



**TRAJECTORY GENERATION FOR OPEN/CLOSE GAP MANEUVERS IN
VEHICLE STRINGS**

ERKAM ÇANKAYA

MAY 2017

TRAJECTORY GENERATION FOR OPEN/CLOSE GAP MANEUVERS IN
VEHICLE STRINGS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY

BY
ERKAM ÇANKAYA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
ELECTRONIC AND COMMUNICATION ENGINEERING

MAY 2017

Title of the Thesis: **Trajectory Generation for Open/Close Gap Maneuvers in Vehicle Strings**

Submitted by **Erkam ÇANKAYA**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.



Prof. Dr. Can ÇOĞUN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.



Prof. Dr. Hüseyin Selçuk GEÇİM
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

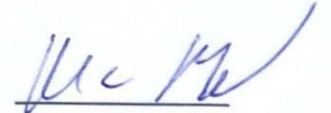


Assoc. Prof. Dr. Klaus Werner SCHMIDT
Supervisor

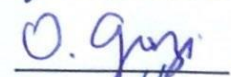
Examination Date: 29.05.2017

Examining Committee Members

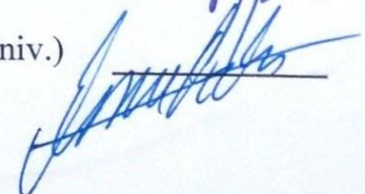
Assoc. Prof. Dr. Klaus Werner SCHMIDT (Çankaya Univ.)



Assoc. Prof. Dr. Orhan GAZI (Çankaya Univ.)



Assist. Prof. Dr. Emre ÖZKAN (Middle East Technical Univ.)



STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Erkam,ÇANKAYA

Signature :

Date :

29.05.2017

ABSTRACT

ÇANKAYA, Erkam

M.Sc., Department of Electronic and Communication Engineering

Supervisor: Assoc. Prof. Dr. Klaus Werner SCHMIDT

May 2017, 60 pages

Cooperative adaptive cruise control (CACC) is an advanced technology allowing vehicle following at a small inter-vehicle spacing. In its classical usage, CACC assumes that vehicles are arranged in the form of a vehicle string and follow each other at a velocity-dependent distance. Nonetheless, practical driving situations include the case of lane changes, where vehicles can join or leave a vehicle string. In such case, it is required that gaps for joining vehicles are provided or gaps after leaving vehicles are closed in order to ensure safe and efficient driving.

This thesis is concerned with gap opening and closing maneuvers in vehicle strings. Introducing a suitable control architecture, gap opening and closing maneuvers can be realized by the generation of feedforward input signals. To this end, the first contribution of the thesis is the development of five methods for the computation and representation of gap opening and closing trajectories that fulfill additional safety and comfort constraints. The first method is based on the solution of an optimal control problem, the second method uses a polynomial trajectory and plant inversion, the third method concatenates three polynomials and uses nonlinear programming to determine the polynomial coefficients, the fourth method uses a high-order polynomial and the fifth method uses concatenated polynomials in order to approximate the optimal control solution. A simulation study shows that the fifth

method is particularly useful in practical applications since it computes trajectories that approximate the optimal control solution in real-time. The second contribution of the thesis is the implementation of a vehicle model that realizes CACC and additional feedforward signals in the form of a Matlab S-function.

Keywords: Intelligent Transportation System, CACC, Feedforward signal



ÖZ

ÇANKAYA, Erkam

Yüksek Lisans., Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Tez Yöneticisi: Doç. Dr. Klaus Werner SCHMIDT

Mayıs 2017, 60 sayfa

Kooperatif Otomatik Seyir Kontrolü (CACC), araçlar arasındaki dar boşlukta araç takibine izin veren gelişmiş bir teknolojidir. Klasik kullanımda CACC araçların bir araç dizisi halinde düzenlendiğini ve birbirini hıza bağlı bir mesafede takip ettiğini varsayar. Bununla birlikte pratik sürüş durumları araçların bir araç dizisine katılabildiği ve ayrılabilirdiği durumları da içerir. Böyle bir durumda boşlukların, katılacak araçlar için açılması veya araçlar ayrıldıktan sonra emniyetli ve etkili bir şekilde kapanması gerekmektedir.

Bu tez araç dizilerindeki boşluk açma ve kapatma manevraları ile ilgilidir. Uygun bir kontrol mimarisinde, boşluk açma ve kapatma manevraları ileri besleme giriş sinyallerinin üretilmesi ile gerçekleştirilebilir. Bu amaca ulaşmak için, tezin ilk katkısı ek güvenlik ve konfor kısıtlamalarını karşılayan boşluk açma ve kapatma yörüngelerinin hesaplanması ve gösterimi için beş yöntem geliştirilmesidir. Birinci yöntem optimal kontrol probleminin çözümüne dayanır, ikinci yöntem bir polinom yörüngesi ve plant ters çevirmesi kullanır, üçüncü yöntem üç polinomu birbirine bağlar ve polinom katsayılarını belirlemek için doğrusal olmayan programlamayı kullanır, dördüncü yöntem yüksek dereceli bir polinom kullanır ve beşinci yöntem optimal kontrol çözümünü yakınsamak için birbirine bağlanmış polinomları kullanır. Beşinci yöntemin gerçek zamanlı olarak optimal kontrol çözümüne yaklaşan yörüngeleri hesaplaması nedeniyle pratik uygulamalarda özellikle kullanışlı

olduđunu gsteren bir simlasyon alıřması yapıldı. Tezin ikinci katkısı, CACC ve ilave ileri besleme sinyallerini bir Matlab S-fonksiyonu řeklinde gerekleřtiren bir ara modelinin uygulanmasıdır.

Anahtar Kelimeler: Akıllı Ulařtırma Sistemleri, Kooperatif Otomatik Seyir Kontrol, İleri beslemeli sinyal,



ACKNOWLEDGEMENTS

I sincerely thank to my thesis advisor Assoc. Prof. Dr. Klaus Werner SCHMIDT, for providing guidance towards my research. It has been a great learning experience working with him. I would like to express my deepest gratitude towards him for offering instructive advice at all time.

Also I thank dearest my father, my mother and my sister for supporting me and encouraging me with their best wishes throughout all my studies.

TABLE OF CONTENTS

STATEMENT OF NON PLAGIARISM.....	iii
ABSTRACT.....	Iv
ÖZ.....	Vi
ACKNOWLEDGEMENTS.....	Viii
TABLE OF CONTENTS.....	Ix
LIST OF FIGURES.....	Xi
LIST OF ABBREVIATIONS	Xiii

CHAPTERS:

1. INTRODUCTION.....	1
2. BACKGROUND.....	4
2.1. CACC and Vehicle Following	4
2.2. String Stability.....	5
2.3. Controller Design for CACC and Simulations.....	6
3. OPENING AND CLOSING GAPS FOR SAFE LANE CHANGES.....	11
3.1 Basic Maneuvers.....	11
3.2 Requirements and Parameters.....	14
3.3 Trajectory Computing Using Optimal Control.....	16
3.4 Trajectory Computing Using 1 Polynomial.....	19
3.5 Trajectory Computing with 3 Polynomials Using Nonlinear	25
Optimization.....	23
3.5.1 Trajectory Computation.....	25
3.5.2 Parametrization of the Solution.....	31
3.6 Approximation of the Optimal Control Trajectory.....	35
3.7 Piecewise of Approximation of the Optimal Control Trajectory.....	37

3.8	Comparison	42
4.	VEHICLE IMPLEMENTATION IN THE FORM OF AN S-FUNCTION.....	45
4.1.	General S-Function Description.....	45
4.1.1.	MdlInitializeSizes.....	46
4.1.2.	MdlInitializeSamplesTimes.....	46
4.1.3	MdlOutputs.....	47
4.1.4	MdlTerminate.....	47
4.1.5	MdlDerivatives.....	47
4.1.6	MdlStart.....	48
4.1.7	MdlUpdate.....	48
4.1.8	MdlInitializeConditions.....	48
4.2	Requirements for Vehicle Model.....	48
4.2.1	Input Implementation.....	49
4.2.2	Output Implementation.....	51
4.2.3	Update Function Implementation.....	51
4.2.4	Derivative Implementation	52
4.2.5	Example Simulations.....	52
5.	CONCLUSION AND FUTURE WORK.....	56
REFERENCES.....		R1
CURRICULUM VITAE		A1

LIST OF FIGURES

FIGURES

Figure 1	Vehicle string with CACC.....	4
Figure 2	Vehicle string with 4 vehicles.....	6
Figure 3	Vehicle string with 4 vehicles and CACC design that realize strict string stability, a) Position b) Velocity c) Acceleration.....	7
Figure 4	Zoom of Figure 2.3, Velocity and Acceleration plots.....	8
Figure 5	Vehicle string with 4 vehicles and CACC design that realize not strict string stability, a) Position b) Velocity c) Acceleration.....	9
Figure 6	Zoom of Figure 2.5, Velocity and Acceleration plots.....	9
Figure 7	Block diagram of CACC.....	10
Figure 8	Opening gap or adding vehicle in the string.....	11
Figure 9	Closing gap or leaving vehicle in the string.....	12
Figure 10	Block diagram for extended CACC by input signal and desired distance Δq for opening and closing a gap.....	12
Figure 11	Plots of opening a gap by the dyed vehicle in a string that has four vehicles, a) Position b) Velocity c) Acceleration.....	13
Figure 12	Plots of closing a gap by the dyed vehicle in a string that has three vehicles, a) Position b) Velocity c) Acceleration.....	14
Figure 13	An example of polynomial trajectories for desired distance Δq and input signal.....	16
Figure 14	Optimal Control's results with $\Delta q_i=40$, $V_i=5$ a) Desired distance b) Input signal c) Velocity d) Acceleration.....	18
Figure 15	Optimal Control's results with $\Delta q_i=60$, $V_i=-5$ a) Desired distance b) Input signal c) Velocity d) Acceleration.....	19
Figure 16	Optimal Control's results with $\Delta q_i=40$, $V_i=-8$ a) Desired distance b) Input signal c) Velocity d) Acceleration.....	20
Figure 17	Optimal Control's results with $\Delta q_i=60$, $V_i=3$ a) Desired distance b) Input signal c) Velocity d) Acceleration.....	21
Figure 18	1 polynomial's result with $\Delta q_i=40$, $V_i=-5$ a) Desired distance b) Input signal c) Velocity d) Acceleration.....	24
Figure 19	1 polynomial's result with $\Delta q_i=60$, $V_i=-5$ a) Desired distance	24

b) Input signal c) Velocity d) Acceleration.....	
Figure 20 1 polynomial's result with $\Delta q_i=40$, $V_i=-8$ a) Desired distance	25
b) Input signal c) Velocity d) Acceleration.....	
Figure 21 1 polynomial's result with $\Delta q_i=60$, $V_i=3$ a) Desired distance	26
b) Input signal c) Velocity d) Acceleration.....	
Figure 22 Optimal Control's result with 3 polynomial at $\Delta q_i=40$, $V_i=-5$	30
a) Desired distance b) Input signal c) Velocity d) Acceleration.....	
Figure 23 Optimal Control's result with 3 polynomial at $\Delta q_i=60$, $V_i=-5$	30
a) Desired distance b) Input signal c) Velocity d) Acceleration.....	
Figure 24 Optimal Control's result with 3 polynomial at $\Delta q_i=40$, $V_i=-8$	31
a) Desired distance b) Input signal c) Velocity d) Acceleration.....	
Figure 25 Optimal Control's result with 3 polynomial at $\Delta q_i=60$, $V_i=3$	32
a) Desired distance b) Input signal c) Velocity d) Acceleration.....	
Figure 26 $x(4)=v_3$ and $x(8)=w_2$ for the specified values of Δq and V_i	32
Figure 27 Approximated trajectory for $\Delta q=24.3\text{m}$, and $V_i=-2.7\text{m/s}$	33
Figure 28 Approximated trajectory for $\Delta q=55.4\text{m}$, and $V_i=-8.3\text{m/s}$	34
Figure 29 Approximated trajectory for $\Delta q=24.3\text{m}$, and $V_i=-2.7\text{m/s}$	37
Figure 30 Approximated trajectory for $\Delta q=24.3\text{m}$, and $V_i=-8.3\text{m/s}$	38
Figure 31 Piecewise Approximation of the Optimal Control trajectory figures	39
a) Desired distance b) Velocity c) Acceleration.....	
Figure 32 Piecewise Approximation of the Optimal Control trajectory for	42
$\Delta q=24.3\text{m}$, and $V_i=-2.7\text{m/s}$	
Figure 33 Piecewise Approximation of the Optimal Control trajectory for	43
$\Delta q=59\text{m}$, and $V_i=-10\text{m/s}$	
Figure 34 Comparison of different feedforward signals for	44
$\Delta q=40\text{m}$, and $V_i=-4\text{m/s}$	
Figure 35 General model with vehicles and RSU.....	48
Figure 36 Vehicle S-Function Block.....	53
Figure 37 Position of vehicles response when applied to second vehicle for	54
opening a gap.....	
Figure 38 Position of vehicles response when applied to third vehicle for	54
opening a gap.....	
Figure 39 Position of vehicles response when applied to third vehicle for	55
closing a gap.....	

LIST OF ABBREVIATIONS

ITS	Intelligent Transportation Systems
CACC	Cooperative Adaptive Cruise Control
ACC	Adaptive Cruise Control
V2V	Vehicle to Vehicle
V_i	Velocity of Vehicle i
d_i	Actual Inter-Vehicle Distance of Vehicle i
$d_{r,i}$	Desired Inter-Vehicle Distance of Vehicle i
r_i	Standstill Distance of Vehicle i
h	Time Headway
e_i	Spacing Error of Vehicle i
q_i	Rear Bumper Position of Vehicle i
L_i	Length of Vehicle i
G	Plant of Vehicle
θ	Communication Delay
ϕ	Possible Plant Delay
Γ_i	Complementary Sensitivity of Vehicle i
τ	Time Constant
$H(s)$	Spacing Policy
K_{fb}	Feedback Controller
K_{ff}	Feedforward Controller
K_{fbo}	Main feedback Controller
K_{ffo}	Main feedforward Controller

CHAPTER 1

INTRODUCTION

Intelligent Transport Systems (ITS) are developed with the aim of improving the traffic infrastructure, driving safety and efficiency [1, 19, 20, 5] and addressing problems such as traffic congestion, fuel consumption and air pollution [3, 4].

An important application of ITS is platooning, where vehicles follow each other in the form of vehicle strings with a small inter-vehicle spacing. The most recent technology for the realization of platooning is cooperative adaptive cruise control (CACC) [13, 6]. CACC is an extension of adaptive cruise control (ACC). Measuring the inter-vehicle distance and relative velocity between vehicles by RADAR or LIDAR and receiving data from the predecessor vehicles via vehicle to vehicle (V2V) communication, CACC allows controlling the throttle and brake to maintain a certain desired distance.

CACC is applicable for vehicle following in vehicle strings, where vehicles are already arranged at a desired distance. In contrast, practical driving situations require lane change maneuvers, where vehicles join or leave a vehicle string. In order to maintain driving safety, such maneuvers necessitate the opening (for a joining vehicle) or closing (after a leaving vehicle) of gaps in vehicle strings. A basic version of this problem has been addressed for the first time in [10, 11]. The idea of the method is to define a trajectory for the distance signal in the form of a polynomial and then use plant inversion to determine the feedforward input signal. However, several disadvantages of that method such as a long maneuver duration and the violation of velocity and acceleration constraints can be identified.

The main aim of this thesis is the development of methods for computing trajectories for gap opening and closing maneuvers. Hereby, the focus is on feedforward input signals for fast maneuvers without violating the defined constraints. The first method formulates an optimal control problem, whose solution provides the feedforward signal with the shortest possible maneuver while meeting all constraints. On the downside, the computation times for finding the optimal control solution is not suitable for a real-time implementation. A further method with three piecewise polynomials and velocity/acceleration constraints is formulated in the form of a nonlinear program.

Its solution can be obtained faster than the optimal control computation but is still not suitable for a real-time implementation. In addition, a longer maneuver duration has to be accepted for this method. To reduce the computation time, an approximation/interpolation method has been developed. It computes suitable trajectories approximating the nonlinear programming solution for a grid of parameter values offline. Then, trajectories for any parameter combination that is not on the grid can be obtained by linear interpolation. It is shown that this method leads to very good approximations. In order to further reduce the maneuver duration, the same interpolation method is applied to polynomial approximations of the optimal control solution itself. As a result, fast trajectories are obtained. Slight violations of the acceleration constraints are possible if the optimal control solution is approximated by a single polynomial. On the contrary, an approximation by five concatenated polynomials proves very suitable because it directly captures the shape of open/close gap trajectories. The quality of the approximations is demonstrated by various simulation examples. In addition, the thesis implements a simulation model of the vehicle with CACC and the feedforward signal computation in the form of an S-function.

The thesis is organized as follows. Chapter 2 provides background information. In Chapter 3, the developed methods for opening and closing gaps for safe lane changes are presented. Chapter 4 implements a simulation model of the vehicle with CACC and feedforward signal generation as an S-function and Chapter 5 gives conclusions.

CHAPTER 2

BACKGROUND

This chapter gives the necessary background on vehicle following that is realized using the technique of cooperative adaptive cruise control (CACC). Section 2.1 explains the general concept of vehicle following and CACC, Section 2.2 explains the concept of string stability and gives several simulation examples. The controller design for CACC used in this report is described in Section 2.3.

2.1 CACC and Vehicle Following

CACC is developed for overcoming the shortcomings of adaptive cruise control (ACC) [7, 12, 13, 8]. Measuring the inter-vehicle distance, ACC automatically controls the throttle position and brake to maintain a certain desired distance to the predecessor vehicle. CACC extends ACC by also using data that is transmitted from other vehicles by vehicle-to-vehicle communication. That is, CACC is an emerging technology that supports vehicle following at small inter-vehicle distances as shown in Fig. 1.

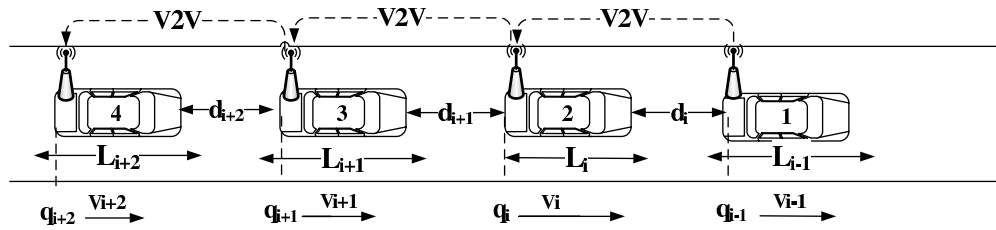


Figure 1: Vehicle string with CACC.

In this figure, the following parameters/variables are used:

- L_i : Length of vehicle i ,
- v_i : Velocity of vehicle i ,
- q_i : Rear bumper position of vehicle i ,
- d_i : Distance between vehicle $i - 1$ and vehicle i .

There are two ways to obtain sensor information for each vehicle:

- vehicle distance from sensor measurements such as RADAR OR LIDAR,
- sensor information from other vehicles (such as velocity or acceleration) via vehicle-to-vehicle (V2V) communication.

In order to formulate an analytical model of CACC according to Fig. 1, the actual distance d_i between vehicle i and vehicle $i - 1$ is evaluated as

$$d_i(t) = q_{i-1}(t) - q_i(t) - L_i. \quad (2.1)$$

In addition, the desired distance $d_{i,r}$ (also denoted as spacing policy) needs to be considered. In this thesis, we use the constant headway time policy which is mostly used in the recent literature. It is given in the following equation.

$$d_{i,r} = r_i + h v_i. \quad (2.2)$$

$d_{i,r}$ describes the desired distance between vehicle i and vehicle $i - 1$. r_i is the distance at standstill and h_i shows the headway time. When the velocity is zero, the desired distance is r_i and $d_{i,r}$ increases proportionally with v_i . Subtracting (2.2) from (2.1), the spacing error $e_i(t)$ is computed as

$$e_i(t) = d_i(t) - d_{i,r}(t) = (q_{i-1}(t) - q_i(t) - L_i) - (r_i + h v_i(t)). \quad (2.3)$$

2.2 String Stability

For safe and comfortable driving, it is required to ensure that errors and signal levels decrease along a vehicle string as in Fig. 1. In other words, any changes of vehicle i should not be amplified by the follower vehicle $i + 1$ and other followers. In the literature, the aforementioned condition is described as strict string stability. Consider a linear system that is modeled by the transfer function

$$\Gamma_i(s) = \frac{U_i(s)}{U_{i-1}(s)}. \quad (2.4)$$

Strict string stability can be ensured for all vehicles i between the control inputs $U_i(s)$ and $U_{i-1}(s)$ if

$$\|\Gamma_i(s)\|_\infty \leq 1. \quad (2.5)$$

Here, $U_i(s)$ represents the Laplace transform of the signal $U_i(t)$ and also $\|\bullet\|_\infty$ represents the H_∞ -norm.

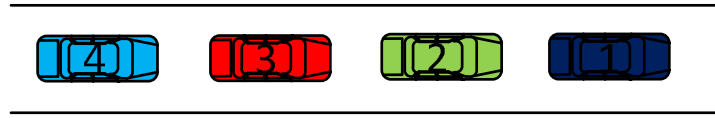


Figure 2: Vehicle string with 4 vehicles.

To give a better understanding of string stability, we prepared an example scenario with 4 vehicles. The leader vehicle (vehicle 1 in dark-blue color) accelerates from 0 to 2 m/sec² within about 5 sec and then decelerates -2 m/sec² within approximately 5 sec as is shown in the acceleration plot (right-hand plot) of Fig. 3. The follower vehicles (green, red and blue colors) show a reduced acceleration and deceleration. The velocity plot (center plot) shows an analogous result for the vehicle velocity. In addition, it can be seen that the desired safe distance between the vehicles is maintained by looking at the position plot (left-hand plot).

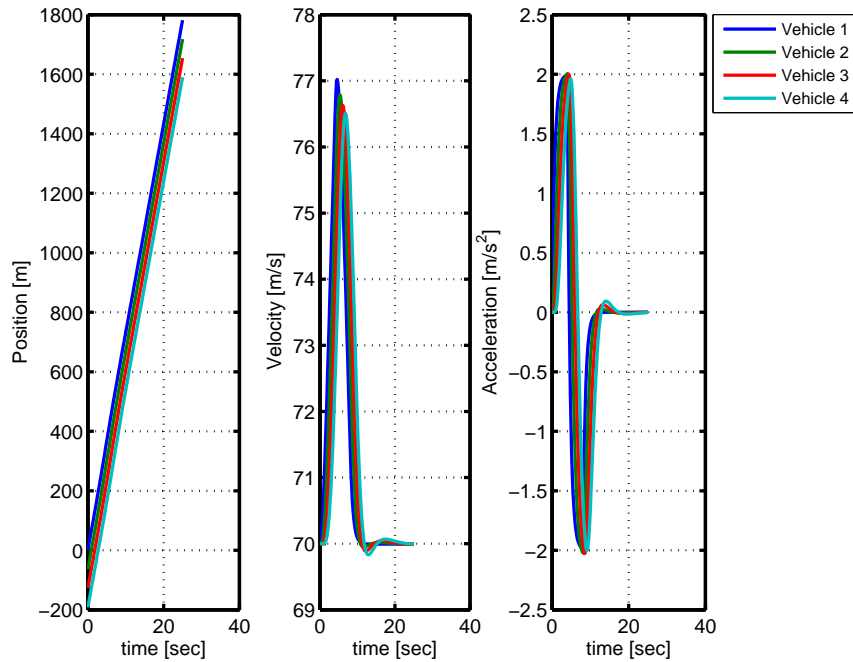


Figure 3: Vehicle string with 4 vehicles and CACC design that realize strict string stability, a) Position b) Velocity c) Acceleration.

Fig. 4 zooms in on the relevant part of the plots in Fig. 3 in order to clarify better the signal attenuation of a strictly string stable vehicle string.

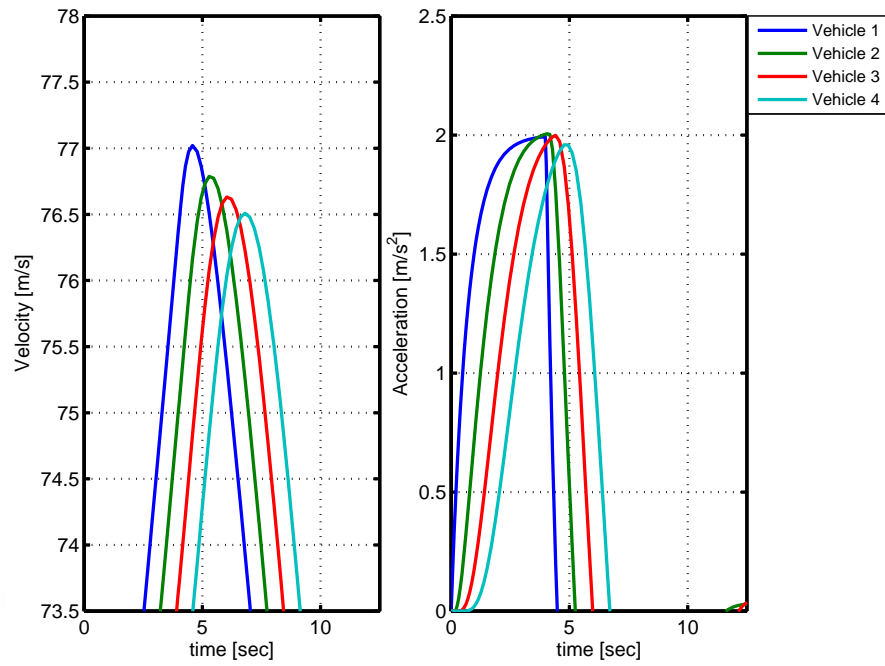


Figure 4: Zoom of Figure 2.3, Velocity and Acceleration plots.

In contrast, the same vehicle string in Fig. 2 is simulated for the case where strict string stability is violated in Fig. 5. Here, the acceleration plot (right-hand plot) and velocity plot (center plot) suggest and increase in the signal amplitude. Again, Fig. 6 with the additional zoom clarifies the violation of strict string stability.

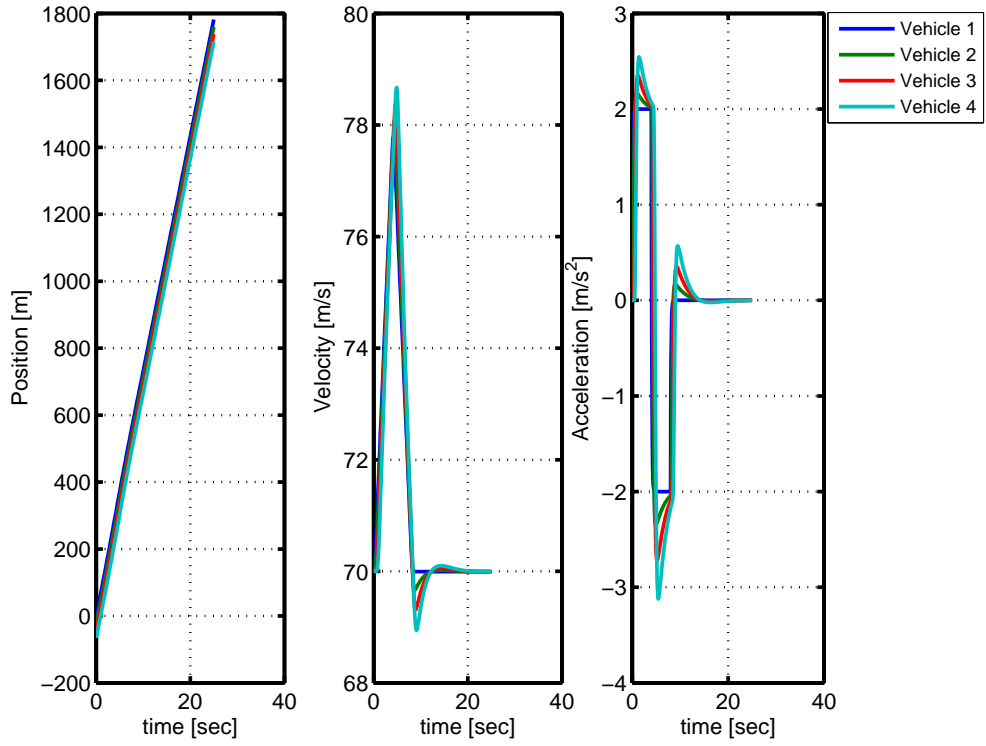


Figure 5: Vehicle string with 4 vehicles and CACC design that realize not strict string stability, a) Position b) Velocity c) Acceleration.

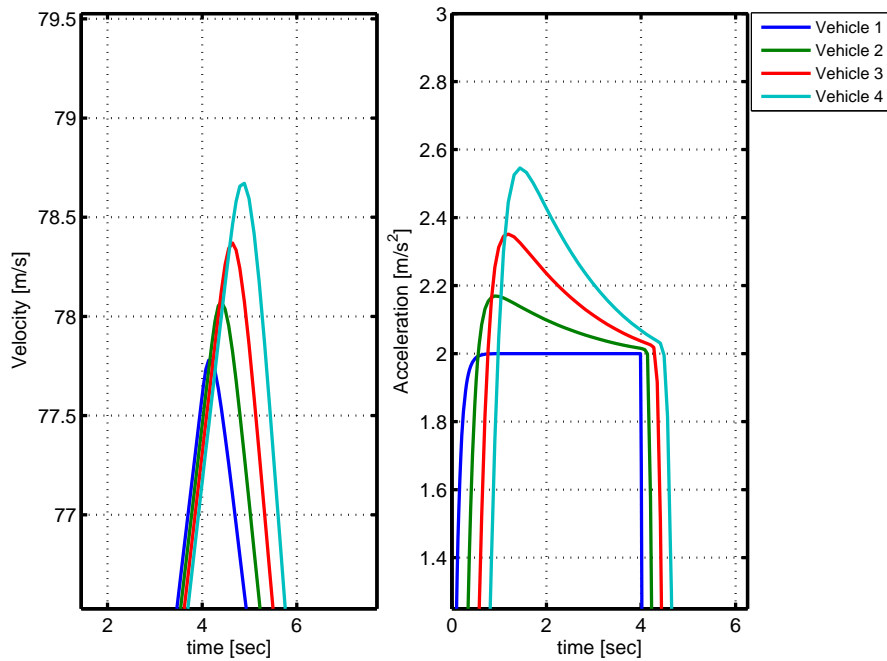


Figure 6: Zoom of Figure 2.5, Velocity and Acceleration plots.

2.3 Controller Design for CACC and Simulations

In the literature, systems are mostly designed for homogeneous strings, that is, all vehicles have the same dynamic features. In the most recent studies the spacing policy in (2.2) is used and the vehicle plant is modeled by the transfer function

$$G_i(s) = \frac{e^{-\phi_i s}}{(1 + s \tau_i) s^2} = \frac{Q_i(s)}{U_i(s)}. \quad (2.6)$$

ϕ_i is a possible plant delay and τ_i is the time constant of the low level drive line dynamics. The first integration obtains the vehicle velocity from the acceleration and the second integration obtains the vehicle position from the velocity.

Then, CACC is realized in the following block diagram in Fig. 7. For vehicle $i - 1$,

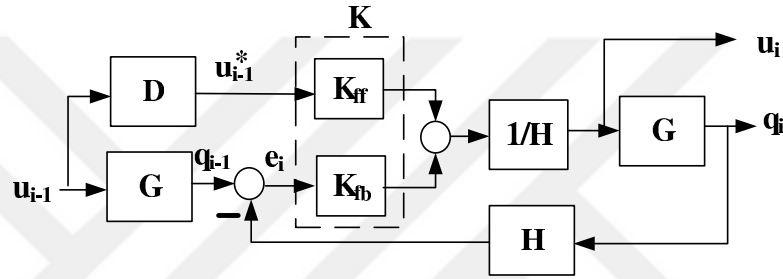


Figure 7: Block diagram of CACC.

the input signal u_{i-1} is transmitted to vehicle i via V2V communication and $D = e^{-\theta s}$ shows a possible communication delay. $H = 1 + hs$ is used to implement the spacing policy in (2.2) with headway h . K is the controller transfer matrix and it can be written as below.

$$K = \begin{bmatrix} K_{ff} & K_{fb} \end{bmatrix}. \quad (2.7)$$

- K_{ff} is a feedforward filter for data input transmitted by the preceding vehicle,
- K_{fb} is also a feedback control transfer function in order to control spacing error e_i based on measurement distance d_i between the current and desired distance.

For homogeneous vehicles, $\tau_i = \tau_{i-1}$ value is provided for every vehicle i and we can write $G(s) = G_i(s)$. Then, the transfer function $\Gamma_i(s)$ in (2.4) can be written as

$$\Gamma(s) = \frac{DK_{ff} + GK_{fb}}{H(1 + GK_{fb})}. \quad (2.8)$$

Therefore, according to (2.5), K has to be designed such that

$$\|\Gamma(s)\|_{\infty} \leq 1. \quad (2.9)$$

There are different designs in the literature that achieve this task. The work in this report is based on the controller design in [8].



CHAPTER 3

OPENING AND CLOSING GAPS FOR SAFE LANE CHANGES

This chapter describes different methods for opening and closing a gap for safe lane changes within a vehicle string. Section 3.1 shows the basic maneuvers and small examples for opening and closing a gap. Section 3.2 outlines the requirements and important parameters, Section 3.3 to 3.7 explain different types of feedforward control computations.

3.1 Basic Maneuvers

We first describe the basic scenario of a lane change and explain the requirement of opening/closing gaps. Consider the example in Fig. 8 with four vehicles, a target lane and a current lane. Vehicles 1, 2, 3 travel on the target lane and the colored vehicle travels on the current lane before the lane change in Fig. 8 (a). In order to join the colored vehicle to the string, a gap needs to be opened between vehicle 1 and 2 in Fig. 8 (b). After the lane change, all 4 vehicles travel on the target lane and follow each other using CACC in Fig. 8 (c).

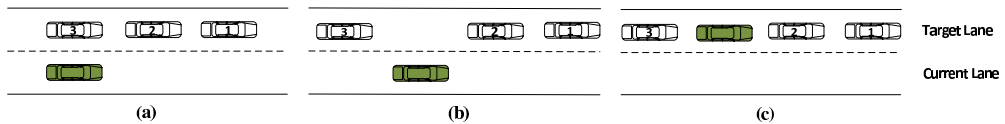


Figure 8: Opening gap or adding vehicle in the string.

A similar procedure is needed when a vehicle leaves a string. For example, when the colored vehicle leaves the vehicle string, a large gap is created between vehicle 1 and 2 in Fig. 9 (b). This gap needs to be closed to obtain the original desired distance according to (2.2) when using CACC in Fig. 9 (c). The idea of this work is to apply additional feedforward input signals for opening or closing a gap between vehicle i and its preceding vehicle $i - 1$. To this end, we use the control architecture in Figure 10.

The figure shows three consecutive vehicles $i - 1, i, i + 1$. Vehicle i implements the desired maneuver. First, it holds that the feedback control architecture is not changed compared to the architecture in Section 2.3. The controller $K_i = \begin{bmatrix} K_{ff,i} & K_{fb,i} \end{bmatrix}$ consists

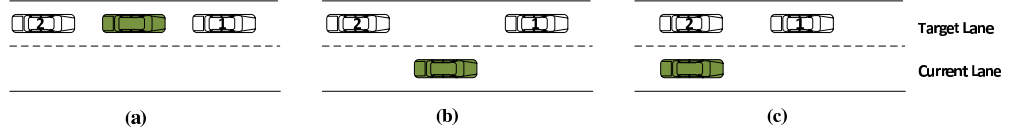


Figure 9: Closing gaps or leaving vehicle in the string.

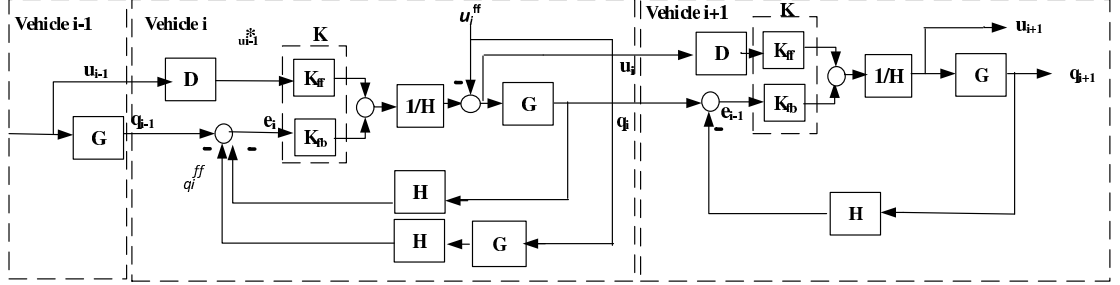


Figure 10: Block diagram for extended CACC by input signal u_i^{ff} and desired distance Δq_i for opening and closing a gap

of the feedforward controller $K_{ff,i}$ that receives signal data from the predecessor vehicle $i - 1$ and the feedback controller $K_{fb,i}$ controls the distance error e_i . In addition, the control architecture in Figure 10 has an exogenous input signal u_i^{ff} that can be used to perform arbitrary maneuvers of vehicle i . There is an additional path with the transfer function $H_i \cdot G_i$ between u_i^{ff} and e_i . This path ensures that the reference signal is adjusted when u_i^{ff} is applied. Without this path, the feedback controller $K_{fb,i}$ will consider u_i^{ff} as a disturbance and hence try to reject the signal instead of enabling the desired maneuver.

In this thesis, we use the state space model below to model vehicle i in the block diagram for the case of PD control and direct input signal feedforward. That is, $K_{fb,i} = K_p + K_d s$ and $K_{ff,i} = 1$ [15].

$$\dot{q}_i = v_i. \quad (3.1)$$

$$\dot{v}_i = a_i. \quad (3.2)$$

$$\dot{a}_i = \frac{-1}{\tau} \cdot a_i + \frac{1}{\tau} \cdot u_i. \quad (3.3)$$

$$\begin{aligned} \dot{u}_i = & \frac{-1}{h} \cdot u_i + \frac{1}{h} \cdot (K_p \cdot (q_{i-1} - q_{i,FF} - q_i - h \cdot v_i - L_i - r_i) + \\ & (K_d \cdot (v_{i-1} - v_{i,FF} - v_i) + (u_{i-1} + u_{i,FF} + h \cdot \dot{u}_{i,FF}))). \end{aligned} \quad (3.4)$$

The control architecture in Figure 10 has not been used in the literature before. In the existing literature, changes in the inter-vehicle distance of a vehicle i are usually realized as reference signal steps with a very bad effect on the following vehicles [9]. The proposed architecture has two additional advantages. First, the control method for

vehicle following is not changed compared to Section 2.3. That is, the controller design method in Section 2.3 for vehicle following is still suitable. Second, the application of u_i^{ff} only affects the vehicle plant G_i but not the feedback control loop. That is, the additional vehicle maneuvers can be designed independent of the control loop for vehicle following.

In order to show position, velocity and acceleration plots when a gap is opened, we prepared a scenario with four vehicles. Positions, velocity and acceleration of the vehicles are plotted in Fig. 11. We used 10 seconds for the time duration of opening a gap. The leader vehicle travels at a constant velocity of 20 m/s. After applying the feedforward input signal to the colored vehicle, a gap is opened between the times 8 sec and 18 sec. After the determined time, vehicle 3 follows the colored vehicle by means of CACC.

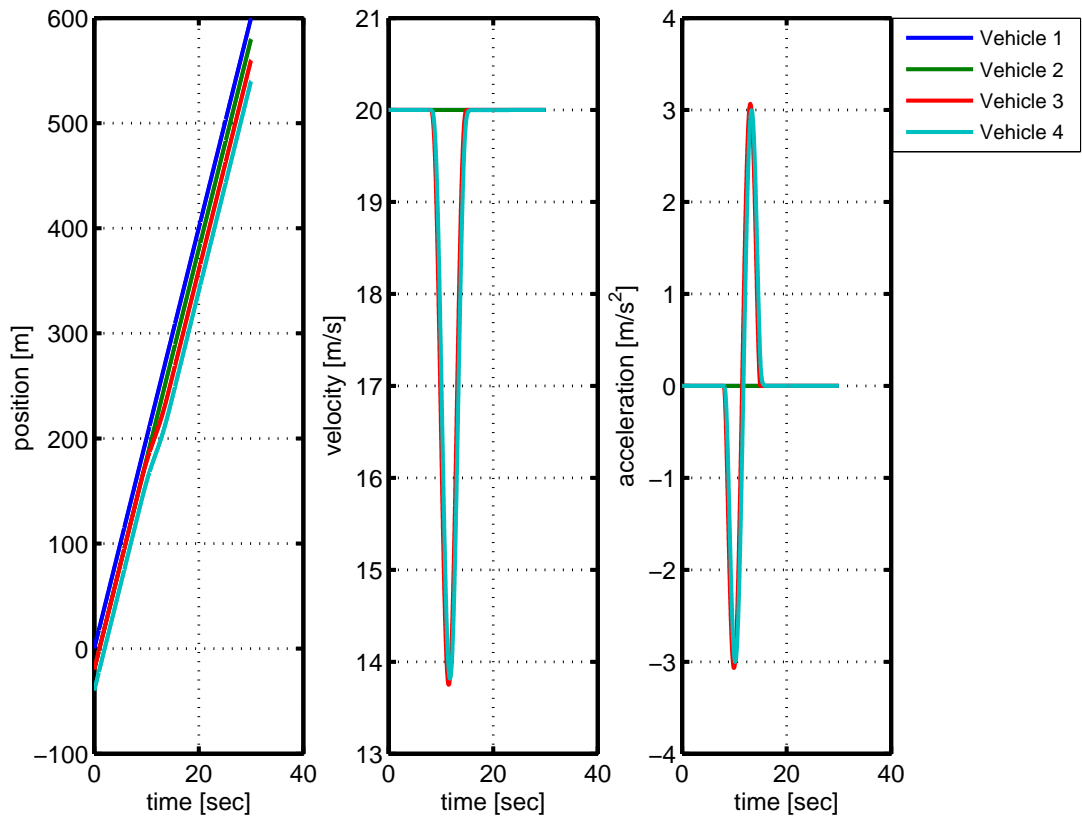


Figure 11: Plots of opening a gap by the dyed vehicle in a string that has four vehicles, a) Position b) Velocity c) Acceleration.

In order to show a close gap maneuver, we consider a scenario with 3 vehicles. Positions, velocity and acceleration graphs of the vehicles are plotted in Fig. 12. We also selected time duration of 10 sec for closing a gap. The leader vehicle moves at 20 m/sec

as before. The gap is closed between 8 and 18 sec.

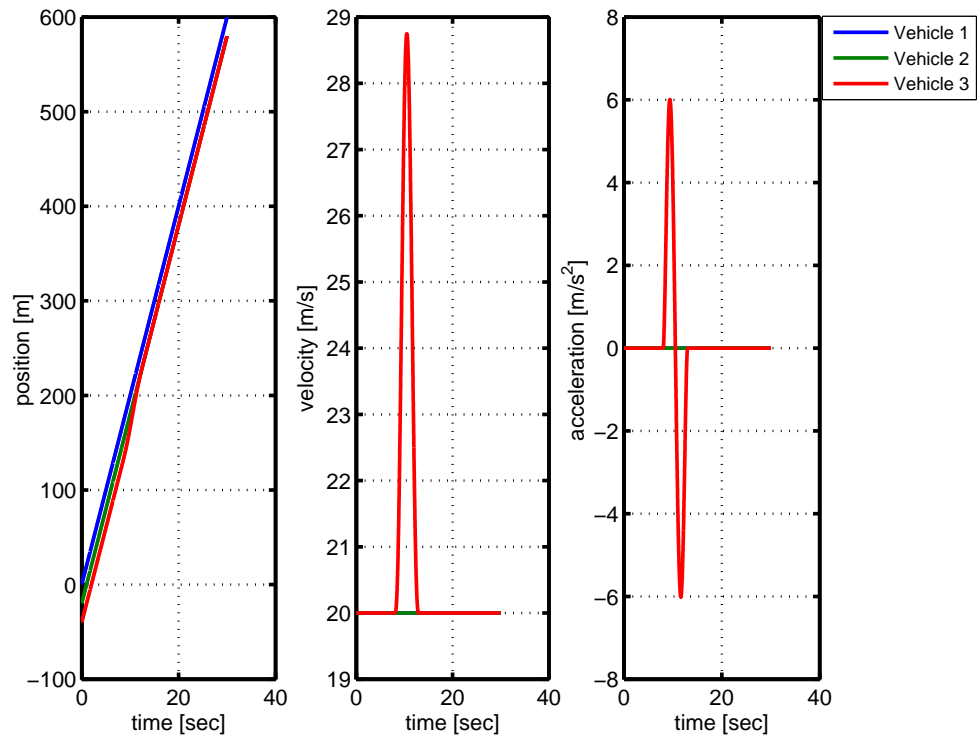


Figure 12: Plots of closing a gap by the dyed vehicle in a string that has three vehicles, a) Position b) Velocity c) Acceleration.

3.2 Requirements and Parameters

The maneuvers for opening/closing gaps of a vehicle i are maneuvers that are carried out in addition to following the predecessor vehicle $i - 1$. That is, they constitute a difference to the trajectory of vehicle $i - 1$. We write $\Delta q_i(t)$, $\Delta v_i(t)$, $\Delta a_i(t)$ for the position difference, velocity difference, acceleration difference, respectively. These difference signals depend on several parameters. First, we consider what happens after completion of a maneuver, whereby we assume that the maneuver starts at time $t = 0$ and its duration is denoted as T . At time T , it is desired that vehicle i follows vehicle $i - 1$ at a certain distance difference Δq at the same velocity and at the same acceleration. That is we want that

$$\Delta q_i(T) = \Delta q \quad (3.5)$$

$$\Delta v_i(T) = \Delta \dot{q}_i(t) = 0 \quad (3.6)$$

$$\Delta a_i(T) = \Delta \ddot{q}_i(t) = 0 \quad (3.7)$$

Looking at time $t = 0$, the position difference starts from zero, whereas there might be a velocity difference to the predecessor vehicle $i - 1$ denoted as V_i . Finally, we assume that maneuvers are carried out as long vehicles do not accelerate. In summary, this leads to the following constraints.

$$\Delta q_i(0) = 0 \quad (3.8)$$

$$\Delta v_i(0) = \Delta \dot{q}_i(0) = V_i \quad (3.9)$$

$$\Delta a_i(0) = \Delta \ddot{q}_i(0) = 0. \quad (3.10)$$

That is, the free parameters for designing additional maneuvers are the desired gap distance Δq and the initial velocity difference V_i . The parameters of an example maneuver are discussed in Fig. 13. Here, a trajectory is determined for $\Delta q = 64$ and $V_i = 0$ and $T = 10$ sec. The figure both shows the position trajectory and the related input signal u_i^{ff} .

The main subject of this thesis is the computation of suitable trajectories for realizing open/close gap maneuvers. Such trajectories should fulfill several properties as follows.

- The distance between vehicle i and vehicle $i - 1$ is changed to a pre-defined value Δq as identified above,
- A potential initial velocity difference V_i of the vehicles is respected and the final

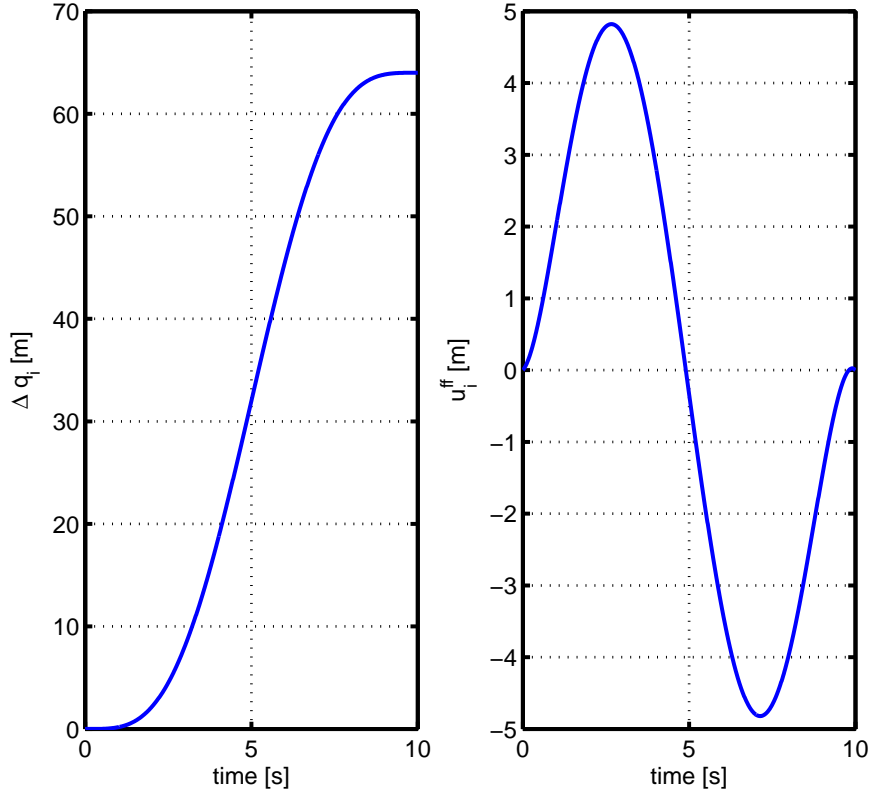


Figure 13: An example of polynomial trajectories for Δq_i and u_i^{ff} .

velocity difference of the vehicles is zero,

- Accelerations/decelerations of vehicle i should be tolerable for human passengers. That is, the acceleration should remain between a lower bound a_{\min} and an upper bound a_{\max} . Suitable bounds identified from the literature are about $a_{\max} = 2 \text{ m/s}^2$ for the maximum acceleration and $a_{\min} = -2 \text{ m/s}^2$ for the minimum acceleration [16, 17, 18],
- The velocity difference of vehicle i should remain between a minimum velocity v_{\min} and should not exceed a maximum velocity v_{\max} ,
- The maneuver should be completed in the shortest possible time.

In the following sections, we develop five methods for the trajectory computation that attempt to fulfill the above items.

3.3 Trajectory Computation using Optimal Control

Firstly, we formulate an optimal control problem that addresses the items in Section 3.2. We want to determine

$$\min \int_0^T 1 dt \quad (3.11)$$

subject to the dynamic constraints

$$\dot{q}_i = v_i \quad (3.12)$$

$$\dot{v}_i = a_i \quad (3.13)$$

$$\dot{a}_i = \frac{-1}{\tau} \cdot a_i + \frac{1}{\tau} \cdot u_i \quad (3.14)$$

initial conditions

$$q_i(0) = 0, \quad v_i(0) = V_i, \quad a_i(0) = 0 \quad (3.15)$$

terminal conditions

$$q_i(T) = \Delta q, \quad v_i(T) = 0, \quad a_i(T) = 0 \quad (3.16)$$

additional constraints

$$v_{\min} \leq v_i(t) \leq v_{\max} \quad (3.17)$$

$$a_{\min} \leq a_i(t) \leq a_{\max} \quad (3.18)$$

The formulated problem is an optimal control problem with linear dynamics and state constraints. It tries to minimize the completion time of the maneuver, while meeting the specified initial and final conditions as well as the velocity and acceleration constraints. We use the "PROPT solver" that is part of the Tomlab library [14] for the solution of this optimal control problem.

In order to demonstrate the computation of optimal trajectories for open gap maneuvers, we select different parameter values for the final position difference Δq and the initial velocity V_i . Recall, the values of the initial position $q_i(0) = 0$, the initial acceleration $a_i(0) = 0$, the final velocity $v_i(T) = 0$ and the final acceleration $a_i(T) = 0$. We further use $v_{\min} = -10$ m/s, $v_{\max} = 10$ m/s, $a_{\min} = -2$ m/s² and $a_{\max} = 2$ m/s².

In the first experiment, we choose $\Delta q = 40$ and $V_i = -5$. The result of the optimal

control computation is shown in Fig. 14. Second, Fig. 15 gives the result for $\Delta q = 60$

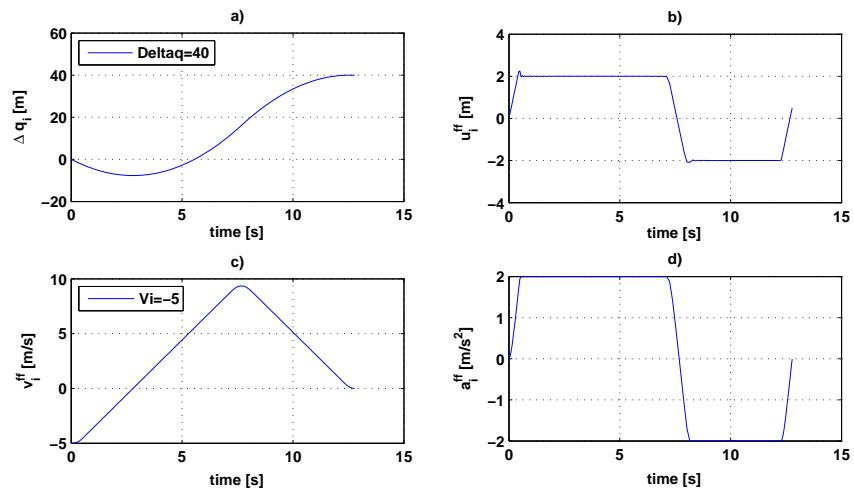


Figure 14: Optimal Control's results with $\Delta q_i = 40$, $V_i = -5$ a) Desired distance b) Input signal c) Velocity d) Acceleration.

and $V_i = -5$.

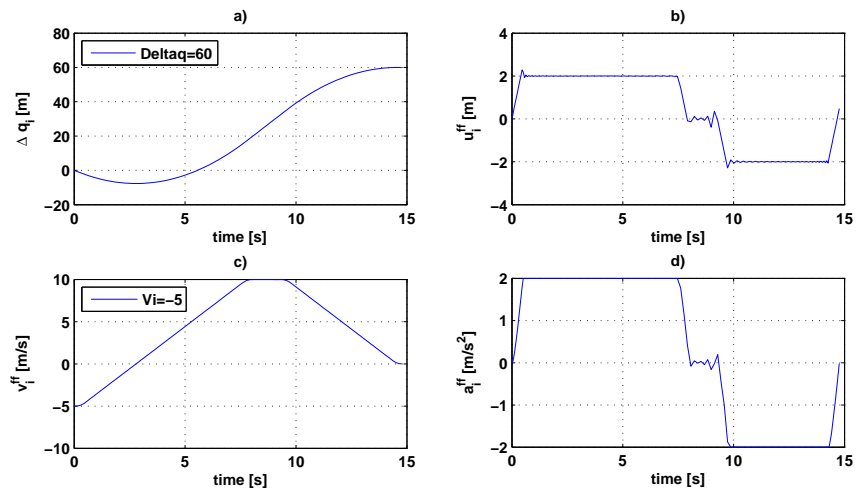


Figure 15: Optimal Control's results with $\Delta q_i = 60$, $V_i = -5$ a) Desired distance b) Input signal c) Velocity d) Acceleration.

Next, Fig. 16 uses the parameters $\Delta q = 40$ and $V_i = -8$. Finally, we use $\Delta q = 60$ and $V_i = 3$ in Fig. 17.

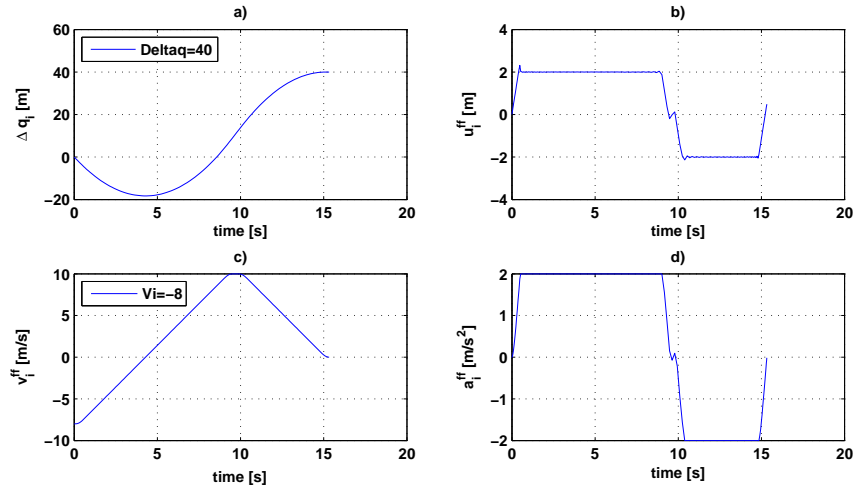


Figure 16: Optimal Control's results with $\Delta q_i = 40$, $V_i = -8$ a) Desired distance b) Input signal c) Velocity d) Acceleration.

We next describe several observations from the optimal control results.

First, looking at Fig. 14, 15 and 16, it holds that the position difference first becomes negative before reaching the desired positive value. This effect is caused by the fact that the initial velocity is negative. That is the position difference first decreases until the velocity reaches positive values. This is different in Fig. 17, where the initial velocity is positive.

Second, we compare Fig. 14 and 15. Here, the initial velocities are the same but the desired distance is different. It can be seen that increasing the desired distance leads to a longer duration of the maneuver. In this case, it can also be seen that the velocity limit is reached in Fig. 15.

Third, we compare Fig. 14 and 16 with the same desired distance but a different initial velocity. Since the initial velocity is smaller in Fig. 16 the maneuver duration is longer.

Finally we compare Fig. 15 and 17. Here, the increased initial velocity in Fig. 17 leads to a shorter maneuver.

In summary, the optimal control formulation is suitable for determining additional input signals for open gap maneuvers. Hereby, all constraints are met since they are included in the formulation. We note that the same formulation can be used for closing gaps. In this case, it is only required to choose Δq negative. Unfortunately, there is one important shortcoming of this approach. The execution time for evaluating the optimal control problem is too large for a real-time implementation, which would be required when using the additional input signals in a practical application. For example, the execution time for the experiment with $\Delta q = 40$ m and $V_i = -5$ m/s is about 11 sec on a

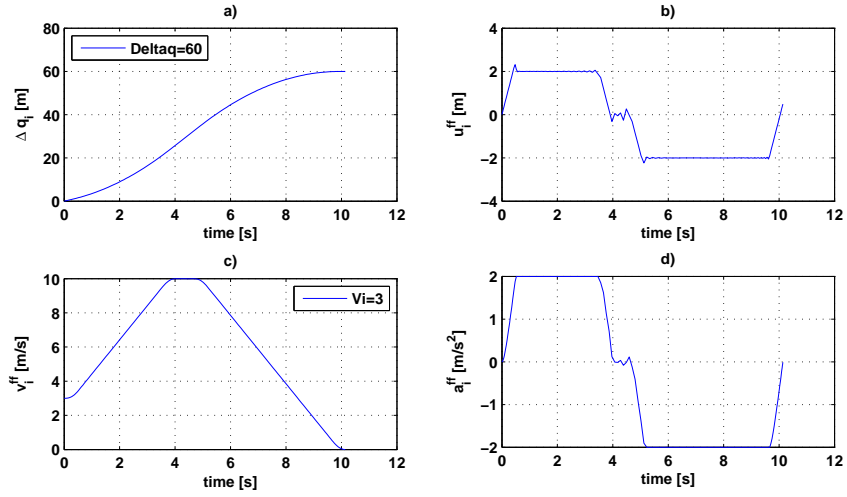


Figure 17: Optimal Control's results with $\Delta q_i = 60$, $V_i = 3$ a) Desired distance b) Input signal c) Velocity d) Acceleration.

laptop computer with the features below. Because of this reason, the following sections provide different methods for computing/representing input signal trajectories that can be evaluated in shorter time.

- Processor: Intel(R) Core(TM) 2 Duo CPU E750 @ 2.93GHz,
- Installed memory (RAM): 8.00 GB,
- System type: 64 bit Operating System,

3.4 Trajectory Computation using 1 Polynomial

A method for computing vehicle trajectories was proposed in the Master thesis [10]. The advantages/disadvantages of this method are also examined in this thesis. The idea of the method is to define a trajectory for the distance signal Δq in the form of a polynomial and then use plant inversion to determine the feedforward input u_i^{ff} .

The plant model is

$$G(s) = \frac{\Delta Q(s)}{U(s)} = \frac{1}{(1 + \tau s)s^2} \quad (3.19)$$

and the desired distance signal is

$$\Delta q(t) = \begin{cases} 0 & \text{for } t < 0 \\ f(t) & \text{for } 0 \leq t \leq T \\ \Delta q & \text{otherwise.} \end{cases}$$

Hereby, the polynomial

$$f(t) = v_0 + v_1 t + v_2 t^2 + \dots + v_l t^l \quad (3.20)$$

models the transition from the initial distance 0 to the final distance Δq . Respecting the additional conditions on the initial and final velocity and acceleration, a polynomial with degree $l = 7$ has to be chosen. Then, the polynomial coefficients are obtained from the conditions

$$f(0) = 0 \text{ and } f(T) = \Delta q. \quad (3.21)$$

$$\dot{f}(0) = V_i \text{ and } \dot{f}(T) = 0 \quad (3.22)$$

$$f^i(0) = 0 \text{ and } f^i(T) = 0 \text{ for } i = 2, 3. \quad (3.23)$$

That is, using

$$y(t) = v_0 + v_1 t + v_2 t^2 + v_3 t^3 + v_4 t^4 + v_5 t^5 + v_6 t^6 + v_7 t^7. \quad (3.24)$$

$$\frac{dy(t)}{dt} = v_1 + 2v_2 t + 3v_3 t^2 + 4v_4 t^3 + 5v_5 t^4 + 6v_6 t^5 + 7v_7 t^6. \quad (3.25)$$

$$\frac{d^2 y(t)}{dt^2} = 2v_2 + 6v_3 t + 12v_4 t^2 + 20v_5 t^3 + 30v_6 t^4 + 42v_7 t^5. \quad (3.26)$$

$$\frac{d^3 y(t)}{dt^3} = 6v_3 + 24v_4 t + 60v_5 t^2 + 120v_6 t^3 + 210v_7 t^4. \quad (3.27)$$

the following equations for the polynomial coefficients are obtained.

$$y(0) = v_0 = y_0, \quad (3.28)$$

$$y(T) = v_0 + v_1 T + v_2 T^2 + v_3 T^3 + v_4 T^4 + v_5 T^5 + v_6 T^6 + v_7 T^7 = y_f \quad (3.29)$$

$$\dot{y}(0) = v_1 = 0, \quad (3.30)$$

$$\dot{y}(T) = v_1 + 2v_2 T + 3v_3 T^2 + 4v_4 T^3 + 5v_5 T^4 + 6v_6 T^5 + 7v_7 T^6 = 0 \quad (3.31)$$

$$\frac{d^2 y(0)}{dt^2} = v_2 = 0 \quad (3.32)$$

$$\frac{d^2 y(T)}{dt^2} = 2v_2 + 6v_3 T + 12v_4 T^2 + 20v_5 T^3 + 30v_6 T^4 + 42v_7 T^5 = 0 \quad (3.33)$$

$$\frac{d^3 y(0)}{dt^3} = v_3 = 0, \quad (3.34)$$

$$\frac{d^3 y(T)}{dt^3} = 6v_3 + 24v_4 T + 60v_5 T^2 + 120v_6 T^3 + 210v_7 T^4 = 0 \quad (3.35)$$

Using the vector of unknown parameters

$$v = [v_0 \ v_1 \ v_2 \ v_3 \ v_4 \ v_5 \ v_6 \ v_7]^T \quad (3.36)$$

the polynomial coefficients follow from the solution of the linear equation $Av = b$ with

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & T & T^2 & T^3 & T^4 & T^5 & T^6 & T^7 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2T & 3T^2 & 4T^3 & 5T^4 & 6T^5 & 7T^6 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 6T & 12T^2 & 20T^3 & 30T^4 & 42T^5 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 6 & 24T & 60T^2 & 120T^3 & 210T^4 \end{bmatrix} \quad (3.37)$$

and the vector b

$$b = \begin{bmatrix} 0 \\ \Delta q \\ V_i \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.38)$$

That is,

$$v = A^{-1} \cdot b. \quad (3.39)$$

Computing a solution with the symbolic toolbox of Matlab, the polynomial coefficients

are determined as

$$\begin{aligned}
v_0 &= 0 \\
v_1 &= V_i \\
v_2 &= 0 \\
v_3 &= 0 \\
v_4 &= \frac{35\Delta q}{T^4} - \frac{V_i(-T^2 + 15T + 120)}{6T^3} \\
v_5 &= \frac{V_i(-T^2 + 14T + 90)}{2T^4} - \frac{84\Delta q}{T^5} \\
v_6 &= \frac{70\Delta q}{T^6} - \frac{V_i(-T^2 + 13T + 72)}{2T^5} \\
v_7 &= \frac{V_i(-T^2 + 12T + 60)}{6T^6} - \frac{20\Delta q}{T^7}
\end{aligned}$$

That is, the polynomial $f(t)$ can be evaluated analytically for all combinations of T , Δq and V_i . In addition, it is possible to determine the input signal u_i^{ff} by plant inversion. It holds that

$$U(s) = (\tau s^3 + s^2) \Delta Q(s) \Rightarrow u(t) = \tau \cdot \frac{d^3 \Delta q(t)}{dt^3} + \frac{d^2 \Delta q(t)}{dt^2}. \quad (3.40)$$

Hence,

$$u(t) = \begin{cases} 0 & \text{for } t < 0 \\ \tau \cdot \frac{d^3 f(t)}{dt^3} + \frac{d^2 f(t)}{dt^2} & \text{for } 0 \leq t \leq T \\ 0 & \text{otherwise} \end{cases} \quad (3.41)$$

Accordingly, $u(t)$ can also be represented by a polynomial for all values of T , Δq and V_i .

In order to evaluate the generation of the additional input signal as described in this section, we use the same combinations of Δq and V_i as in Section 3.3. The resulting trajectories are shown in Fig. 18 to 21. Hereby, $T = 15$ sec was chosen for all examples as the maximum time duration observed in the experiments in Section 3.3.

In comparison, we make the following observations. In Fig. 18 and 20, it can be seen that increasing the distance Δq leads to high acceleration values. Similarly, looking at Fig. 18 and 20, high acceleration values are needed when decreasing the initial velocity, while keeping T and Δq the same. Conversely, smaller acceleration values are required if the initial velocity is increased as can be verified in Fig. 18 and 21.

The main advantage of using the trajectory generation method in this section is its easy

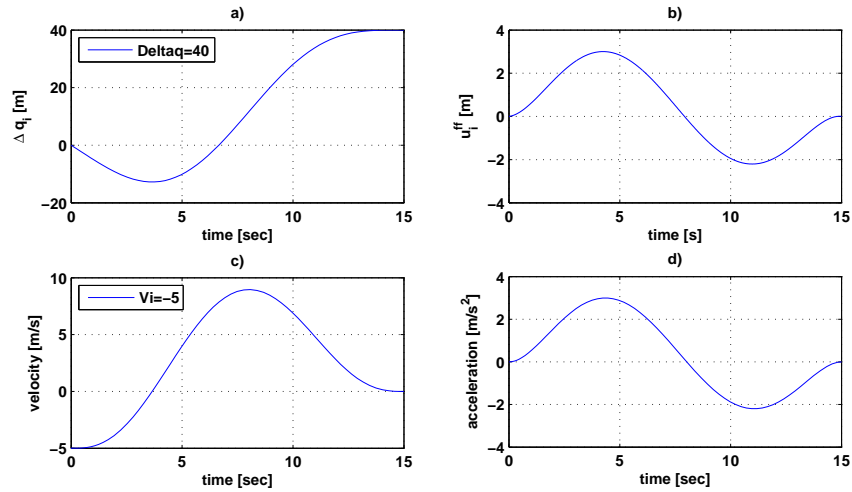


Figure 18: Results for single polynomial with $\Delta q_i = 40$, $V_i = -5$ a) Desired distance b) Input signal c) Velocity d) Acceleration.

computation. Since all polynomial coefficients are analytically available, they can be computed in very short time for all combinations of T , Δq and V_i . On the downside, the suggested polynomial does not approximate the optimal control solution well. As can be seen from all experiments, comparably large acceleration and velocity values are needed, violating the respective constraints. This is true even if larger values of T are selected compared to Section 3.3. The methods in the subsequent section attempt to address this problem.

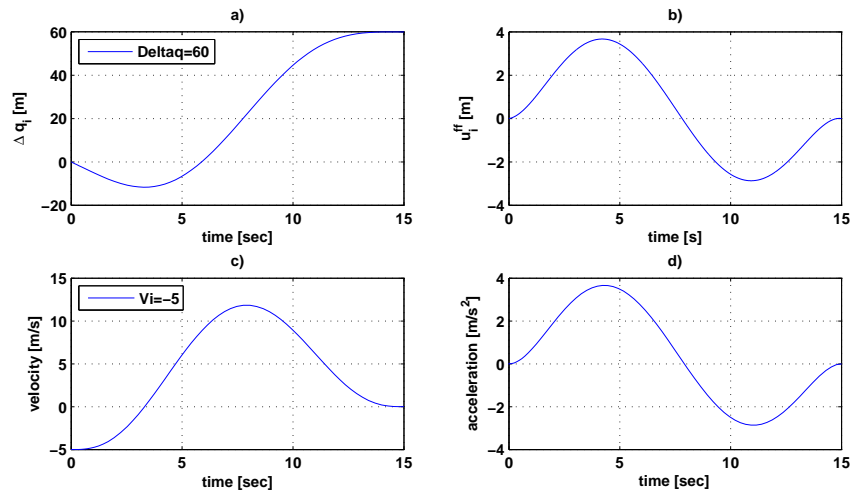


Figure 19: Results for single polynomial with $\Delta q_i = 60$, $V_i = -5$ a) Desired distance b) Input signal c) Velocity d) Acceleration.

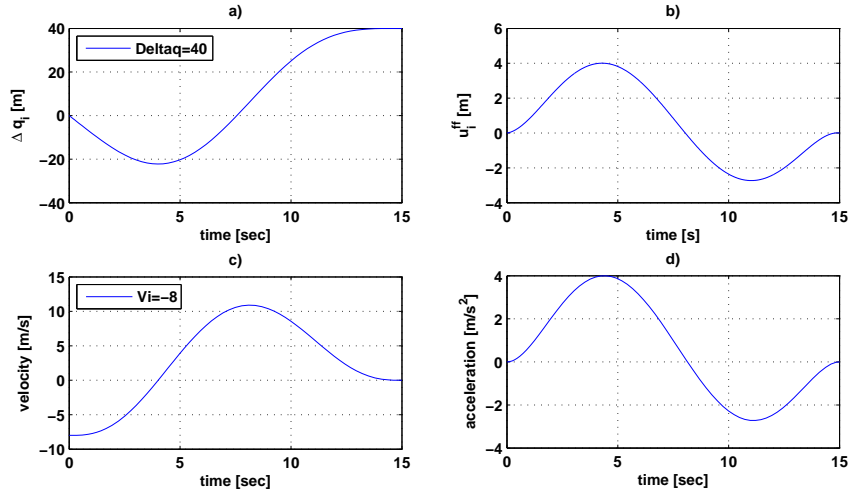


Figure 20: Results for single polynomial with $\Delta q_i = 40$, $V_i = -8$ a) Desired distance b) Input signal c) Velocity d) Acceleration.

3.5 Trajectory Computation with 3 Polynomials Using Nonlinear Optimization

It was concluded from Section 3.4 that the usage of a single polynomial with degree 7 leads to considerable violations of the velocity and acceleration limitations. In the suggested method, we propose to concatenate three polynomials of degree 4 and to formulate a nonlinear optimization problem whose solution meets all constraints.

3.5.1 Trajectory Computation

To this end, we divide the time interval $[0, T]$ for the maneuver into three parts and use a trajectory $\Delta q(t)$ as follows.

$$\Delta q(t) = \begin{cases} p_1(t) & \text{for } 0 \leq t \leq t_1 \\ p_2(t) & \text{for } t_1 \leq t \leq t_2 \\ p_3(t) & \text{otherwise} \end{cases}$$

$$p_1(t) = v_0 + v_1 t + v_2 t^2 + v_3 t^3 + v_4 t^4, \quad (3.42)$$

$$p_2(t) = w_0 + w_1 t + w_2 t^2 + w_3 t^3 + w_4 t^4, \quad (3.43)$$

$$p_3(t) = z_0 + z_1 t + z_2 t^2 + z_3 t^3 + z_4 t^4. \quad (3.44)$$

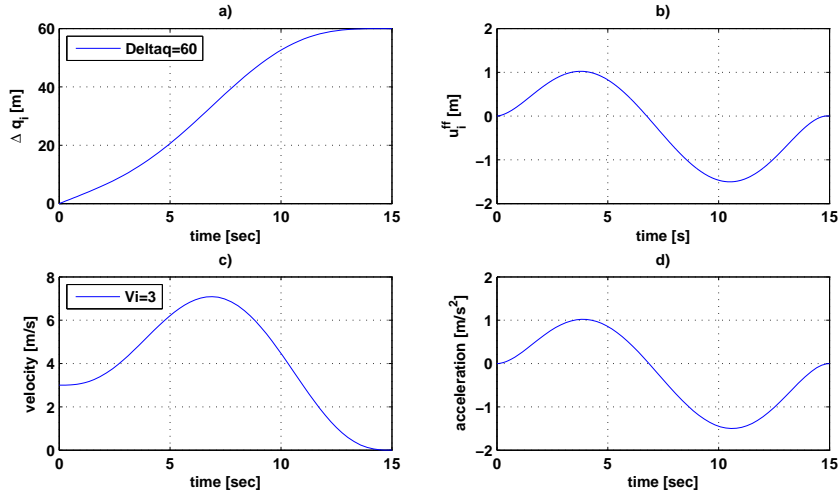


Figure 21: Results for single polynomial with $\Delta q_i = 60$, $V_i = 3$ a) Desired distance b) Input signal c) Velocity d) Acceleration.

Then, we list the constraints for the coefficients that arise from the conditions at time 0 and time T as well as the constraints at the intermediate times t_1 and t_2 . At time $t = 0$, it must hold that

$$p_1(0) = v_0 = 0, \quad (3.45)$$

$$\frac{d p_1(0)}{d t} = v_1 = V_i, \quad (3.46)$$

$$\frac{d^2 p_1(0)}{d t^2} = 2v_2 = 0. \quad (3.47)$$

At $t = t_1$, it must hold that

$$\begin{aligned} p_2(t_1) &= w_0 + w_1 t_1 + w_2 t_1^2 + w_3 t_1^3 + w_4 t_1^4 = p_1(t_1) \\ &= v_0 + v_1 t_1 + v_2 t_1^2 + v_3 t_1^3 + v_4 t_1^4, \end{aligned} \quad (3.48)$$

$$\begin{aligned} \frac{d p_2(t_1)}{d t} &= w_1 + 2w_2 t_1 + 3w_3 t_1^2 + 4w_4 t_1^3 \\ &= \frac{d p_1(t_1)}{d t} = v_1 + 2v_2 t_1 + 3v_3 t_1^2 + 4v_4 t_1^3 \end{aligned} \quad (3.49)$$

$$\frac{d^2 p_2(t_1)}{d t^2} = 2w_2 + 6w_3 t_1 + 12w_4 t_1^2 = \frac{d^2 p_1(t_1)}{d t^2} = 2v_2 + 6v_3 t_1 + 12v_4 t_1^2. \quad (3.50)$$

At $t = t_2$, it must hold that

$$\begin{aligned} p_3(t_2) &= z_0 + z_1 t_2 + z_2 t_2^2 + z_3 t_2^3 + z_4 t_2^4 = p_2(t_2) \\ &= w_0 + w_1 t_2 + w_2 t_2^2 + w_3 t_2^3 + w_4 t_2^4, \end{aligned} \quad (3.51)$$

$$\begin{aligned} \frac{d p_3(t_2)}{d t} &= z_1 + 2z_2 t_2 + 3z_3 t_2^2 + 4z_4 t_2^3 \\ &= \frac{d p_2(t_2)}{d t} = w_1 + 2w_2 t_2 + 3w_3 t_2^2 + 4w_4 t_2^3 \end{aligned} \quad (3.52)$$

$$\frac{d^2 p_3(t_2)}{d t^2} = 2z_2 + 6z_3 t_2 + 12z_4 t_2^2 = \frac{d^2 p_2(t_2)}{d t^2} = 2w_2 + 6w_3 t_2 + 12w_4 t_2^2. \quad (3.53)$$

Finally, it must hold at $t = T$ that

$$p_3(T) = z_0 + z_1 T + z_2 T^2 + z_3 T^3 + z_4 T^4 = \Delta q, \quad (3.54)$$

$$\frac{d p_3(T)}{d t} = z_1 + 2z_2 T + 3z_3 T^2 + 4z_4 T^3 = 0, \quad (3.55)$$

$$\frac{d^2 p_3(T)}{d t^2} = 2z_2 + 6z_3 T + 12z_4 T^2 = 0. \quad (3.56)$$

Using (3.42) to (3.56), linear constraints in the form

$$Ax = b$$

with the parameter vector

$$x = \left[v_0 \ v_1 \ v_2 \ v_3 \ v_4 \ w_0 \ w_1 \ w_2 \ w_3 \ w_4 \ z_0 \ z_1 \ z_2 \ z_3 \ z_4 \right]^T. \quad (3.57)$$

can be formulated. Since the matrix A is large, we write it in two parts as follows.

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$$

with

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & -1 & -t_1 & -t_1^2 & -t_1^3 & -t_1^4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 0 & -1 & -2t_1 & -3t_1^2 & -4t_1^3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6t_1 & 12t_1^2 & 0 & 0 & -2 & -6t_1 & -12t_1^2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.58)$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & t_2 & t_2^2 & t_2^3 & t_2^4 & -1 & -t_2 & -t_2^2 & -t_2^3 & -t_2^4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2t_2 & 3t_2^2 & 4t_2^3 & 0 & 1 & -2t_2 & -3t_2^2 & -4t_2^3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6t_2 & 12t_2^2 & 0 & 0 & -2 & -6t_2 & -12t_2^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & T & T^2 & T^3 & T^4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2T & 3T^2 & 4T^3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6T & 12T^2 \end{bmatrix}. \quad (3.59)$$

In addition, we get

$$b = \left[0 \quad V_i \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \Delta q \quad 0 \quad 0 \right]^T. \quad (3.60)$$

Finally, it is possible to formulate the velocity and acceleration constraints as nonlinear constraints. For the velocity, consider for polynomial $p_1(t)$ that

$$\dot{p}_1(t) = v_1 + 2v_2 t + 3v_3 t^2 + 4v_4 t^3$$

$$\ddot{p}_1(t) = 2v_2 + 6v_3 t + 12v_4 t^2$$

$$\dddot{p}_1(t) = 6v_3 + 24v_4 t$$

Evaluating $\ddot{p}_1(t) = 0$ gives the solutions

$$t_{1,2}^v = -\frac{3v_3 \pm \sqrt{9v_3^2 - 24v_2v_4}}{12v_4}.$$

That is, $\dot{p}_1(t)$ has a maximum or minimum at $t_{1,2}^v$. Accordingly, we formulate the nonlinear constraint

$$v_{\min} \leq \dot{p}_1(t_{1,2}^v) \leq v_{\max}. \quad (3.61)$$

Similarly, we compute for $\ddot{p}_1(t) = 0$ that

$$t^a = -\frac{v_3}{4v_4}.$$

That is, $\dot{p}_1(t)$ has a maximum or minimum at t^a and we add the constraint

$$a_{\min} \leq \ddot{p}_1(t^a) \leq a_{\max}. \quad (3.62)$$

The same constraints are added for the polynomials $p_2(t)$ and $p_3(t)$.

Then, we want to solve the nonlinear optimization problem

$$\min\{x^2\} \tag{3.63}$$

subject to the constraints

$$Ax = b \tag{3.64}$$

$$v_{\min} \leq \dot{p}_1(t_{1,2}^v) \leq v_{\max} \tag{3.65}$$

$$a_{\min} \leq \ddot{p}_1(t^a) \leq a_{\max} \tag{3.66}$$

$$v_{\min} \leq \dot{p}_2(t_{1,2}^v) \leq v_{\max} \tag{3.67}$$

$$a_{\min} \leq \ddot{p}_2(t^a) \leq a_{\max} \tag{3.68}$$

$$v_{\min} \leq \dot{p}_3(t_{1,2}^v) \leq v_{\max} \tag{3.69}$$

$$a_{\min} \leq \ddot{p}_3(t^a) \leq a_{\max}. \tag{3.70}$$

This optimization problem can be solved using the `fmincon` solver of Matlab. We next evaluate the same examples as in Section 3.3. Hereby, we represent the different trajectories by different colors as follows.

- Blue trajectories describe q_1 at times between 0 and determined t_1 ,
- Green trajectories describe q_2 at times between t_1 and determined t_2 ,
- Red trajectories describe q_3 at times between t_2 and determined t_3 .

The resulting plots for the four example scenarios are given below. Again, we choose $T = 15$.

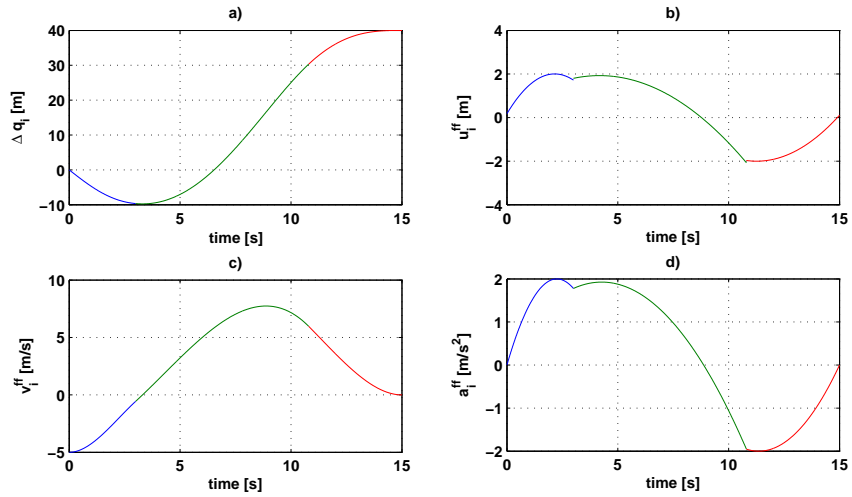


Figure 22: Optimal control's results with 3 polynomial at $\Delta q_i = 40$ and $V_i = -5$. a) Desired distance b) Input signal c) Velocity d) Acceleration.

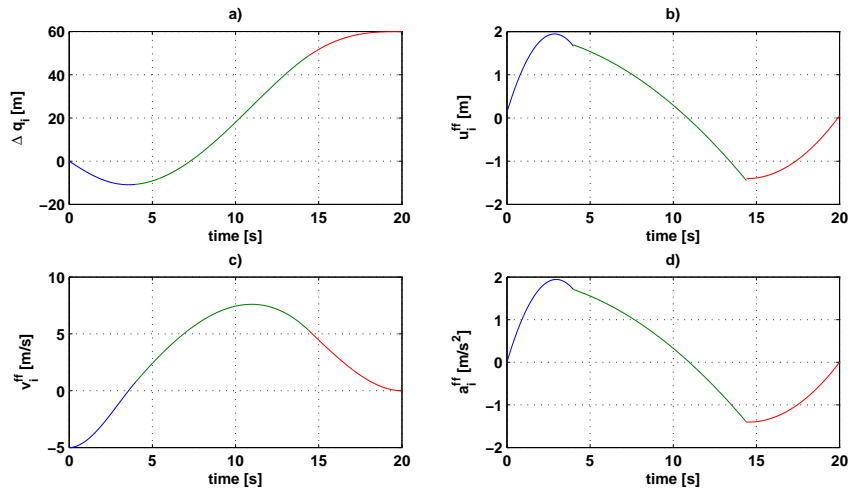


Figure 23: Optimal control's results with 3 polynomial at $\Delta q_i = 60$ and $V_i = -5$. a) Desired distance b) Input signal c) Velocity d) Acceleration.

In all example cases, an optimal solution was found and a trajectory that fulfills the given constraints could be obtained. Hereby, the trajectories in Fig. 22 and 23 reach the acceleration limit. Differently, the trajectory in Fig. 24 does not reach the lower acceleration limit. The reason is that this trajectory already starts with a lower velocity such that less deceleration is needed to open the desired gap. Finally, the trajectory in Fig. 25 does not reach the upper acceleration limit because the trajectory already starts with a larger velocity.

In summary, the proposed method with 3 polynomials is suitable for computing trajectories that meet the velocity and acceleration limits. On the one hand, the maneuver

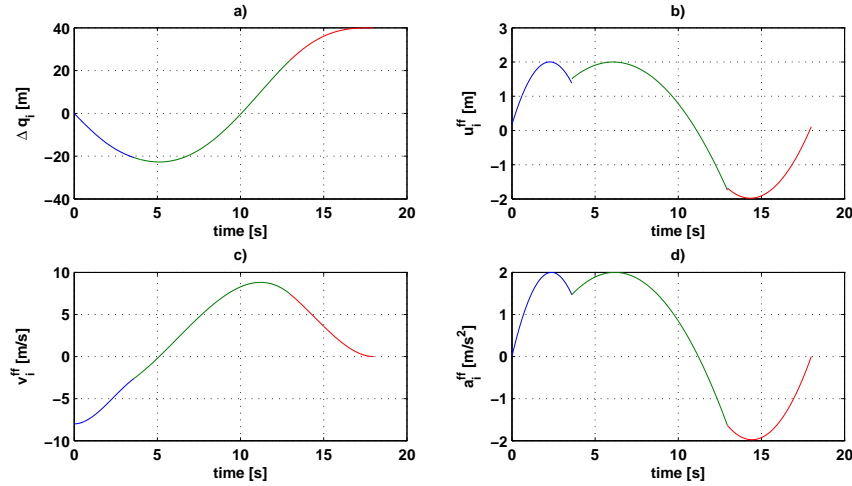


Figure 24: Optimal control's results with 3 polynomial at $\Delta q_i = 40$ and $V_i = -8$. a) Desired distance b) Input signal c) Velocity d) Acceleration.

time T has to be selected larger than the maneuver time for the optimal control solution in Section 3.3. On the other hand, the computation time for solving the nonlinear optimization problem is in the order of 1 sec and hence smaller than that of the optimal control problem. Nevertheless, it is still too large for a real-time implementation.

3.5.2 Parametrization of the Solution

Because of the large computation time when using optimization, we suggest to determine the solution x of the nonlinear optimization problem offline for a grid of parameter combinations of Δq and V_i . Then, we determine an interpolation of the found parameters in order to cover the whole relevant range of parameter combinations.

We consider a range of $\Delta q \in [20, 60]$ and a range of $V_i \in [-10, 10]$ for our evaluation. Then, we determine the solution $x_{\Delta q, V_i}$ for combinations of $\Delta q = 20 + q_{\text{grid}} j$, $j = 0, \dots, k_q$ and $V_i = -10 + v_{\text{grid}} k$, $k = 0, \dots, k_v$ as grid points. Here, q_{grid} and v_{grid} determine the granularity of the grid and k_q and k_v determine the number of grid points in the direction of Δq and V_i , respectively. That is, for each entry $x(l)$ of the solution vector x , we obtain the entries $x_{\Delta q, V_i}$ for all combinations $(\Delta q, V_i)$.

In order to compute the coefficient vector x for a given combination $(\Delta q, V_i)$ that does not belong to the computed grid, we first determine the neighboring grid points $\Delta q^l < \Delta q < \Delta q^u$ and $V_i^l < V_i < V_i^u$. Using the grid points $(\Delta q^l, V_i^l)$, $(\Delta q^u, V_i^l)$, $(\Delta q^l, V_i^u)$ and $(\Delta q^u, V_i^u)$ and the corresponding values $x_{(\Delta q^l, V_i^l)}$, $x_{(\Delta q^u, V_i^l)}$, $x_{(\Delta q^l, V_i^u)}$ and $x_{(\Delta q^u, V_i^u)}$, we compute a two-dimensional linear interpolation polynomial $p_l(a, b) = p_i^{0,0} + p_i^{1,0} a + p_i^{0,1} b$ in order to represent $x_{\Delta q, V_i}(l)$ in the whole range of $[\Delta q^l, \Delta q^u] \times [V_i^l, V_i^u]$. Hereby,

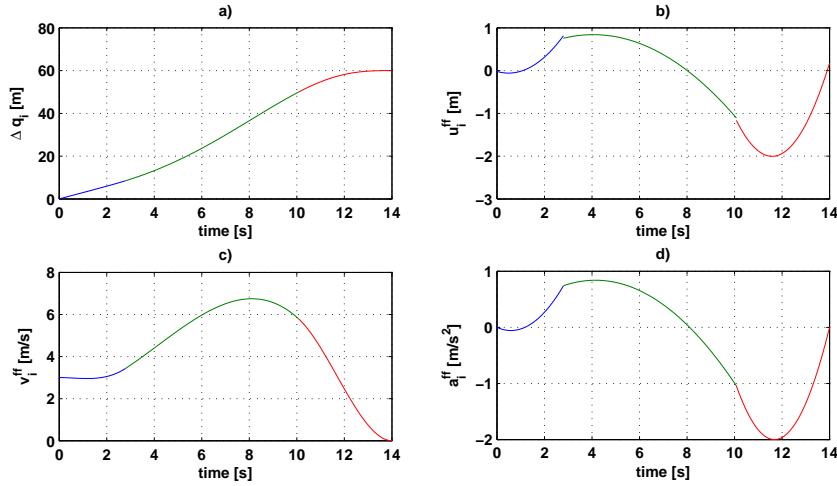


Figure 25: Optimal control's results with 3 polynomial at $\Delta q_i = 60$ and $V_i = 3$. a) Desired distance b) Input signal c) Velocity d) Acceleration.

we assume that the coefficient values change smoothly between the computed values. Examples for $x(4) = v_3$ and $x = w_2$ are shown in Fig. 26.

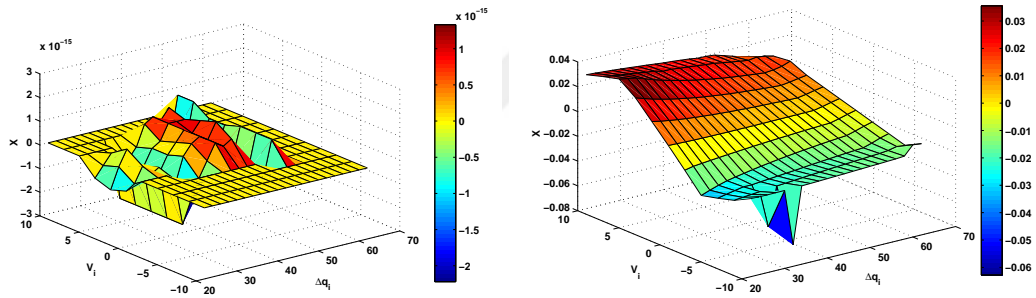


Figure 26: $x_{\Delta q, V_i}(4) = v_3$ and $x_{\Delta q, V_i}(8) = w_2$ for the specified values of Δq and V_i .

Then, using $p_l(a, b)$, each coefficient $x(l)$ for a given combination of Δq and V_i can be evaluated as $p_l(\Delta q, V_i)$. That is, storing p_l for $l = 1, \dots, 15$ in the form of a look-up table allows a very fast computation of an approximation of the trajectory for each value of Δq and V_i .

In order to illustrate this computation, we determine trajectories for values of Δq and V_i that are between the computed supporting points. In the example, we use $\Delta q = 24.3$ m and $V_i = -2.7$ m/s, $q_{\text{grid}} = 0.5$ m and $v_{\text{grid}} = 0.5$ m/s. That is, the neighboring

grid points are

$$\begin{aligned}(\Delta q^l, V_i^l) &= (24.0, -3.0), & (\Delta q^u, V_i^l) &= (24.5, -3.0), \\(\Delta q^l, V_i^u) &= (24.0, -2.5), & (\Delta q^u, V_i^u) &= (24.5, -2.5).\end{aligned}$$

For example, inspecting the coefficient $x(5) = v_4$, the corresponding values

$$\begin{aligned}x_{(24.0, -3.0)}(5) &= -0.0223, & x_{(24.5, -3.0)}(5) &= -0.0221, \\x_{(24.0, -2.5)}(5) &= -0.0199, & x_{(24.5, -2.5)}(5) &= -0.0197\end{aligned}$$

are obtained. The corresponding interpolation polynomial is

$$p_5(a, b) = -0.02 + 4.992 \cdot 10^{-4} a + 4.748 \cdot 10^{-3} b.$$

Then, the approximated coefficient $x(5)$ for the given values $\Delta q = 24.3$ m and $V_i = -2.7$ m/s is

$$p_5(24.3, -2.7) = -0.0208.$$

The same computation can be performed to obtain all coefficients x and hence determine the trajectory for the parameters $\Delta q = 24.3$ m and $V_i = -2.7$ m/s. The resulting trajectory is shown in Fig. 27.

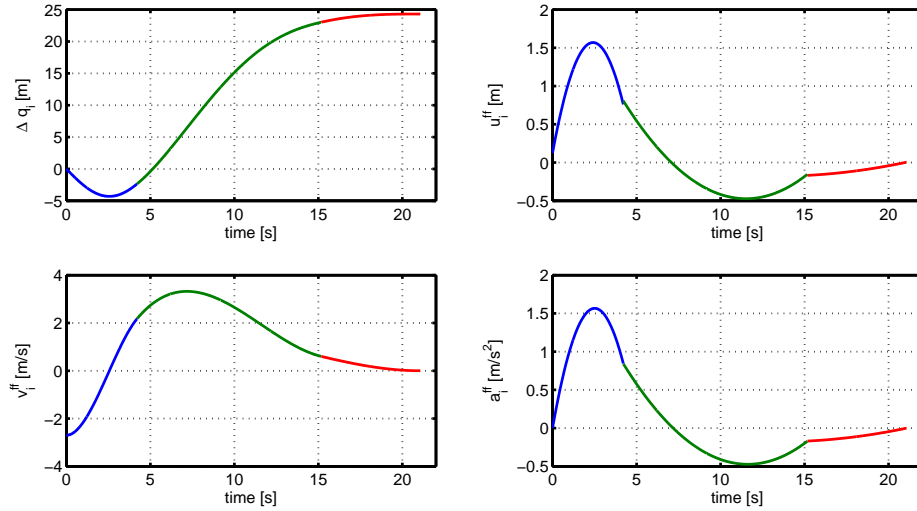


Figure 27: Approximated trajectory for $\Delta q = 24.3$ m and $V_i = -2.7$ m/s.

It is readily observed that the trajectory very well approximates the desired behavior. The final distance value is ≈ -24.3 m and the final velocities and accelerations are very

close to zero. In addition, the velocity and acceleration constraints are met. A similar observation can be made in the next example with $\Delta q = 55.4$ m and $V_i = -8.3$ m/s. Here, the distance value is larger and the initial velocity is smaller. Nonetheless, a suitable trajectory is obtained as can be seen in Fig. 28.

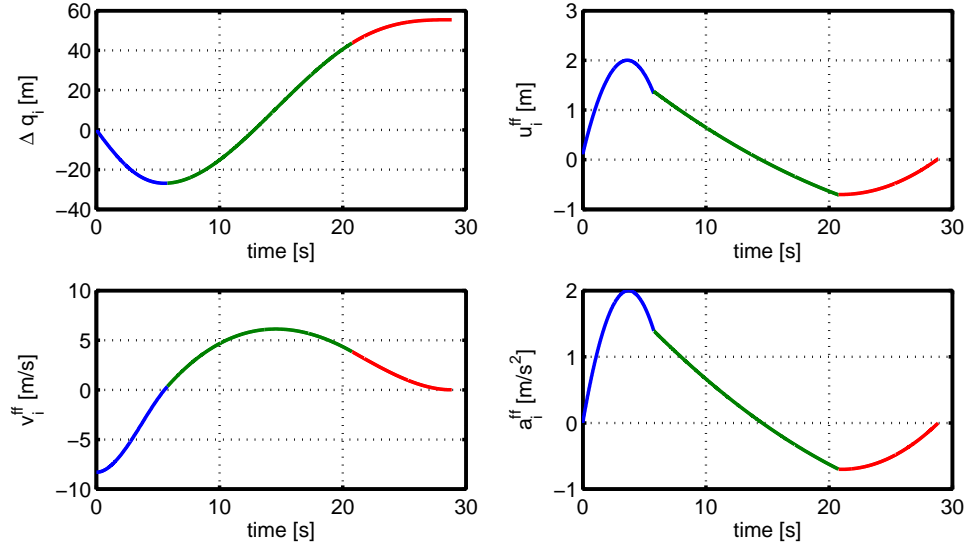


Figure 28: Approximated trajectory for $\Delta q = 55.4$ m and $V_i = -8.3$ m/s.

In summary, this section proposes a method for computing trajectories for opening/closing gaps using 3 concatenated polynomials. First, it is shown that trajectories that meet the formulated constraints can be computed using nonlinear programming. Since it turns out that obtaining trajectories by nonlinear programming is too time-consuming for a real-time implementation, an approximation of the obtained trajectories is suggested. To this end, a sufficient number of trajectories is generated for a grid of values in Δq and V_i using nonlinear programming. Then, the trajectory for each parameter combination $(\Delta q, V_i)$ that is not on the grid can be found using a similar linear interpolation. Our test results show that the resulting trajectories are suitable for the desired open/close gap maneuvers and can be computed in very short time. The only disadvantage of this method is the usually larger completion times for maneuvers compared to the optimal control solution in Section 3.3.

3.6 Approximation of the Optimal Control Trajectory

The next method attempts to obtain a good polynomial approximation of the optimal control trajectory in order to reduce the completion time of each maneuver compared to the method in Section 3.5. To this end, we take a similar approach to the previous

section. We first find polynomial approximations of pre-computed optimal control solutions for combinations of $\Delta q = 20 + q_{\text{grid}} j$, $j = 0, \dots, j_{\text{grid}}$ and $V_i = -10 + v_{\text{grid}} k$, $k = 0, \dots, k_{\text{grid}}$. Then, the same linear interpolation as in Section 3.5 is used to determine the polynomial coefficients for combinations of Δq and V_i that are not on the grid.

Consider the optimal control solution $q_{\Delta q, V_i}(t)$ for a given combination $(\Delta q, V_i)$. Regarding the approximation of the optimal control trajectory, a polynomial

$$p(t) = p_0 + p_1 t + \dots + p_{l-1} t^{l-1} + p_l t^l \quad (3.71)$$

is used. We first evaluate the initial and terminal conditions for this polynomial. It must hold that

$$p(0) = p_0 = 0 \quad (3.72)$$

$$\dot{p}(0) = p_1 = V_i \quad (3.73)$$

$$\ddot{p}(0) = 2p_2 = 0 \quad (3.74)$$

$$p(T) = p_0 + p_1 T + \dots + p_{l-1} T^{l-1} + p_l T^l = 0 \quad (3.75)$$

$$\dot{p}(T) = p_1 + 2p_2 T + \dots + (l-1)p_{l-1} T^{l-2} + l p_l T^{l-1} = 0 \quad (3.76)$$

$$\ddot{p}(T) = 2p_2 + \dots + (l-1)(l-2)p_{l-1} T^{l-3} + l(l-1)p_l T^{l-2} = 0 \quad (3.77)$$

That is, the first three coefficients of this polynomial are already fixed by (3.72) to (3.74) and there are three more linear constraint equations for the remaining coefficients in (3.75) to (3.77). We next solve these equations to eliminate three more coefficients of $p(t)$ using

$$\begin{aligned} p_3 T^3 + p_4 T^4 + p_5 T^5 &= -V_i T - p_6 T^6 - \dots - p_{l-1} T^{l-1} - p_l T^l, \\ 3p_3 T^2 + 4p_4 T^3 + 5p_5 T^4 &= -V_i - 6p_6 T^5 - \dots - (l-1)p_{l-1} T^{l-2} - l p_l T^{l-1}, \\ 6p_3 T + 12p_4 T^2 + 20p_5 T^3 &= -30p_6 T^4 - \dots - (l-1)(l-2)p_{l-1} T^{l-3} - \\ &\quad l(l-1)p_l T^{l-2}. \end{aligned}$$

This can be written as

$$\begin{bmatrix} T^3 & T^4 & T^5 \\ 3T^2 & 4T^3 & 5T^4 \\ 6T & 12T^2 & 20T^3 \end{bmatrix} \begin{bmatrix} p_3 \\ p_4 \\ p_5 \end{bmatrix} = A \begin{bmatrix} p_3 \\ p_4 \\ p_5 \end{bmatrix} = B \begin{bmatrix} p_6 \\ \vdots \\ p_l \end{bmatrix} \Rightarrow \begin{bmatrix} p_3 \\ p_4 \\ p_5 \end{bmatrix} = A^{-1} B \begin{bmatrix} p_6 \\ \vdots \\ p_l \end{bmatrix}.$$

That is, the coefficients p_0, \dots, p_5 can be represented by the coefficients p_6, \dots, p_l .

Hereby, it is ensured that all initial and terminal conditions are met. In order to determine the remaining coefficients, we perform a nonlinear optimization that tries to minimize the least square error $(p(t) - q_{\Delta q, V_i})^2$ between the polynomial approximation and the optimal control solution as

$$\min_{p_6, \dots, p_l} \int_{t=0}^T (p(t) - q_{\Delta q, V_i}(t))^2 dt. \quad (3.78)$$

Then, we perform the same approximation with a two-dimensional linear polynomial for values $(\Delta q, V_i)$ that are not on the grid. The polynomial order that was found suitable in our experiments is $l = 11$.

We consider the same examples as in Section 3.5. First, let $\Delta q = 24.3$ m and $V_i = -2.7$ m/s. The resulting trajectory is shown in Fig. 29. It can be seen that, as specified, Δq_i reaches the desired terminal value, whereas the terminal velocity and acceleration are zero. Due to the polynomial approximation, there is a violation of the acceleration constraint. Nevertheless, in comparison to Fig. 27 the maneuver can be completed in a much shorter time.

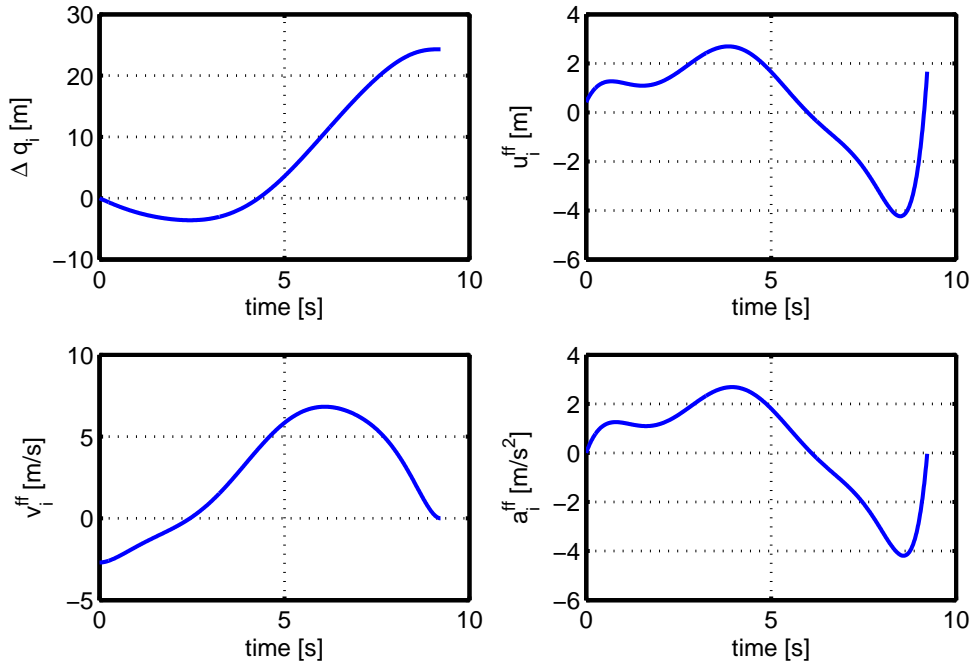


Figure 29: Approximated trajectory for $\Delta q = 24.3$ m and $V_i = -2.7$ m/s.

A similar observation is made for $\Delta q = 54.4$ m and $V_i = -8.3$ m/s in Fig. 30 in comparison to Fig. 28.

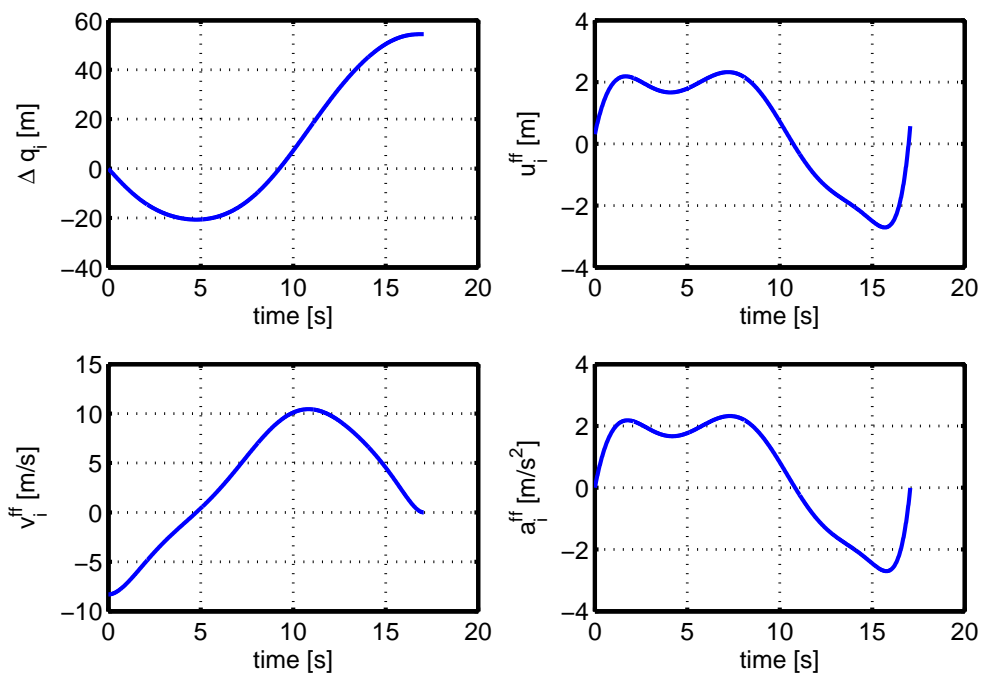


Figure 30: Approximated trajectory for $\Delta q = 54.4$ m and $V_i = -8.3$ m/s.

3.7 Piecewise of Approximation of the Optimal Control Trajectory

Our last method is based on the evaluation of the general shape of the acceleration signal when opening/closing a gap. This shape is shown together with the velocity and position signal in Figure 31.

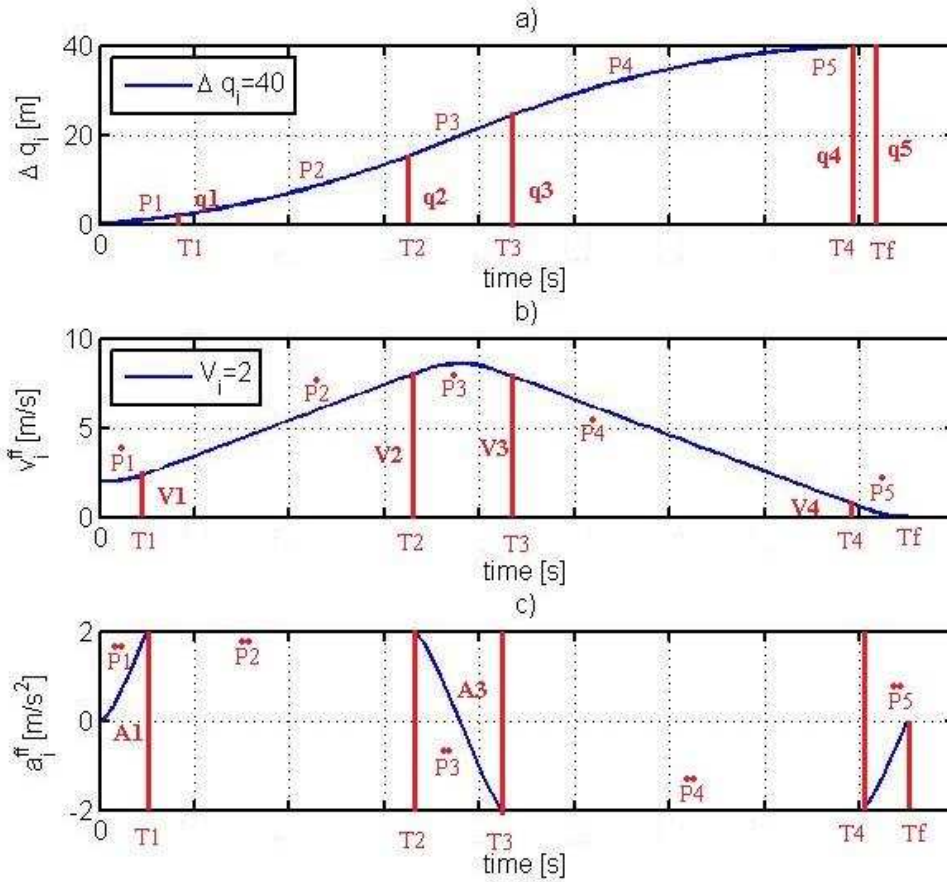


Figure 31: Piecewise Approximation of the Optimal Control trajectory figures a) Desired distance b) Velocity c) Acceleration

It can be observed from the figure that the acceleration signal can be divided into 5 different parts. The respective time instants are denoted as T_1, T_2, T_3, T_4 and T_f and the time durations of each part are $\Delta T_1, \Delta T_2, \Delta T_3, \Delta T_4, \Delta T_5$. Then, it holds that the acceleration increases from 0 to its maximum value until T_1 , the acceleration is constant a_{\max} between T_1 and T_2 , the acceleration decreases from its maximum value to its minimum value between T_2 and T_3 , the acceleration assumes its minimum value between T_3 and T_4 and the acceleration becomes zero again at the end of the maneuver. In addition, the velocity starts from the initial value V_i and becomes zero at the end of the maneuver

and the position difference evolves from 0 to Δq . Defining $T_0 = 0$, we denote the position, velocity and acceleration values of the optimal control solution at the respective times instants as $\Delta q_i = \Delta q(T_i)$, $v_i = v(T_i)$ and $a_i = a(T_i)$ for $i = 0, \dots, 5$. In particular, $\Delta q_0 = 0$, $\Delta q_5 = \Delta q$, $v_0 = V_i$, $v_5 = 0$, $a_0 = 0$ and $a_5 = 0$.

Respecting this separation into five parts, we propose to concatenate five polynomials of appropriate degrees and to determine their coefficients using the given conditions on the acceleration, velocity and position. The polynomials capture the position Δq as follows

$$\Delta q(t) = \begin{cases} P_1(t) = a_1 + b_1 t + c_1 t^2 + d_1 t^3 + e_1 t^4 + f_1 t^5 & \text{for } 0 \leq t \leq T_1 \\ P_2(t) = a_2 + b_2 t + c_2 t^2 & \text{for } T_1 \leq t \leq T_2 \\ P_3(t) = a_3 + b_3 t + c_3 t^2 + d_3 t^3 + e_3 t^4 + f_3 t^5 & \text{for } T_2 \leq t \leq T_3 \\ P_4(t) = a_4 + b_4 t + c_4 t^2 & \text{for } T_3 \leq t \leq T_4 \\ P_5(t) = a_5 + b_5 t + c_5 t^2 + d_5 t^3 + e_5 t^4 + f_5 t^5 & \text{for } T_4 \leq t \leq T_f \end{cases} \quad (3.79)$$

Hereby, the polynomials P_2 and P_4 are chosen with degree 2 since the respective parts of the trajectory show a constant acceleration. The polynomials P_1 , P_3 , P_5 are chosen with degree 5 in order to provide a sufficient number of coefficients for the existing conditions.

The conditions for the polynomial coefficients are for $i = 1, 2, 3, 4, 5$

$$P_i(0) = q_{i-1} \text{ and } P_i(T_i) = \Delta q_i. \quad (3.80)$$

$$\dot{P}_i(0) = v_{i-1} \text{ and } \dot{P}_i(T_i) = v_i. \quad (3.81)$$

$$\ddot{P}_i(0) = a_{i-1} \text{ and } \ddot{P}_i(T_i) = a_i \quad (3.82)$$

Using the above conditions, it directly follows that

$$P_2(t) = q_1 + v_1 t + a_{\max} t^2 \quad (3.83)$$

$$P_4(t) = q_3 + v_3 t + a_{\min} t^2 \quad (3.84)$$

In addition, we consider the time derivatives of the remaining polynomials for $i = 1, 3, 5$:

$$\dot{P}_i(t) = b_i + 2c_i t + 3d_i t^2 + 4e_i t^3 + 5f_i t^4. \quad (3.85)$$

$$\ddot{P}_i(t) = 2c_i + 6d_i t + 12e_i t^2 + 20f_i t^3. \quad (3.86)$$

Then, the following equations for the polynomial coefficients are obtained for $i =$

1,3,5:

$$P_i(0) = a_i = q_{i-1}. \quad (3.87)$$

$$P_i(\Delta T_i) = a_i + b_i \Delta T_i + c_i \Delta T_i^2 + d_i \Delta T_i^3 + e_i \Delta T_i^4 + f_i \Delta T_i^5 = q_i. \quad (3.88)$$

$$\dot{P}_i(0) = b_i = v_{i-1}. \quad (3.89)$$

$$\dot{P}_i(\Delta T_i) = b_i + 2c_i \Delta T_i + 3d_i \Delta T_i^2 + 4e_i \Delta T_i^3 + 5f_i \Delta T_i^4 = v_i. \quad (3.90)$$

$$\ddot{P}_i(0) = 2c_i = a_{i-1}. \quad (3.91)$$

$$\ddot{P}_i(\Delta T_i) = 2c_i + 6d_i \Delta T_i + 12e_i \Delta T_i^2 + 20f_i \Delta T_i^3 = a_i. \quad (3.92)$$

Considering that a_i, b_i, c_i are already known from (3.87), (3.89) and (3.91), it remains to compute d_i, e_i, f_i for $i = 1, 3, 5$ from (3.88), (3.90) and (3.92). Using the vector of unknown parameters

$$r_i = \begin{bmatrix} d_i & e_i & f_i \end{bmatrix}, \quad (3.93)$$

we obtain the following linear equation

$$\underbrace{\begin{bmatrix} \Delta T_i^3 & \Delta T_i^4 & \Delta T_i^5 \\ 3\Delta T_i^2 & 4\Delta T_i^3 & 5\Delta T_i^4 \\ 6\Delta T_i & 12\Delta T_i^2 & 20\Delta T_i^3 \end{bmatrix}}_{M_i} r_i = \underbrace{\begin{bmatrix} q_i - v_i \Delta T_i - q_{i-1} - \frac{a_{i-1}}{2} \Delta T_i^2 \\ v_i - v_{i-1} - a_{i-1} \Delta T_i \\ a_i - a_{i-1} \end{bmatrix}}_{l_i}. \quad (3.94)$$

Since M_i is invertible, the polynomial coefficients are obtained by computing

$$r_i = M_i^{-1} \cdot l_i. \quad (3.95)$$

Computing a solution with the symbolic toolbox of Matlab, the polynomial coefficients are determined as

$$d_i = \frac{20q_i - 20q_{i-1} - 28\Delta T_i v_i + 8\Delta T_i v_{i-1} + \Delta T_i^2 a_i - 3\Delta T_i^2 a_{i-1}}{2\Delta T_i^3}$$

$$e_i = -\frac{30q_i - 30q_{i-1} - 44\Delta T_i v_i + 14\Delta T_i v_{i-1} + 2\Delta T_i^2 a_i - 3\Delta T_i^2 a_{i-1}}{2\Delta T_i^4}$$

$$f_i = \frac{12q_i - 12q_{i-1} - 18\Delta T_i v_i + 6\Delta T_i v_{i-1} + \Delta T_i^2 a_i - \Delta T_i^2 a_{i-1}}{2\Delta T_i^5}$$

That is, the polynomial P_i can be evaluated analytically for all combinations of $\Delta T_i, q_i, q_{i-1}, v_i, v_{i-1}, a_i$ and a_{i-1} for $i = 1, 3, 5$. Considering that P_2 and P_4 are already given by (3.83) and (3.84), the polynomial approximation of the optimal control trajectory in

(3.79) is fully determined. In addition, it is possible to determine the input signal u^{ff} by plant inversion. We again separate the input signal into five parts such that

$$u(t) = \begin{cases} u_1(t) = & \text{for } 0 \leq t \leq T_1 \\ u_2(t) = & \text{for } T_1 \leq t \leq T_2 \\ u_3(t) = & \text{for } T_2 \leq t \leq T_3 \\ u_4(t) = & \text{for } T_3 \leq t \leq T_4 \\ u_5(t) = & \text{for } T_4 \leq t \leq T_f \end{cases} \quad (3.96)$$

Using the plant model and noting that $q_i(t) = P_i(t)$, it holds that

$$U_i(s) = (\tau s^3 + s^2) Q_i(s) \Rightarrow u(t) = \tau \cdot \frac{d^3 p_i(t)}{dt^3} + \frac{d^2 p_i(t)}{dt^2}. \quad (3.97)$$

Hence,

$$u_i(t) = \begin{cases} 2a_{\max} & \text{for } i = 2 \\ 2a_{\min} & \text{for } i = 4 \\ \tau(6d_i + 24e_i t + 60f_i t^2) + 2c_i + 6d_i t + 12e_i t^2 + 20f_i t^3 & \text{for } i = 1, 3, 5 \end{cases} \quad (3.98)$$

In order to evaluate the quality of this approximation, we consider the same examples as in Section 3.5. First, let $\Delta q = 24.3$ m and $V_i = -2.7$ m/s. The resulting trajectory is shown in color in Fig. 32 together with the trajectories of the neighboring grid points. It can be seen that the desired maneuver is performed in the same time as the optimal control solution. In addition, the velocity constraints and acceleration constraints are met.

Secondly, let $\Delta q = 59$ m and $V_i = -10$ m/s. The resulting trajectory is shown in Fig. 33. Again, the desired maneuver is performed without violating the constraints. It has to be noted that the proposed interpolation method allows determining suitable trajectories in practically no time.

3.8 Comparison

We next perform a comparison of the different feedforward trajectories determined in the scope of this thesis. The methods used are listed as

- 1: optimal control
- 2: using a single polynomial and plant inversion

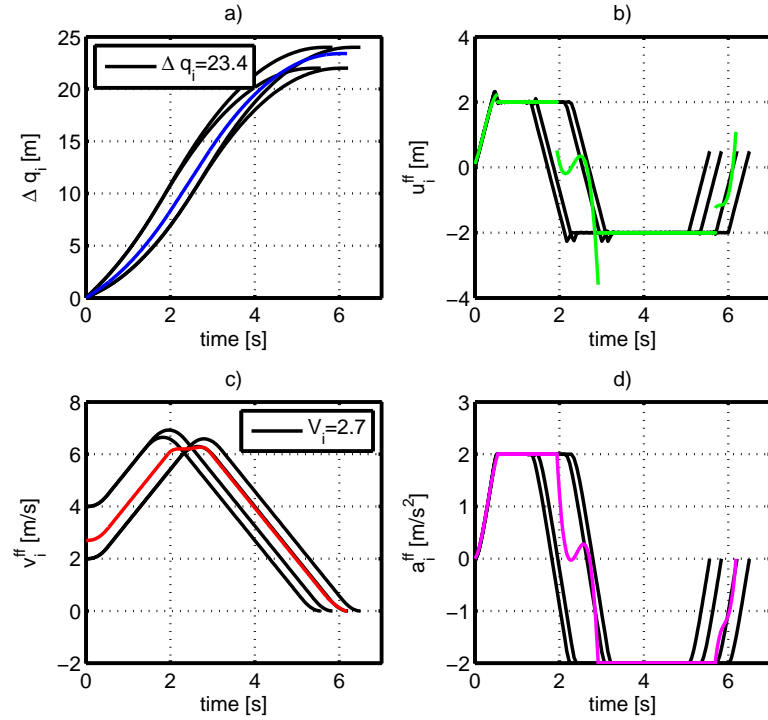


Figure 32: Piecewise Approximation of the Optimal Control trajectory for $\Delta q = 24.3$ m and $V_i = -2.7$ m/s.

- 3: nonlinear optimization
- 4: optimal control approximation with a single polynomial
- 5: piecewise optimal control approximation with five polynomials

Fig. 34 shows the computation for opening a gap of $\Delta q = 40$ m, whereby the initial velocity difference between vehicle i and its predecessor vehicle is $V_i = -4$ m/s. It can be seen that methods 1,4,5 perform the maneuver in a shorter time compared to methods 2. This is due to the fact that method 1 constitutes the optimal control solution for a minimum time maneuver. Methods 4,5 are approximations of the optimal control solution and hence also perform the maneuver in the shortest possible time. Hereby, the approximation with a single high-order polynomial shows some deviation from the actual optimal control solution. In particular, the acceleration constraint of ± 2 m/s² is violated when using this trajectory. In contrast, trajectory 5 very closely approximates the optimal control trajectory. This is due to the fact that the piecewise approximation is chosen in order to follow the characteristic shape of the acceleration trajectory.

In summary, we conclude that the method in Section 3.7 is most suitable for computing input signal trajectories for open/close gap maneuvers. This method requires

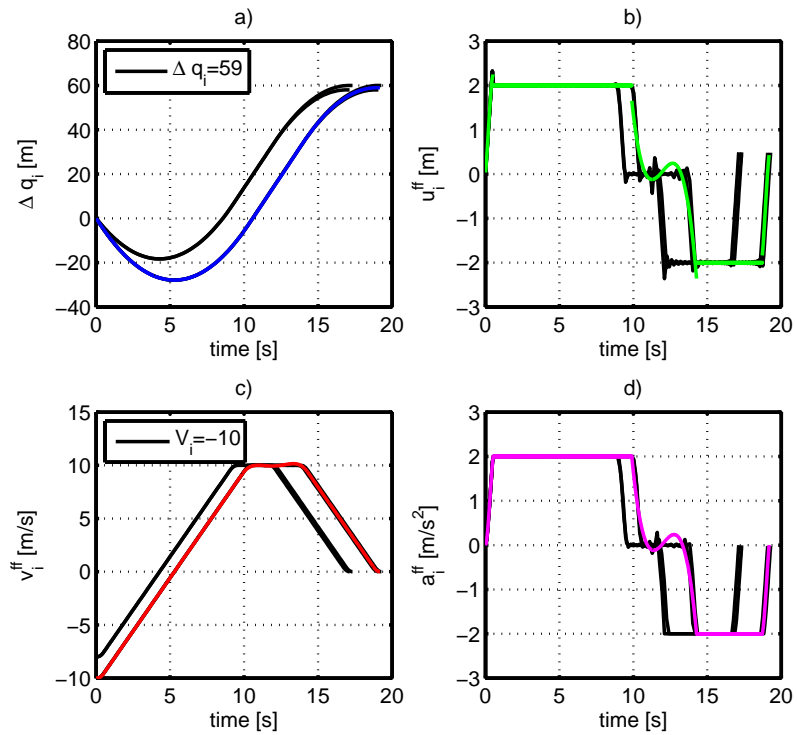


Figure 33: Piecewise Approximation of the Optimal Control trajectory for $\Delta q = 59$ m and $V_i = -10$ m/s.

computing the optimal control solution for a grid of values for the desired gap distance Δq and the initial velocity difference V_i of the vehicles, which can be done offline. Then, the actual trajectories for any combination of Δq and V_i are evaluated by a simple linear interpolation, which can be performed in real time.

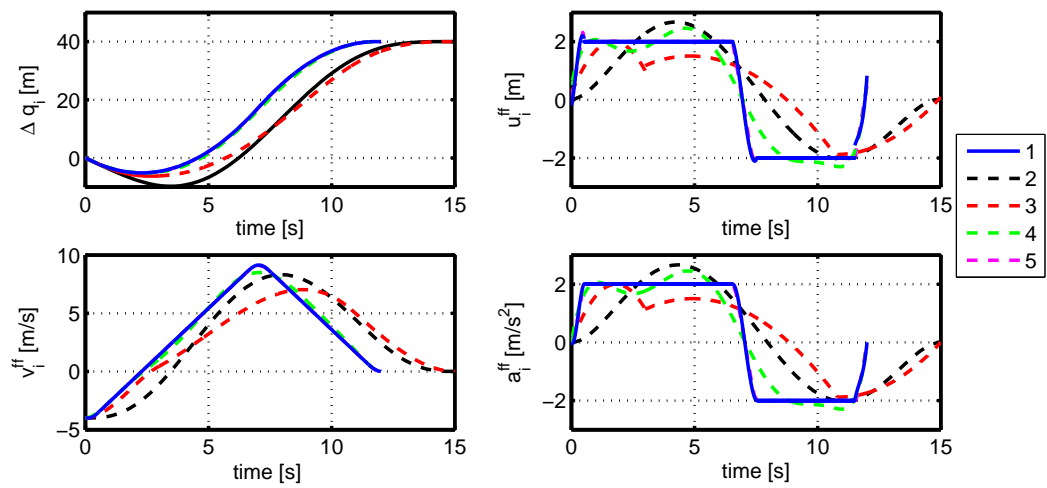


Figure 34: Comparison of different feedforward signals for $\Delta q = 40$ m and $V_i = -4$ m/s.

CHAPTER 4

VEHICLE IMPLEMENTATION IN THE FORM OF AN S-FUNCTION

The aim of this chapter is the development of a vehicle model that performs vehicle following using CACC and additional open/close gap maneuvers in Simulink. In order to obtain a model that can be used in the scope of a large simulation with many vehicles, the vehicle model is implemented in the form of an S-function. Hence, the chapter first gives a brief description of S-functions in Simulink in Section 4.1. Then, requirements for the desired S-function are summarized in Section 4.2. Implementation details are given in Section 4.3.

4.1 General S-Function Description

S-functions (system-functions) are a powerful tool in order to efficiently use Simulink. Using an S-Function a simulation block in Simulink can be programmed in C, C++ or another computer language. S-functions depend on sub-functions that can be loaded and executed by the MATLAB execution engine by itself. There are many applications to use S-functions such as;

- To create new Simulink blocks for a special aim
- To add blocks that describe a hardware device
- To define a dynamic system for certain mathematical equations
- To use graphical animations

S-function offer various callback methods. Some of them are mandatory, whereas others are optional when implementing an S-function. The most important required callback methods are;

- `mdlInitializeSizes`
- `mdlInitializeSampleTimes`
- `mdlOutputs`

- mdlTerminate

We next describe the most important callback functions and their usage.

4.1.1 mdlInitializeSizes

Required Syntax: void mdlInitializeSizes(SimStruct *S)

This function defines the number of inputs, outputs, states, parameters, and other items of the S-function.

- Define the number of parameters using `ssSetNumSFcnParams`.
- Define the number of states using `ssSetNumContStates` and `ssSetNumDiscStates`.
- Define the number of input ports using `ssSetNumInputPorts`.
- Define the dimensions of the input ports using `ssSetInputPortWidth` or `ssSetInputPortMatrixDimensions`.
- Define whether it has direct feed-through for each inputs using `ssSetInputPortDirectFeedThrough`.
- Define the number of output ports using `ssSetNumOutputPorts`.
- Define the dimensions of the outputs `mdlSetOutputPortWidth`.
- Define the number of DWork vectors using `ssSetNumDWork`.
- Define the dimensions of the DWork vectors using `ssSetDWorkWidth`.
- Define the stored data type of a DWork vector using `ssSetDWorkDataType`.
- Define the name of a stored data type work vector using `ssSetDWorkName`.

4.1.2 mdlInitializeSampleTimes

Required Syntax: void mdlInitializeSampleTimes(SimStruct *S)

This function describes the sample time(s) in the S-function.

- Define the sample time for each sample rate using `ssSetSampleTime(S, sampleTimeIndex, sample-time)`.
- Define the offset time for each sample rate using `ssSetOffsetTime(S, offsetTimeIndex, offset-time)`.

4.1.3 mdlOutputs

Required Syntax: `void mdlOutputs(SimStruct *S, int-T tid)` This function computes the output signals of the S-function.

- Get the numeric type of an output port using `Signal-T ssGetOutputPortComplexSignal(SimStruct *S, int-T port)`.
- Get a pointer to an output signal of type double using `real-T *ssGetOutputPortRealSignal(SimStruct *S, int-T port)`.
- Get the vector of signal elements emitted by an output port using `void *ssGetOutputPortSignal(SimStruct *S, int-T port)`.

4.1.4 mdlTerminate

Required Syntax: `void mdlTerminate(SimStruct *S)`

When external termination of the simulation is required, this function will take precautions. There is not any function inside the S-function for terminating the simulation.

In addition to the mandatory callback methods, there are various optional callback methods. In the context of this thesis, the most important optional callback methods are;

- `mdlDerivatives`
- `mdlStart`
- `mdlUpdate`
- `mdlInitializeConditions`

4.1.5 mdlDerivatives

Required Syntax: `void mdlDerivatives(SimStruct *S)`

- Get the derivatives of a block's continuous states using `real-T *ssGetdX(SimStruct *S)`.

4.1.6 mdlStart

Required Syntax: `void mdlStart(SimStruct *S)`

Initialize the state vectors of the S-function.

4.1.7 mdlUpdate

Required Syntax: `void mdlUpdate(SimStruct *S, int-T tid)`

Update the state vectors in S-function.

4.1.8 mdlInitializeConditions

Required Syntax: `void mdlInitializeConditions(SimStruct *S)`

Initialize the DWork vectors (internal memory) in the S-function.

4.2 Requirements for Vehicle Model

Using the generic callback methods mentioned in the previous section, this section states requirements for the vehicle model to be realized in the scope of this thesis. We focus on a vehicle that can occupy two different lanes on a road and that has several surrounding vehicles and a road side unit (RSU) nearby. Vehicles communicated with each other using V2V communication and the vehicles and RSU communicate with infrastructure to vehicle communication (I2V). The following schematic illustrates the described scenario.

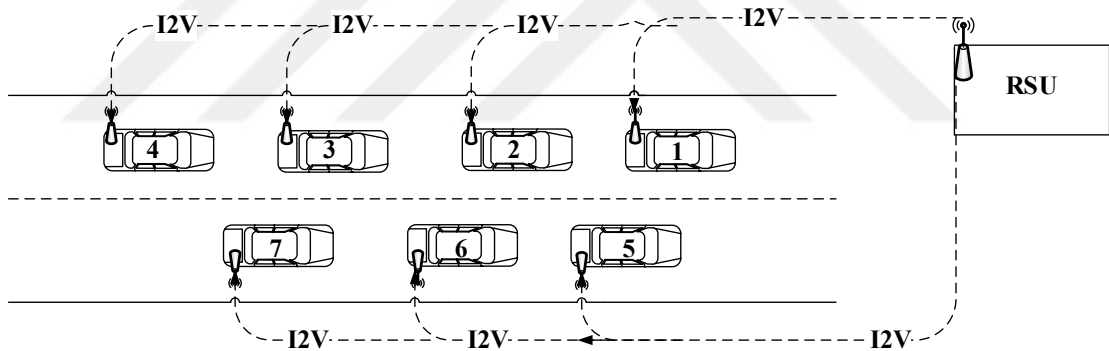


Figure 35: General model with vehicles and RSU.

In our vehicle S-function, the following implementation choices were made:

- The dynamic vehicle parameters, initial values and controller parameters are provided as block parameters
- Input, acceleration, position and velocity signals of all vehicles are collected by the RSU and provided to all vehicles (inputs U_{i-1} , $A_{i,i}$, V_{i-1} , Q_{i-1})
- The desired distance, and start times of the feedforward signals for open/close gap maneuvers come from the RSU (input T_s)

- The method in Section 3.7 is used for the computation of feedforward signals. The parameter values for the computed grid points are provided as input vectors (inputs X7 to X12 and T_f) to the S-function
- Each vehicle has a number and an initial value for the lane (inputs vehicle_number and LC).
- Each vehicle provides its position, velocity, acceleration and input signal as an output (outputs U_i, A_i, V_i, Q_i)
- Additional outputs for test purposes are the computed feedforward signal (FF) and the current lane (LC)

The basic Simulink block for the required functionality is shown in Fig.



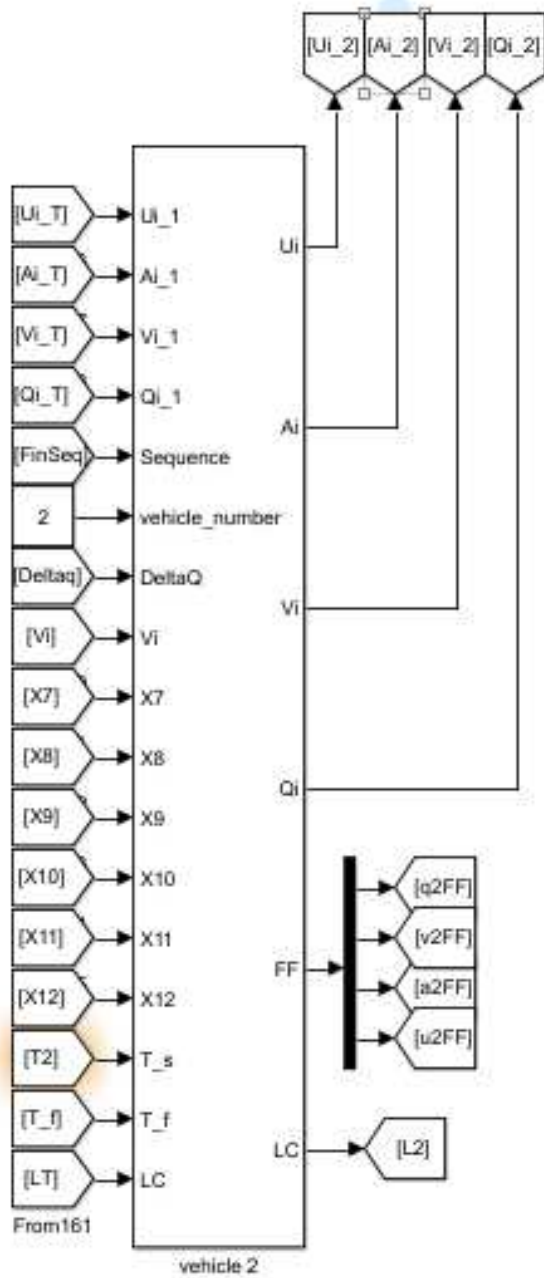


Figure 36: Vehicle S-Function Block.

4.3 Vehicle S-function Implementation

Inputs

- Communicated from RSU: Input signal, Acceleration signal, Velocity signal, Position signal, Sequence number, Desired Distance gap, Initial velocity, Start time and Final time.
- Vehicle number.

Outputs

- Communicated to RSU: Input signal, acceleration signal, velocity signal, position signal
- For observation: feedforward signals.

Parameters

- Initial position, Initial velocity, length of vehicle, headway time, distance at standstill constant and some controller constants

4.3.1 Input implementation

We used many inputs, parameters and DWork parameters in the Vehicle S-function of our thesis. There are seventeen inputs, nine parameters and 21 allocated DWork parameters.

The inputs for the acceleration, velocity, position, input signal and vehicle sequence have different dimensions depending on the number of vehicles in the simulation. For example, if there are 7 vehicles, we use `ssSetInputPortWidth(S, 1, 7);` for the acceleration input signal vector. Here, 1 is the index number of the input and 7 is the dimension of the signal vector. The vehicle number is a scalar that is assigned by `ssSetInputPortWidth(S, 5, 1);`. The desired feedforward signal is specified by the inputs "DeltaQ", "Vi" and "T_s". For example, for DeltaQ, the input is allocated as `ssSetInputPortWidth(S, 6, 1);`. 6 is the index number of the input. The grid values of the optimal control approximation are provided as the input values X7 to X12 and T.f. They are allocated in the form `ssSetInputPortMatrixDimensions(S, 8, 21, 11);`. For example, for X7, 8 is the index number of the input. 21 and 11 define the dimension of the parameter matrix. In addition, there are some supplementary functions that specify properties of

the inputs. `ssSetInputPortRequiredContiguous(S, 0, true);` specifies that the input signals are contiguous. `ssSetInputPortDirectFeedThrough(S, 0, 0);` defines that the respective input is not a direct feed through port.

There are 9 parameters such as initial velocity and position of the vehicle in the vehicle S-function. For illustration, consider `ssSetSFcnParamTunable(S,0,false);`, which allocates the parameter with index number 0. Here, "false" indicates that the parameter is not tunable that is, the parameter can not change during simulation.

In addition, our vehicle model uses 21 Dwork memory blocks in the vehicle S-function to store internal data. For example, one Dwork memory is allocated for the input signal U_{i-1} as

```
ssSetDWorkWidth(S, 0, 1);
ssSetDWorkDataType(S, 0, SS-DOUBLE);
ssSetDWorkName(S, 0, "Ui-1");
```

0 is the index number of Dwork and 1 is the width of the Dwork block. SS-DOUBLE is the data type of the Dwork and "Ui-1" is the name of the Dwork block. There are 21 such Dwork block in the vehicle S-function.

4.3.2 Output implementation

There are 6 outputs in the vehicle S-function implementation. For example, the velocity output is configured as `ssSetOutputPortWidth(S, 2, 1);` 2 is the index number of the output and 1 is the defined width of the output. There are 5 outputs like this and there is 1 output with width 4.

4.3.3 Update Function Implementation

In the update function, we define the required and updated output values, parameters, Dwork memory blocks and external inputs. These values are computed according to Section 3.7 and include the coefficients for the optimal control approximation. In order to evaluate the approximation of the optimal control trajectory, a linear interpolation on a grid with 4 points is needed. These points are arranged around the desired point as can be seen in the following table. Here, P is desired value and Q_{11} , Q_{12} , Q_{21} and Q_{22} are the values at the neighbor points. The respective coordinates are x, y and $x_1 < x < x_2$ and $y_1 < y < y_2$ are the grid points.

Neighbor Points			
	x_1	x	x_2
y_1	Q_{11}		Q_{21}
y		P	
y_2	Q_{12}		Q_{22}

In order to determine P from the neighbor points, we use the bilinear interpolation equation below.

$$\begin{aligned}
P \approx & (Q_{11}) \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} + (Q_{21}) \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} \\
& + (Q_{12}) \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} + (Q_{22}) \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)}.
\end{aligned} \tag{4.1}$$

This equation is evaluated for all relevant coefficients in Section 3.6. These coefficients are then used to determine the feedforward position, velocity, acceleration and input signal for opening and closing gaps.

4.3.4 Derivative Implementation

In this callback function, the state space model of the vehicle with CACC is realized according to (3.1). Each state derivative is allocated in the form `real-T *dx = ssGetdX(S)`; In our vehicle S-function, we use 4 state derivative values for the vehicle model and one state derivative for the simulation time.

4.3.5 Example Simulations

In this section, we show simulation examples with 3 Vehicles by using the vehicle S-function.

Firstly, we opened a desired distance of 29 m between the first and second vehicle using the computed feedforward signal. The feedforward signal is computed by the vehicle S-function instance of vehicle 2 such that vehicle 2 opens a gap to the first vehicle. Its follower vehicle 3 slows down together with vehicle 2 due to the usage of CACC. The maneuver can be inspected in Figure 37.

Secondly, we opened a desired distance of 29 m between vehicle 2 and 3. The resulting response can be seen in Figure 38.

Finally, we close a gap of 29 m between vehicle 2 and 3. The resulting position plot is shown in Figure 39.

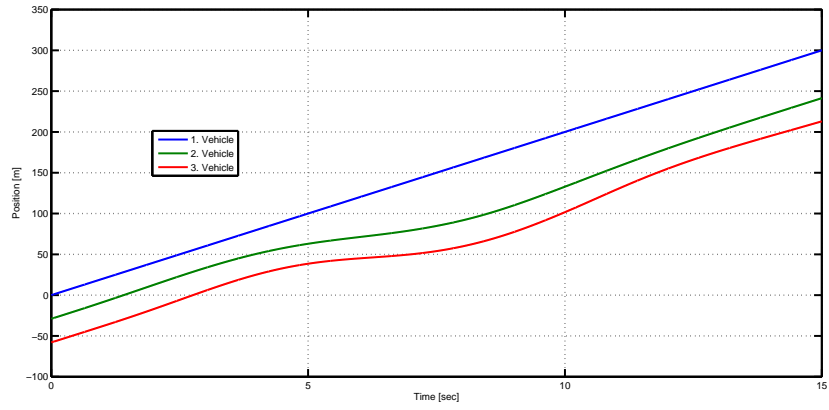


Figure 37: Position of vehicles when applying a gap opening maneuver to vehicle 2.

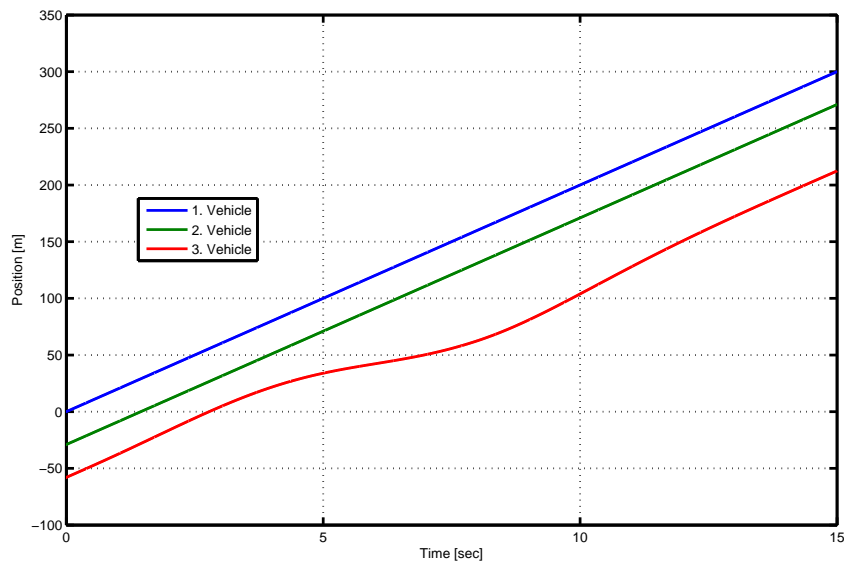


Figure 38: Position of vehicles when vehicle 3 opens a gap.

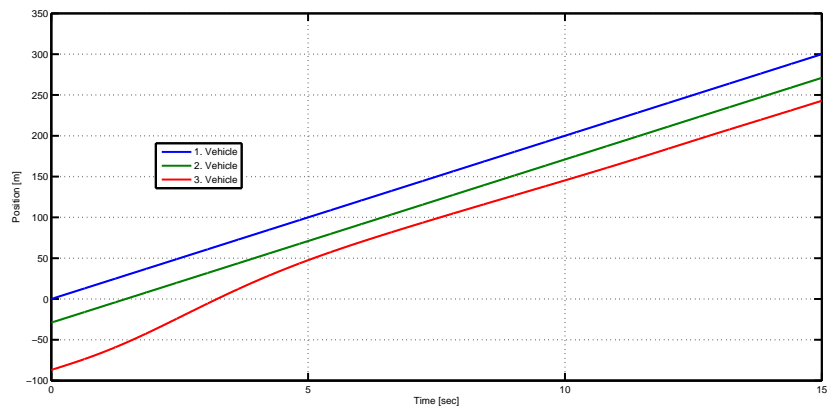


Figure 39: Position response when vehicle 3 closes a gap.

CHAPTER 5

CONCLUSION

The subject of the thesis is the realization of gap opening and closing maneuvers in vehicle strings. To this end, the control architecture for cooperative adaptive cruise control (CACC) is extended by a feedforward signal.

Using this architecture, five methods for the computation and representation of open/close gap trajectories for vehicles in vehicle strings are proposed. The first method is based on the solution of an optimal control problem, the second method uses a polynomial trajectory and plant inversion, the third method concatenates three polynomials and uses nonlinear programming to determine the polynomial coefficients, the fourth method uses a high-order polynomial and the fifth method uses concatenated polynomials in order to approximate the optimal control solution.

Hereby, the optimal control solution represents the desired solution which performs the respective maneuver in the shortest possible time while meeting all constraints. However, the computation times for finding the optimal control solution are not suitable for a real-time implementation as would be required in practice. The polynomial trajectory obtained by plant inversion can be computed very quickly. Nevertheless, a longer maneuver duration needs to be selected and the acceleration constraints are usually violated. The method of concatenating three polynomials allows meeting the acceleration constraints and its computation time is shorter than the optimal control computation. Nevertheless, a longer maneuver duration has to be accepted. In order to further reduce the computation time, an interpolation method has been developed. It computes suitable trajectories for a grid of parameter values offline. Then, trajectories for any parameter combination that is not on the grid can be obtained by linear interpolation. It is shown that this method leads to very good results. In order to further reduce the maneuver duration, the same interpolation method is applied to polynomial approximations of the optimal control solution itself. As a result, fast trajectories are obtained. Slight violations of the acceleration constraints are possible if the optimal control solution is approximated by a single polynomial. In contrast, an approximation by five concatenated polynomials proves very suitable since it directly captures the shape of open/close gap trajectories. In addition, the vehicle model with CACC and feedfor-

ward computation is implemented in the form of an S-function for fast simulation in Simulink. Simulation examples confirm the suitability of the computed trajectories.



REFERENCES

- [1] **Mashrur A. Chowdhury,; Adel Wadid Sadek, (2003),** "*Fundamentals of Intelligent Transportation Systems Planning*" 2003, ARTECH HOUSE, INC. P.P. 1–7/P.P. 35–55.
- [2] **Robayet Nasim,; Andreas Kessler, (2012),** "*Distributed Architectures for Intelligent Transport Systems: A Survey*" 2012 IEEE Second Symposium on Network Cloud Computing and Application.
- [3] **Petros A. Ioannou (ed), (1997),** "*Automated Highway Systems*", Springer-Verlag US.
- [4] **Astrid Linder, Albert Kircher, Anna Vadeby, Sara Nygardhs, (2007),** "*Intelligent Transport Systems (ITS) in Passenger Cars and Methods for Assessment of Traffic Safety Impact*", VTI, report 604A.
- [5] **Bin Ran, Peter J. Jin, David Boyce, Tony Z. Qiu, Yang Cheng, (2013),** "*Cooperative Adaptive Cruise Control, Design and Experiments Transportation Research: Impact of Intelligent Transportation System Technologies on Next-Generation Transportation Modeling*", Taylor and Francis Group, LLC.
- [6] **P. Fernandes and U. Nunes, (2012),** "*Platooning With IVC-Enabled Autonomous Vehicles: Strategies to Mitigate Communication Delays, Improve Safety and Traffic Flow*", Intelligent Transportation Systems, IEEE Transactions on , Vol.13, No.1, P.P. 91- 106
- [7] **R. Rjamani and S. E. Shladover, (2001),** "*An experimental comparative study of autonomous and co-operative vehicle-follower control systems*", Transportation Research Part C, Vol. 9, P.P. 15–31.
- [8] **Jeroen Ploeg, Dipan P. Shukla, Nathan van de Wouw, and Henk Nijmeijer, (2014),** "*Controller Synthesis for String Stability of Vehicle Platoons*", IEEE Transactions on Intelligent Transportation Systems, Vol. 15, No. 2.

- [9] **Ivan Mallocci, Zhiyun, Gangfeng Yan, (2012)** ”Stability of interconnected impulsive switched systems subject to state dimension variation”, *Nonlinear Analysis: Hybrid Systems*, 6, P.P. 960-971.
- [10] **Inas Deaibil, (2015)**, *Platooning of Vehicles in Intelligent Transportation Systems*, Master Thesis, Department of Electronic and Communication Engineering, Cankaya University.
- [11] **Deaibil, I. B. A., Schmidt, K.W. (2015)**: Performing Safe and Efficient Lane Changes in Dense Vehicle Traffic, *Engineering and Technology Symposium*, Ankara, Turkey.
- [12] **Christopher Nowakowski, Steven E. Shladover, Delphine Cody, Fanping Bu, Jessica O’Connell, John Spring, Susan Dickey, David Nelson, (2011)**, ”Co-operative Adaptive Cruise Control: Testing Drivers Choices of Following Distances”, California PATH Research Report, UCB-ITS-PRR-2011-01.
- [13] **Gerrit J. L. Naus, René P. A. Vugts, Jeroen Ploeg, Marinus (René) J. G. van de Molengraft, and Maarten Steinbuch, (2010)**, ”String-Stable CACC Design and Experimental Validation: A Frequency-Domain Approach”, *IEEE Transactions on Vehicular Technology*, Vol. 59, No. 9.
- [14] **P. E. Rutquist and M. M. Edvall, (2010)**, *PROPT – Matlab optimal control software*, Available: [http://tomopt.com/docs/TOMLAB PROPT.pdf](http://tomopt.com/docs/TOMLAB_PROPT.pdf)
- [15] **J. Ploeg, N. van de Wouw, and H. Nijmeijer, (2014)** ”Lp string stability of cascaded systems: Application to vehicle platooning,” *Control Systems Technology*, *IEEE Transactions on*, vol. 22, no. 2, pp. 786-793.
- [16] **Z. Wu and Y. Liu and G. Pan, (2009)** ”A Smart Car Control Model for Brake Comfort Based on Car Following”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1., pp. 42–46.
- [17] **E. Dovgan and T. Tušar and M. Javorski and B. Filipič, (2012)** ”Discovering Comfortable Driving Strategies using Simulation-based Multiobjective Optimization”, *Informatica*, vol. 36, no. 3, pp. 319–326.
- [18] **Arup K. Nandi and Debasri Chakraborty and Warren Vaz, (2015)** ”Design of a comfortable optimal driving strategy for electric vehicles using multi-objective optimization”, *Journal of Power Sources*, vol. 283, pp. 1–18.

- [19] **Robayet Nasim, Andreas Kassler, (2012),** *"Distributed Architectures for Intelligent Transport Systems: A Survey"*, 2012 IEEE Second Symposium on Network Cloud Computing and Application.
- [20] **Bin Ran, Peter J. Jin, David Boyce, Tony Z. Qiu, Yang Cheng, (2013),** *"Cooperative Adaptive Cruise Control, Design and Experiments Transportation Research: Impact of Intelligent Transportation System Technologies on Next-Generation Transportation Modeling"*, Taylor and Francis Group, LLC.



CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Çankaya , Erkam

Nationality: Turkey

Date and Place of Birth: 03 April 1991 , Ankara

Marital Status: Single

Phone: +90 543 382 69 77

Email: erkamcankaya1@gmail.com

EDUCATION

Degree	Institution	Year
MS	Çankaya University, Electronic and Communication Eng.	2017
B.Sc.	Çankaya University, Electronics and Communication Engineering.	2014
High School	Ankara High School	2009

FOREIGN LANGUAGES

Turkish, English

HOBBIES

Tennis, Traveling, Video games