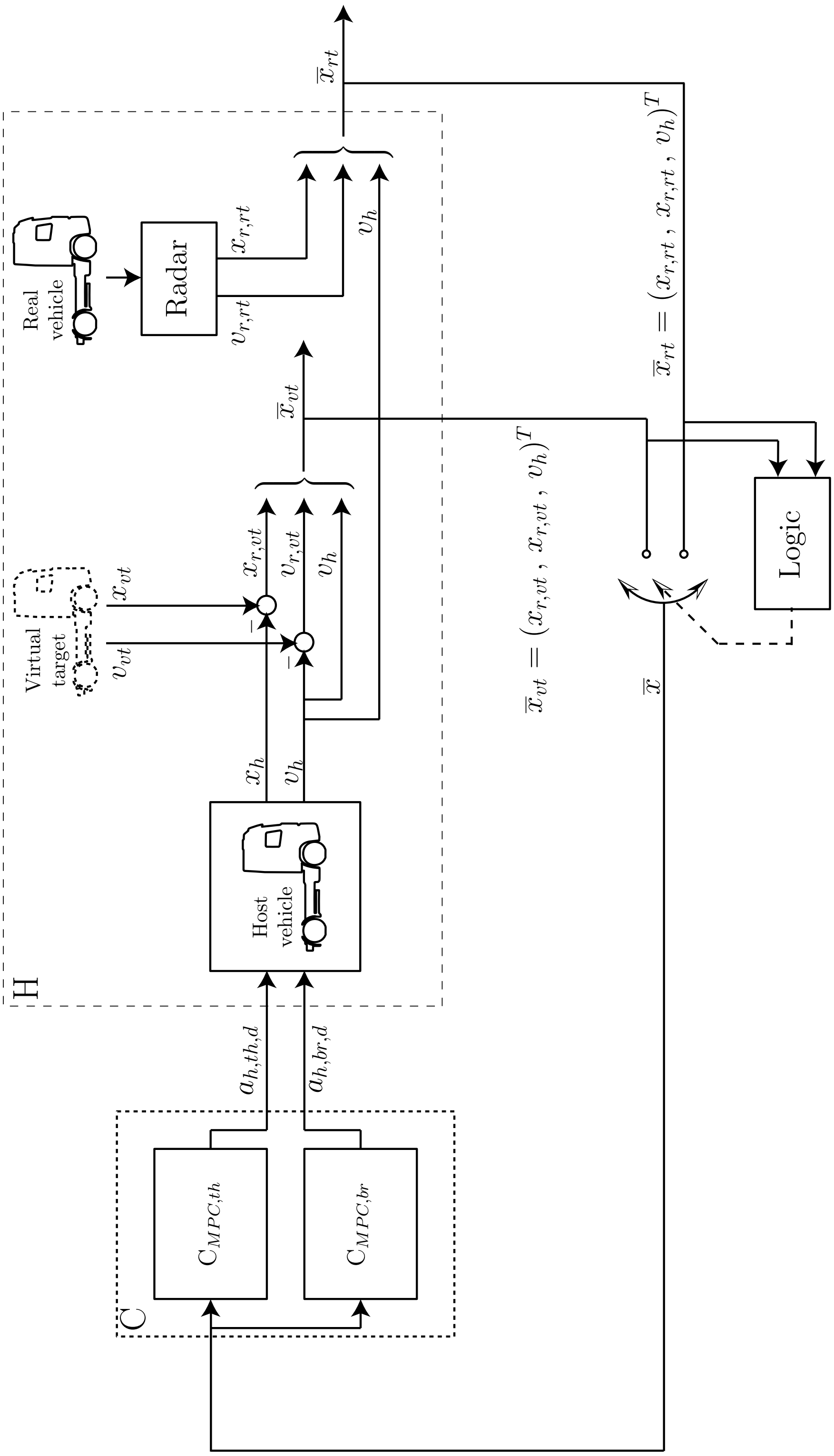
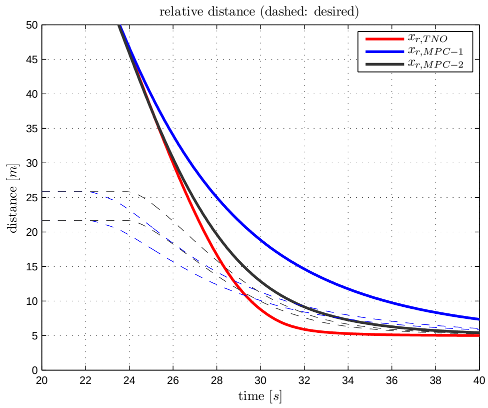


Figure E.2: Overall control structure with controller input switching (same as Figure 4.12)

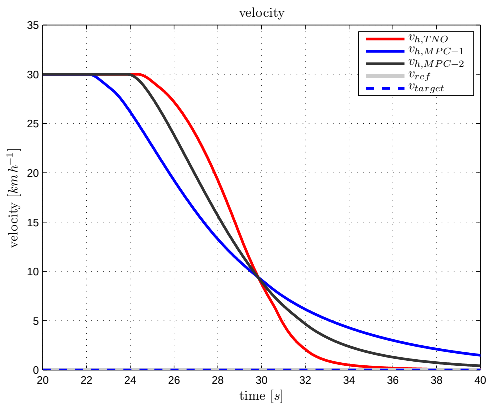
Appendix F

Simulations results

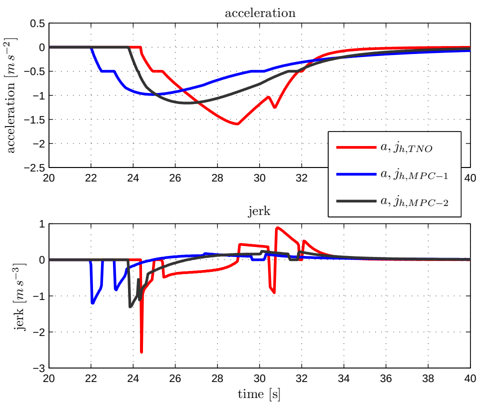
In this appendix the figures of Table 5.1 of Subsection 5.2.2 are depicted.



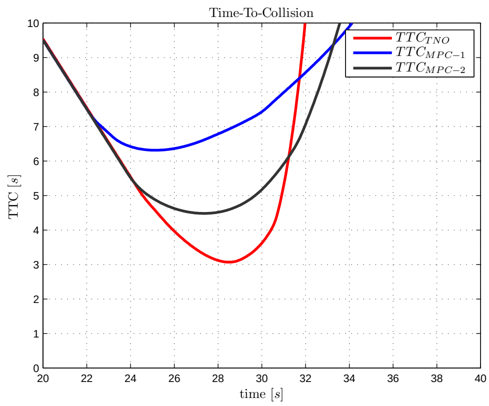
(a) Relative distance x_r



(b) Absolute velocity v_h/v_t

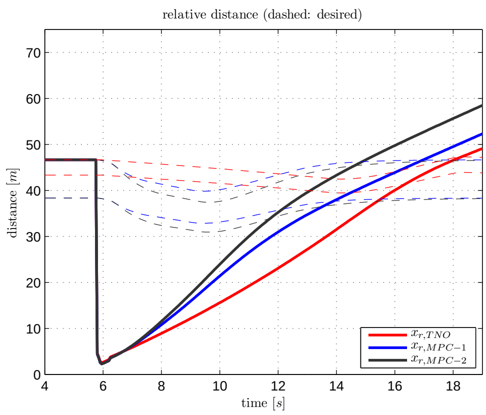


(c) Acceleration a and jerk j

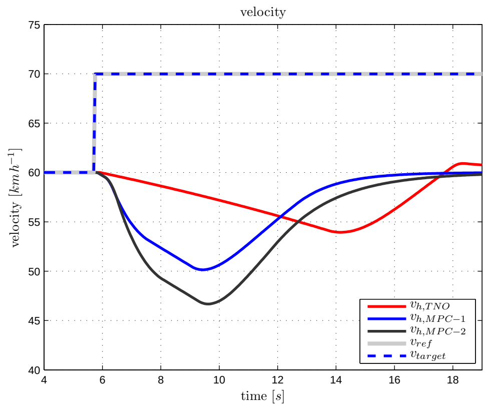


(d) TTC

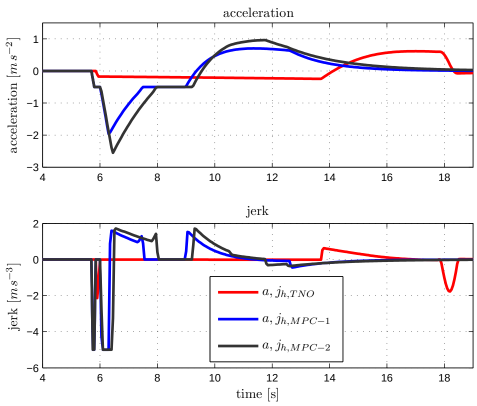
Figure F.1: Sit. 1: Approach of a standstill vehicle, yielding a CC to ACC switch



(a) Relative distance x_r

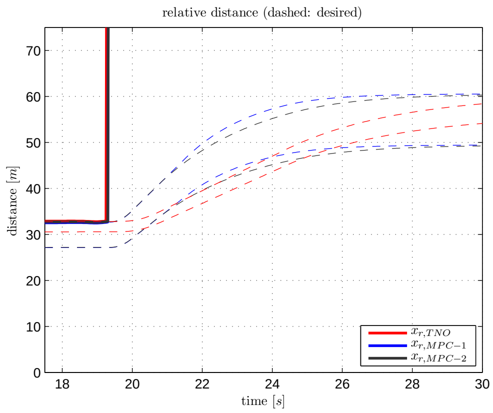


(b) Absolute velocity v_h/v_t

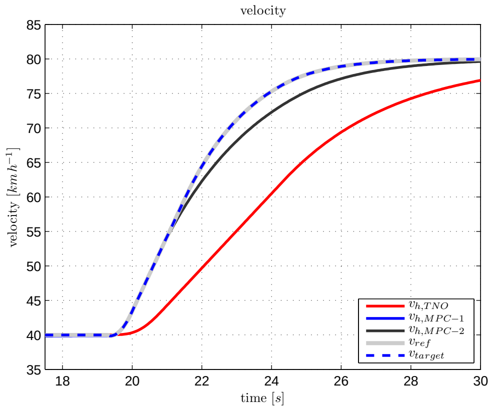


(c) Acceleration a and jerk j

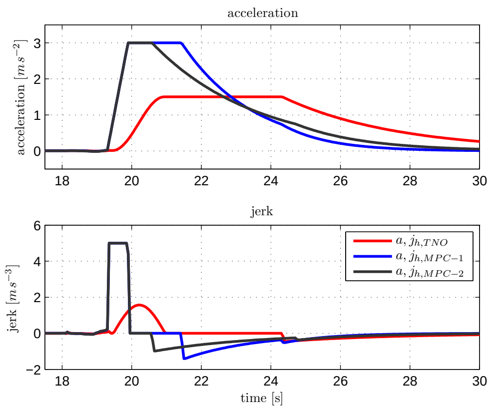
Figure F.2: Sit. 3: Cut-in with a positive velocity difference



(a) Relative distance x_r

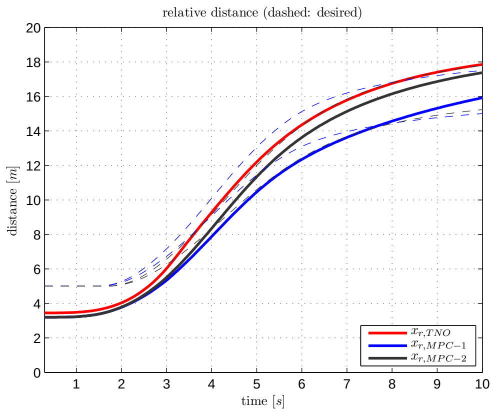


(b) Absolute velocity v_h/v_t

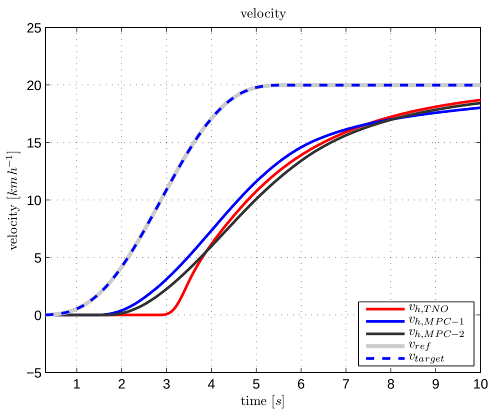


(c) Acceleration a and jerk j

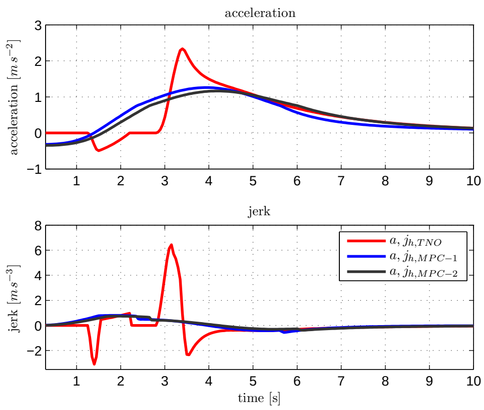
Figure F.3: Sit. 4: Cut-out of a target vehicle, yielding an ACC to CC switch



(a) Relative distance x_r

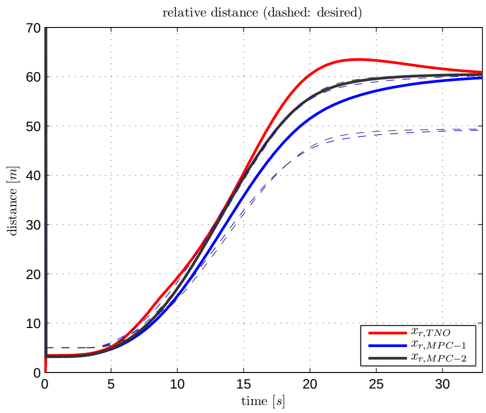


(b) Absolute velocity v_h/v_t

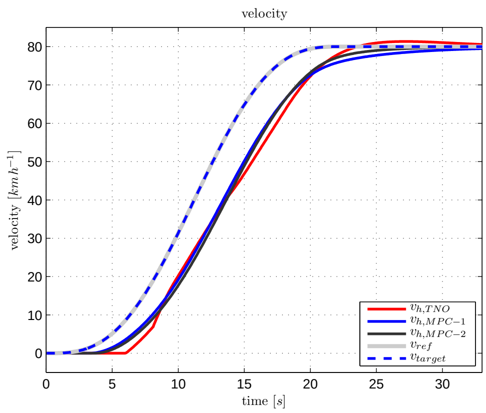


(c) Acceleration a and jerk j

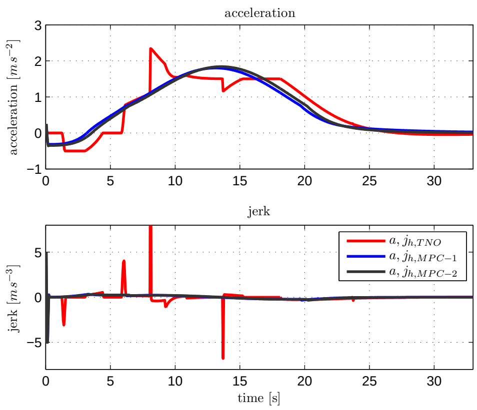
Figure F.4: Sit. 6: Accelerating at a traffic light



(a) Relative distance x_r

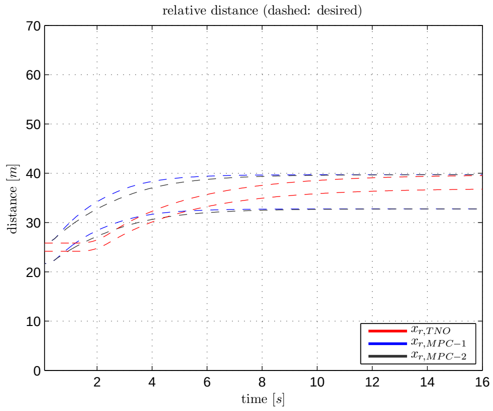


(b) Absolute velocity v_h/v_t

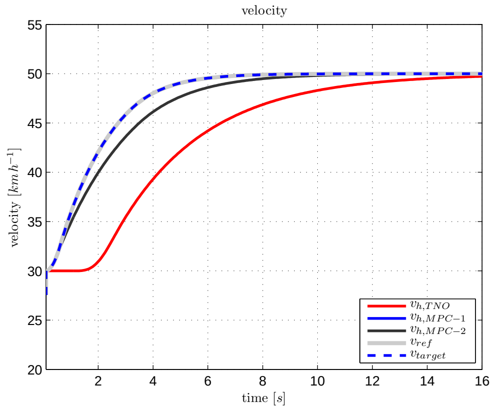


(c) Acceleration a and jerk j

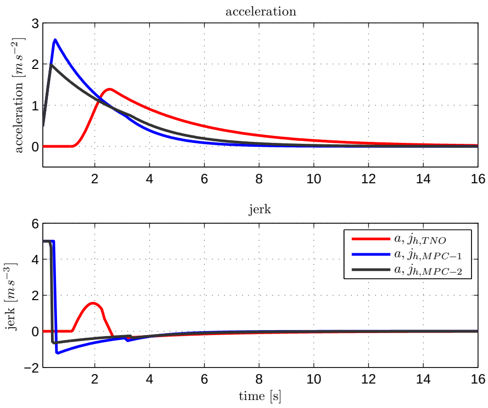
Figure F.5: Sit. 7: Accelerating at a traffic jam



(a) Relative distance x_r



(b) Absolute velocity v_h/v_t



(c) Acceleration a and jerk j

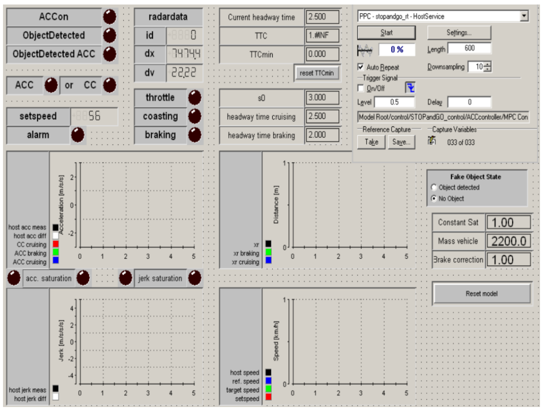
Figure F.6: Sit. 8: CC behaviour: accelerating to v_{CC}

Appendix G

CarLab

TNO Automotive has created a Car Laboratory (CarLab) for the development of advanced driver assistance and longitudinal control. For this purpose, a vehicle is equipped with several sensors for vehicle dynamics measurements and an interface for electronic braking and throttle. The CarLab environment provides all the systems needed to investigate all kinds of intelligent vehicle systems. The vehicle is equipped with a system for rapid control prototyping, dSpace AutoBox, to provide an environment which can control the engine and the brakes of the vehicle. In Figure G.1 several pictures are shown, containing the CarLab itself equipped with the FMCW radar (Figure G.1(a)), in Figure G.1(b): dSpace ControlDesk to create an experimental environment with virtual instrument panels and with the possibility of editing parameters. In Figure G.1(c) the hardware (dSpace AutoBox) of the CarLab is depicted and Figure G.1(d) shows a picture of the interieur of the CarLab with an LCD screen.

(a) CarLab on a chassis dynamometer



(b) dSpace ControlDesk interface (software)

(c) dSpace AutoBox (hardware)

(d) Interieur CarLab with LCD

Figure G.1: Overview experiments