Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Control Engineering

DIPLOMA THESIS

Distributed control of platoons of racing slot cars

Prague, 2012

Author: Dan Martinec

Declaration

I hereby confirm that I wrote this diploma thesis on my own and that I listed all the used materials in the references.

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

Prague, 11th May 2012

Dan Martinec

Acknowledgement

I would like to thank my supervisor Ing. Zdeněk Hurák Ph.D. for exemplary supervision, invaluable remarks and obliging cooperation. Next I thank the whole AA4CC group for many advices, recommendations and friendly amenable atmosphere. Once again I thank Ing. Milan Brejl from Freescale Semiconductor company and Ing. Vít Hlinovský CSc. from Department of Electric Drives and Traction FEE CTU for generous donation of several PCBs.

My last but deepest thanks goes to my parents and brothers for their endless stream of love, patience and support during the whole studies. I will always be grateful to God for such a wonderful family.

We cannot change the cards we are dealt, just how we play the hand.

Randy Pausch

Poděkování

Tímto bych chtěl poděkovat vedoucímu diplomové práce Ing. Zdeňku Hurákovi Ph.D. za příkladné vedení práce, neocenitelné připomínky a ochotnou spolupráci. Dále celé skupině AA4CC za mnohé rady, doporučení a přátelskou atmosféru při práci. Tímto bych chtěl ještě jednou poděkovat Ing. Milanu Brejlovi ze společnosti Freescale Semiconductor a Ing. Vítu Hlinovskému CSc. z Katedry elektrických pohonů a trakce FEL ČVUT za štědrý dar několika osazených desek plošných spojů.

V neposlední řadě hluboce děkuji svým rodičům a bratrům za jejich bezmeznou lásku, trpělivost a podporu po celé studium. Vždy budu Bohu vděčný za tak skvělou rodinu.

Nezměníme karty, které nám byly dány. Můžeme se s nimi jen naučit hrát.

Randy Pausch

Abstract

This thesis examines several known control algorithms for a platoon of ten slot car vehicles, such as a predecessor-following concept, leader-following concept, combination of these two concepts and a bidirectional algorithm. Pros and cons of these algorithms are analytically and experimentally assessed, in particular the so-called *string stability*, and compared with the LQR controllers. An insight into known mathematical models of the platoon is given by using three formalisms: state space, decentralized state space and 2-D polynomial approach. The experiments are carried out on a platform developed for these purposes. The platform is based on commercially available racing slot cars made by Carrera company equipped with an onboard 32-bit microcontroller-based control system. The assembled PCB board was provided by the Freescale Semiconductor, Rožnov pod Radhoštěm, Czech Republic, within their Freescale Race Challenge 2012. A few additional components such as an infrared range sensor and a wireless communication module have been added to the original system. Technical parameters of the platform and mathematical model of the slot cars are described in the thesis.

Abstrakt

Tato práce zkoumá známé algoritmy pro řízení kolon na deseti autodráhových autíčkách, jako jsou řízení k předchůdci, řízení k vůdci, kombinace obou těchto přístupů a také obousměrné řízení. Dále analyticky a experimentálně vyšetřuje klady a zápory těchto algoritmů, vlastnost zvanou *stringová stabilita* a porovnává jejich výkon s LQR řízením. V práci je uveden přehled možných matematický modelů řízení kolony pomocí tří formalismů: stavový popis, decentralizovaný stavový popis a 2-D polynomiální přístup. Experimenty jsou provedeny na platformě, která k tomuto účelu byla v rámci práce sestrojena. Tento přípravek je možné umístit uvnitř místnosti a je tvořen ekonomicky dostupnými autodráhovými autíčky od výrobce Carrera. Autíčka mají 32-bitový mikropočítač umístěný na desce plošných spojů. Ta byla poskytnuta společností Freescale Semiconductor, Rožnov pod Radhoštěm, Česká republika, v rámci soutěže Freescale Race Challenge 2012. Autíčka byla následně dovybavena dalšími součástkami: infračerveným vzdálenostním senzorem a bezdrátovým komunikačním modulem. Technické parametry této experimentální platformy a matematický model použitých aut jsou také součástí diplomové práce.

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Dan Martinec**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Distribuované řízení kolony závodních autodráhových autíček**

Pokyny pro vypracování:

1. Navrhněte a implementujte distribuovaný řídicí systém pro kolonu většího počtu (cca 10) autonomních autíček pro závodní autodráhu.
2. Řídicí systém musí usměrňovat jízdu celé kolony konstantní rychlostí a při zadaných rozestupech, stejně jako změnu této společné rychlosti a/nebo rozestupů.
3. Pomocí simulací i experimentů demonstrujte základní výsledky z oblasti distribuovaného řízení kolon vozidel, jako jsou nemožnost stabilizace kolony pomocí relativních měření.
4. Kromě v literatuře běžně popisovaného jednosměrného řízení prozkoumejte i možnosti obousměrného řízení (řídicí systém daného vozidla zpracovává změřenou vzdálenost nejen k předchůdci ale i k následovníkovi).

Seznam odborné literatury:

[1] M. R. Jovanovic and B. Bamieh, On the ill-posedness of certain vehicular platoon control problems, Automatic Control, IEEE Transactions on, vol. 50, no. 9, pp. 1307-1321, 2005.
[2] I. Lestas and G. Vinnicombe, Scalable Decentralized Robust Stability Certificates for Networks of Interconnected Heterogeneous Dynamical Systems, Automatic Control, IEEE Transactions on, vol. 51, no. 10, pp. 1613-1625, 2006.
[3] P. Seiler, A. Pant, and K. Hedrick, Disturbance propagation in vehicle strings, Automatic Control, IEEE Transactions on, vol. 49, no. 10, pp. 1835-1842, 2004.
[4] M. Šebek and Z. Hurák, 2-D polynomial approach to control of leader following vehicular platoons, presented at the 18th World Congress of the International Federation of Automatic Control (IFAC), Milano, Italy, 2011.

Vedoucí: Ing. Zdeněk Hurák, Ph.D.

Platnost zadání: do konce letního semestru 2012/2013

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
L.S.
-4-

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 2. 12. 2011

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Control Engineering

DIPLOMA THESIS ASSIGNMENT

Student: **Bc. Dan Martinec**

Study program: Cybernetics and Robotics
Branch of study: Systems and Control

Title of the thesis: **Distributed control of platoons of racing slot cars**

Instructions:

1. Design and implement a distributed control system for a platoon of a larger number (cca 10) of autonomous racing slot cars.
2. The control system guarantees constant cruising speed of the whole platoon while simultaneously keeping prespecified intervehicular distances. In addition, both the travelling speed and the distances should be subject to change.
3. Using both simulations and experiments, demonstrate certain inherent features and limitations of vehicular platoons such as string instability while using relative measurements only.
4. Besides the unidirectional control, explore also the bidirectional control, in which the onboard control systems processes not only the distance to the precedessor but also to the follower.

Literature:

[1] M. R. Jovanovic and B. Bamieh, On the ill-posedness of certain vehicular platoon control problems, Automatic Control, IEEE Transactions on, vol. 50, no. 9, pp. 1307-1321, 2005.
[2] I. Lestas and G. Vinnicombe, Scalable Decentralized Robust Stability Certificates for Networks of Interconnected Heterogeneous Dynamical Systems, Automatic Control, IEEE Transactions on, vol. 51, no. 10, pp. 1613-1625, 2006.
[3] P. Seiler, A. Pant, and K. Hedrick, Disturbance propagation in vehicle strings, Automatic Control, IEEE Transactions on, vol. 49, no. 10, pp. 1835-1842, 2004.
[4] M. Šebek and Z. Hurák, 2-D polynomial approach to control of leader following vehicular platoons, presented at the 18th World Congress of the International Federation of Automatic Control (IFAC), Milano, Italy, 2011.

Supervisor:
Ing. Zdeněk Hurák, Ph.D.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The concept of a long platoon (or a string) of independently controlled vehicles has been studied since 1960s. The motivation has apparently been to solve various problems related to traffic control such as increasing traffic capacity, decreasing the driver's stress, while allowing him to pay attention to other duties, reducing the fuel consumption and improving the transportation safety. This field was reexamined in last decade and many papers were published on this topic.

Experiments on real cars is intensively studied in recent years. A highly advanced program experimenting with real car platoons is California PATH program, which manage to drive 8 real cars on a highway already 15 years ago. These cars were driving at 60 mph with only 6.5 meters spacing between each other. Interested readers are referred to [1]. Several other platooning experiments were later conducted in this program.

Another currently-in-progress program experimenting in platooning is European Project SARTRE, which is led by Ricardo UK Ltd and closely cooperates with Volvo company as a result, first functioning real car platoon is demonstrated (see [2]).

Platoons of two cars (leader and follower) is even quite common in these days. Adaptive Cruise Control (ACC) is state of the art cruise controllers and it enables to set safety distance to the preceding vehicle. The system is capable to measure distance to the front vehicle and adjust the speed of the car to keep a desired distance. ACC is currently deployed in most of the modern cars such as Porsche, BMW, Volkswagen, Ford, TWR etc. However, there is no communication between the leader and the follower, more interestingly, the leader might not even know that is leading other car. Therefore

(a) Photography of experiments with 8 cars in California PATH program from [1]

(b) Photography of experiments in Project SARTRE from [2]

Figure 1.1: Real car platoon experiments

the distance between the cars is kept long enough to enable a safe stop of the follower without any risk on crash. This system is certainly a relieve for a driver, though there is no significant reduction of fuel consumption, since the distance is too long to use the effect of the wind tunnel.

Last but not least motivation of this thesis was to extend already gained experience with platooning experiments, see [3].

1.2 Contribution

There are several points of view of how to evaluate the contribution of this thesis.

One outcome of this thesis is a design of prototype platform for various types of experiments on advanced control of the platoon of vehicles. Other experiments investigating the effect of communication delay on the platoon control is simultaneously examined in diploma thesis by Karel Němec. The repetitive learning control experiments are other possible utilization of the platform.

Another outcome is the presentation of the popular control algorithms in the field of vehicular platooning. These algorithms are thoroughly examined, their pros and cons are accessed and ultimately simulated with a mathematical model of the car.

The final contribution is experimental verification of three (from total four) algorithms on ten slot car vehicles on a circular (infinite) track. These algorithms are easy to simulate, yet their practical feasibility is tricky. This introduces several new problems to solve, such as measuring the distance to the leader of the platoon, the delay in communication between vehicles, different mathematical models of the vehicles etc. They are discussed in this thesis.

1.3 Related work

1.3.1 Platooning

The early theoretical treatments of platooning are published in [4] and [5]. The control design problem for a long but finite string of vehicles is solved by applying the standard LQ-optimal control design methodology. Moreover, [6], which introduces the bilateral z-transform in order to extend the LQ setting to an infinite vehicular strings. In addition, [7] deals with the same setting of the bilateral z-transform but with enforced spatial constraints on the distributed controller.

Other noteworthy publication [8] introduces the term *asymptotic stability*, later extended and (re)named to *string stability* by [9]. Considering this new concept of stability, [10] demonstrates that it is impossible to achieve a string stability by measuring only a distance to the vehicle ahead and using a local PID controller. [11] later argues that not only a PID but every linear controller is incapable of string stabilization of a platoon with such a measurement configuration. The last decade has witnessed a new interest in the vehicular platooning problem exploting recent results on spatially distributed systems, such as [12], [13], [14], [15] and [16]. In fact, some of these advanced results provied new insight into the research area which was already considered explored enough. Namely, [17] showes that the problem of state feedback stabilization of the platoon in the spirit of the original papers [4], [5] and [6] is inherently ill-conditioned. In other words, it is not possible to string-stabilize an infinitely long platoon of vehicles. On the other hand, it makes a good sense to aim at reducing the divergence rate. Some improvements are proposed in [18].

Similar experiments with the platooning were conducted with the platform LEGO Mindstorms NXT reported in [19]. Lego Mindstorms turned out to be a well performing platform but it lacks in versatility. Bluetooth communication turned out to be inefficient for our purposes and also absence of the circular track was inconvenient.

1.3.2 Electronics of the vehicle

The main part of the car electronics, the PCB, was provided by Freescale Semiconductor, Rožnov pod Radhoštěm, Czech Republic. They provided a fully assembled PCB board within their Freescale Race Challenge 2012 competition of individual autonomous slot cars organized for student teams in Czech and Slovak republics.

First five cars equipped with XBee module and distance sensor (without velocity sensor) were assembled during the team project with colleagues Karel Němec, Jaroslav Marek and Jan Auersvald.

1.4 Content

Chapter 2 introduces the electronic equipment of the car. It discusses how velocity, distance and current are measured. It also briefly shows how the car was constructed and gives an insight into a wireless communication between cars.

Chapter 3 begins with discussion of how the model of an individual car was developed and identified, and how the controlling algorithm for the car was designed. The second part of this chapter begins with the core of the thesis by showing possible mathematical models of the platoon.

Chapter 4 thoroughly discusses several controlling algorithms for the platoon as well as the term of string stability. It also shows an LQR controllers as an possible alternative.

Chapter 5 documents the experiments with various types of the controlling algorithms. It graphically shows characteristic phenomena present in the platoon control.

Chapter 6 summarizes the results of the thesis. It gives the final comparison of all implemented controllers.

To keep the fluency of the thesis several appendices are included dealing with the $\mathcal{LZ}_1$ transform properties, XBee wireless packets and the complete electronic scheme of the PCB.

Chapter 2

Design modification of electronics

2.1 Original design

2.1.1 Overview

The PCB was designed by Freescale Semiconductor, Rožnov pod Radhoštěm, and is offered for free to the student teams participating in the Freescale Race Challenge competition where single autonomous cars compete in driving ten laps as fast as possible with the actual shape of the track unknown before the competition.

The basic block diagram is shown in Fig.2.1. A detailed documentation of the PCB is available at [20] and more detailed schematic is given in Appendix B. The PCB was used as a platform for additional electronic components such as a wireless communication module, a distance measurement sensor and a velocity measurement sensor, see Fig.2.14.

The PCB is populated with 64-pin 32-bit MCF51JM64 microcontroller from *ColdFire* series, running at 3.3 V and 48 MHz with 64 KB Flash memory and 16 KB RAM. Another key component is MC33931 H-bridge operating at 13.6 V and 8 kHz. Additionally, two voltage regulators 7805 and LF33CV convert the rectified track voltage 13.6 V to 5 V and to 3.3 V, respectively. These regulators are not parts of the original PCB. 3M Card Connector provides data logging to a microSD card. This is for the basic equipment as provided by Freescale Semiconductor. The extra hardware added to measure velocity and distance, and to enable wireless communication is described in the dedicated sections below.

The microcontroller's own 12-bit A/D converters operates at 3.3 V. Voltage (in Volts)

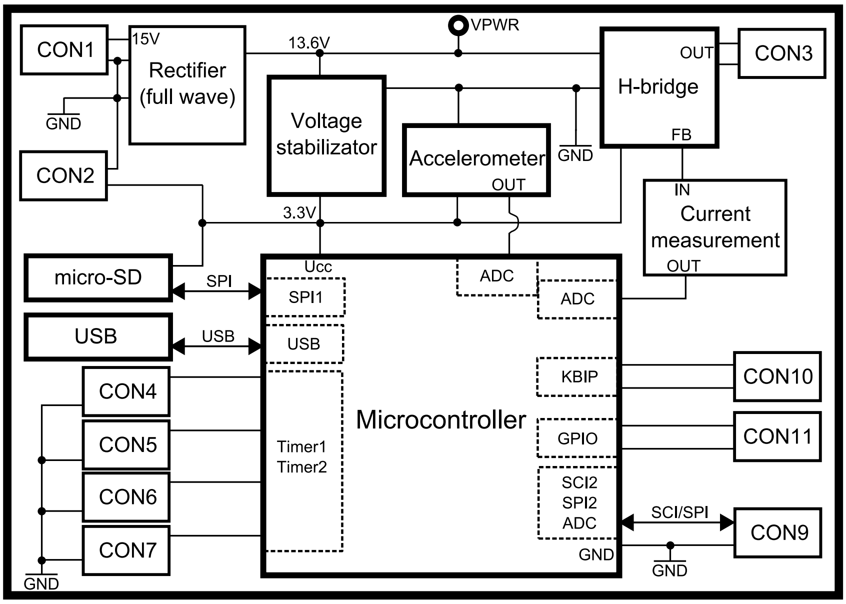


Figure 2.1: Block scheme of the provided PCB.

measured by the A/D converters is calculated by

$$U = \frac{3.3 \cdot N}{2^{12}}, \tag{2.1}$$

where N is an integer number returned from the A/D converter. A single-polarity voltage can only be measured.

Note that the accelerometer populated on PCB, is not used in our experiments.

2.1.2 H-bridge and motor current

PWM generation

The motor is powered by the PWM signal from the H-bridge. The H-bridge is operated from the microcontroller using Timer2 switched to PWM mode. The applied voltage on the motor is controlled by the PWM duty cycle, which is set in the program with resolution of $\frac{1}{60}$%. However, the actual PWM value on the motor is always 6.4% smaller then, the duty cycle set in the program.

Motor current measurement

The PCB also provides measuring of the motor current. This is done by H-bridge, which provides a portion of the motor load current on the so called feedback pin (FB pin). This is measured with the frequency of 8 kHz. The pin is connected to the analog low-pass RC filter with bandwidth $f_{bw} = 590\,\mathrm{Hz}$ which filters disturbances caused by PWM frequency. The resistor with $R_{IR} = 270\Omega$ then converts the current to the voltage U_{IR} measured by the A/D converter.

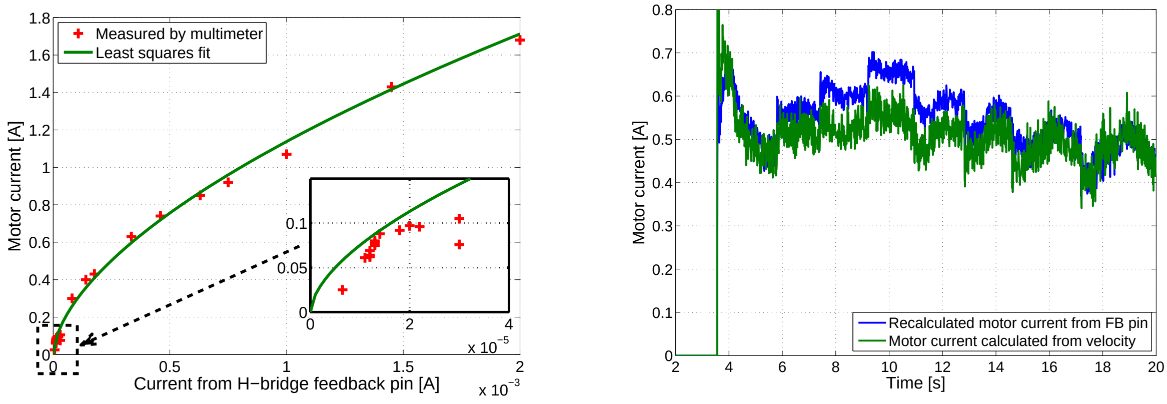
Unfortunately, a portion of current provided on the FB pin is nonlinear with respect to the motor current and data given in datasheet are not accurate enough to identify a nonlinear behavior. Therefore, non-linearity was found out from the following experiment. Various values of voltages were applied on the motor and wheels were braked by hand to generate various values of current. Simultaneously, the current in the motor were measured by a multimeter and compared with the current on the FB pin. A function $f(x) = a \cdot x^b$ was chosen to model non-linearity and *Curve Fitting Toolbox* from *Matlab* was employed to find a and b parameters with the result

$$I_M = 67 \cdot (I_{FB})^{0.59}, \tag{2.2}$$

where I_M stands for actual motor current and I_{FB} is current on the feedback pin. This experiment is depicted in Fig.2.2a, whereas Fig.2.2b shows the comparison of the current calculated by function (2.2) and the motor current calculated by equation (more detail is given in the next chapter)

$$I_M = \frac{U_{in} - k_e \cdot \omega}{R}, \tag{2.3}$$

where U_{in} is voltage applied on the motor, k_e is back-emf constant, ω is motor angular velocity and R is resistance constant of the motor. The identification of these constants are thoroughly explained in the next chapter.



(a) Function recalculating current from FB pin to motor current

(b) Comparison of two ways of calculating motor current

Figure 2.2: Current sensing

Though the motor current measurement is relatively accurate, one might ask for a better measurement precision. A good alternative way is to add a resistor in the series with the motor and measure the voltage on this resistor using a differential operational amplifier (see Fig.2.3).

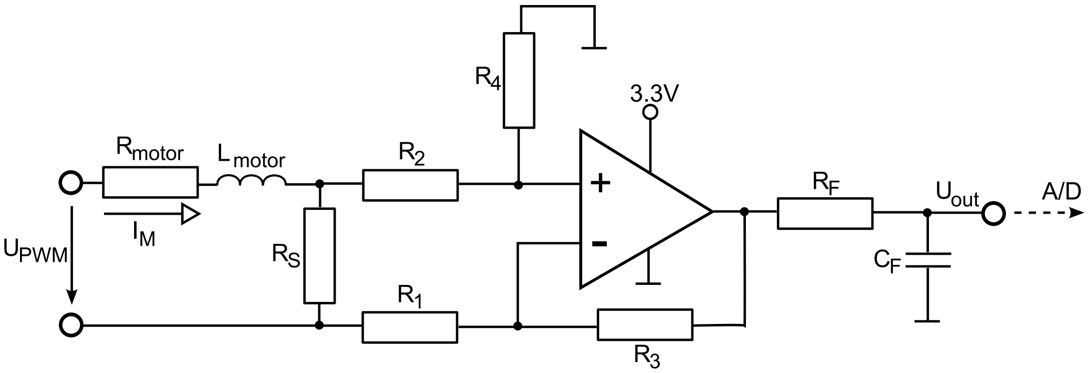


Figure 2.3: Scheme of alternative current measuring circuit

The motor current is then calculated by

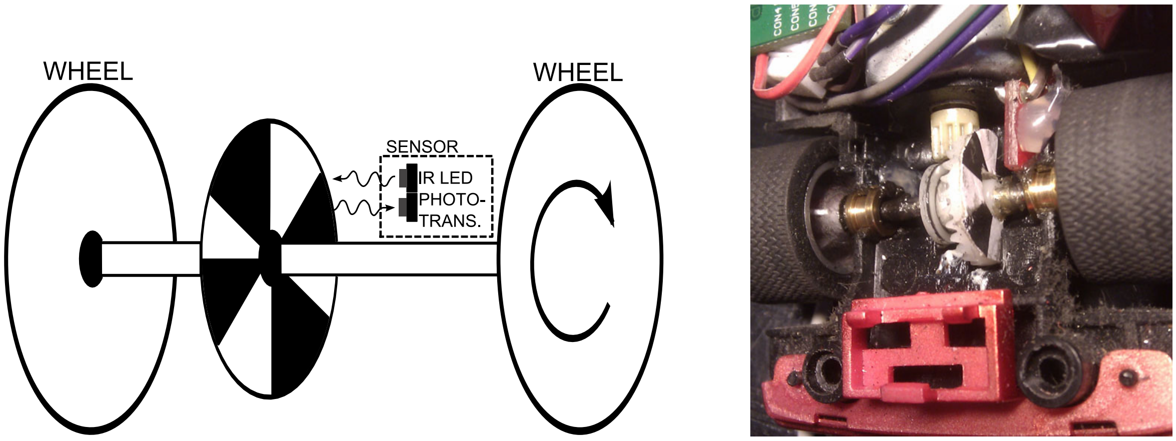
$$I_M = U_{out} \frac{R_1}{R_3 R_S}, \tag{2.4}$$

under condition that $R_1 = R_3$ and $R_2 = R_4$. One has to be careful in choice of resistor R_S since it has to sustain even 2.5 A. The output of the amplifier is connected to a low-pass RC filter to suppress the effect of the PWM signal on the measurement.

2.2 Velocity measurements

2.2.1 Optical encoder on axis

The car velocity is measured by the IR reflectance sensor QRE1113. The sensor is oriented against a small paper with black and white stripes and operates on the same principle as a rotary encoder, see Fig.2.4. The output of the sensor is connected to the so called *Input capture* pin of the processors Timer1 making it possible to detect the paper black-white threshold and to measure the time from the previous threshold detection.



(a) Velocity sensor conceptual scheme

(b) Velocity sensor photography

Figure 2.4: Velocity sensor

It is important to measure time periods both from black to black and from white to white stripes, respectively. Since black and white stripe are detected for a different time period even if the car is traveling at a constant speed (see Fig.2.5).

There are 4 black and 4 white stripes on the paper, the wheel radius is $r = 9.2\,\text{mm}$ and the timer runs at the frequency $f_{\text{input capture}} = 3\,\text{MHz}$. The car velocity is then

$$v[mm/s] = \frac{2\pi r f_{\text{input capture}}}{4 \cdot C_n} \doteq \frac{43.6 \cdot 10^6}{C_n} \tag{2.5}$$

where C_n is the number of pulses in the timer from white-to-white or black-to-black stripe detection.

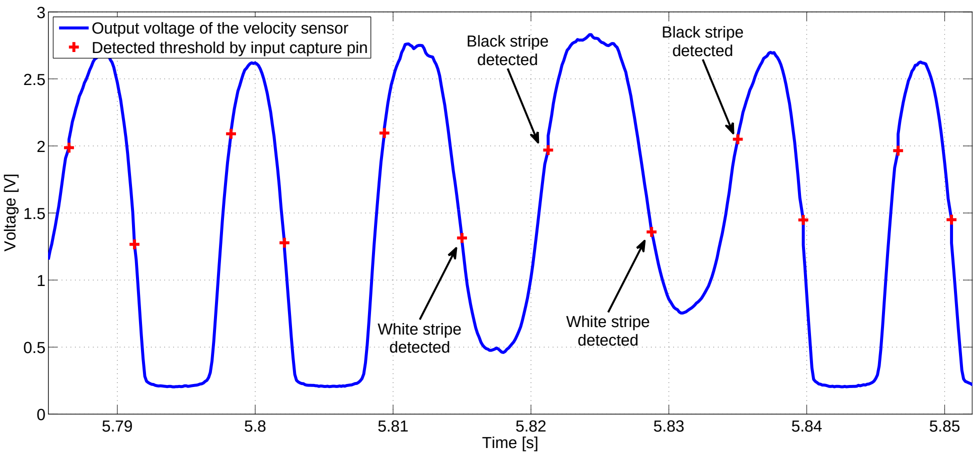


Figure 2.5: Output voltage of the velocity sensor at a constant velocity of the car.

The implemented rate limiter limits the maximum velocity change from the previous measurement to 200 mm · s^{-1}. This precaution minimizes velocity measurement errors. Additionally, the velocity is smoothed with the 8-sampled moving average filter.

A disadvantage of the sensor lies in its sampling dependency on the velocity of the car. The slower the car moves the longer it takes to measure its velocity. Assuming the average car velocity of 800 mm · s^{-1}, the interval between measurements is 9.8 ms.

2.2.2 Estimation from current

An interesting phenomenon appears in the measuring of the motor current. Switching of the rotor poles on the commutator causes oscillations of the current signal as is shown in Fig.2.6. Though these oscillations do not significantly affect cars velocity, they can be utilized to measure the velocity since they contain several periodical signals caused by switching of the commutator.

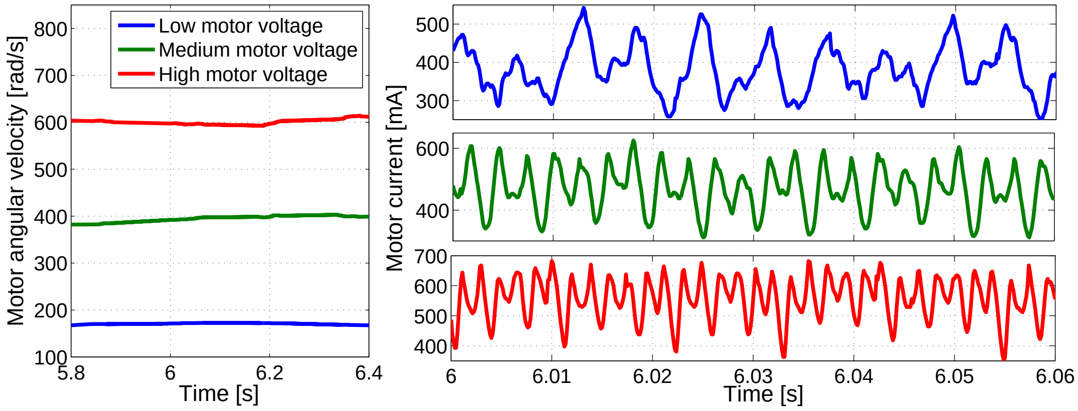


Figure 2.6: Oscillations of the motor current for three different values of applied voltage on the motor

A filtered current with commented detected peaks is shown in Fig.2.7. This additional velocity measurement was successfully implemented. Unfortunately it turned out that it consumes too much processor time consuming and was scratched out.

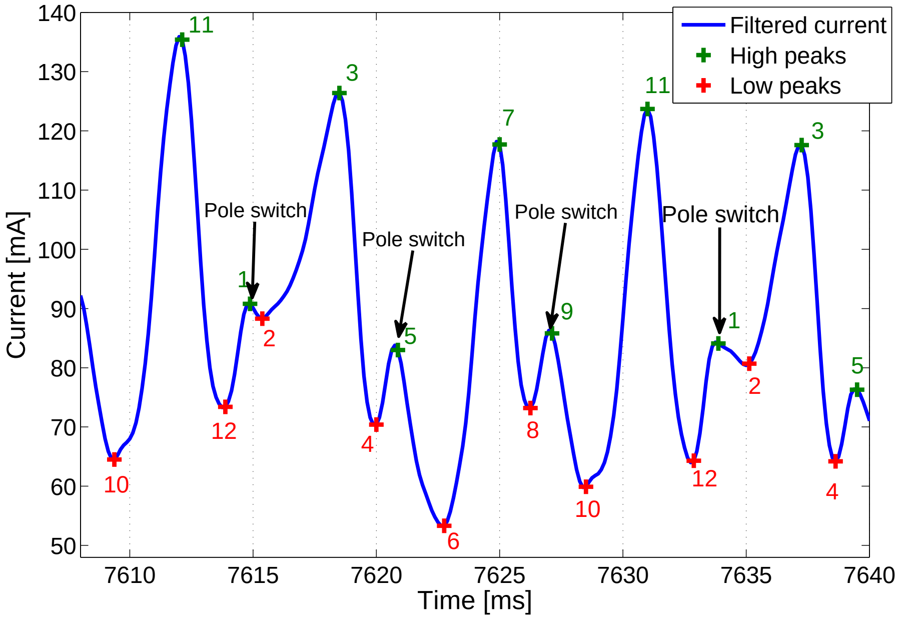


Figure 2.7: Periodical signals in the filtered current signal with detected peaks

2.3 Distance measurements

2.3.1 Infrared sensor

The distance to the vehicle ahead is measured by the analog infrared SHARP GP2D120 (see Fig.2.8) sensor with the range from 5 cm up to 50 cm. The sensor output is connected to the A/D converter of the processor. The measured distance of the sensor is calculated from voltage as

$$distance[mm] = \frac{133000}{\text{voltage[mV]}}. \tag{2.6}$$

Figure 2.8: Analog infrared sensor SHARP GP2D120

Distance is measured with frequency 800 Hz and is filtered with the 8-sample moving average filter.

A disadvantage of the sensor is its ambiguous output. If an object is closer than 5 cm, its output is the same as for much further objects. Another serious drawback lies in limited range of the detection caused by the radius of the circular track. At the distance about 30 cm a car gets out of the line-of-sight of the sensor. This limits the operational distance from 5 cm to 30 cm. Though this limitation is prevailed by an advantage of limitless track length. Both the distance measuring problems can be solved by using the odometry, which is discussed in the next section.

2.3.2 Calculation from odometry

Distance measuring might be improved by using odometry from the velocity measurement. Each car measures its total driven distance on the track and sends this value with the period of 10 ms to the following car. The increment to the last measured distance is calculated as

$$\delta d_{n+1}(k) = (x_n(k) - x_n(k-1)) - (x_{n+1}(k) - x_{n+1}(k-1)) \tag{2.7}$$

This measurement is reliable only for a short time, then the error caused by integration of odometry inaccuracies becomes large and the calibration of the distance from the sensor is necessary to carry out.

2.4 Wireless communication

Wireless module

The communication of a leading car with operator's computer and the communication among the cars is realized by the XBee 802.15.4 OEM RF module from DIGI operating at 3.3 V in the packetized API mode. The structure of the typical XBee packets is shown in Appendix B and a more detailed documentation is available at [21].

A packetized communication has several advantages. It contains the packet length, the source address and the packet identifier (used to determine the purpose of the packet). Moreover, the module retries to send the packet if the receiver does not acknowledge reception.

There are two possibilities of how to communicate between the cars, either using one-to-one or one-to-all (broadcast) communication. The former way was chosen since the experiment is supposed to simulate conditions where the broadcast communication is not possible (e.g. due to large distances between cars). However, in situations when PC sends a command to change a reference distance for all vehicles, the broadcast communication is used to minimize time delay of a new command reception.

Wireless configuration

The module is designed to communicate with the processor via the UART serial interface. A converter such as XBee Explorer USB is required to enable the connection between the wireless module and a computer USB port. This device allows an easy configuration of the XBee module employing the supplemented software X-CTU.

There are a few configuration settings that are to be in place before the first use of the module. Those are a) a module address which has to be unique for all modules, b) the modules Interface Data Rate and the processors SCI Baud rate have to be set to the same value of communication speed, c) API option has to be enabled. Besides of these settings there are several others that keep communication more secure.

More about wireless communication is in diploma thesis of Karel Němec.

2.5 Control of the car

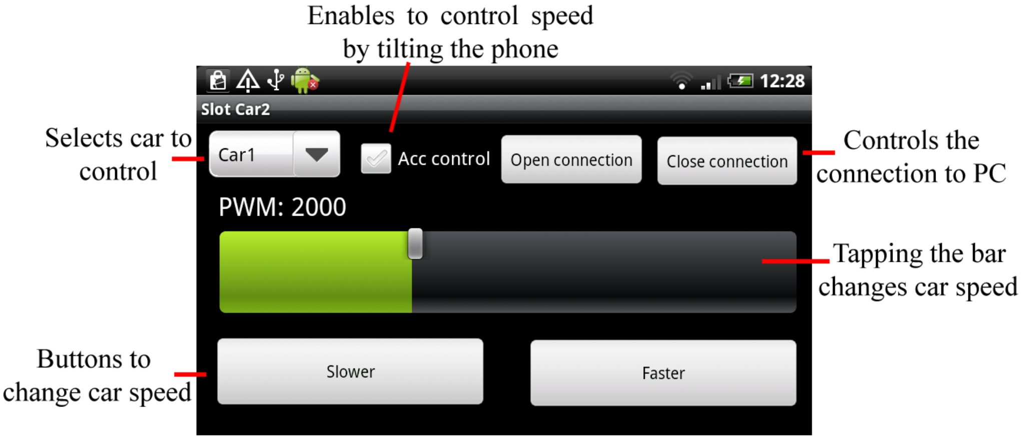
The code for the car's onboard microcontroller is created in the C language. In particular, CodeWarrior Development Studio for Microcontrollers v6.3 from Freescale was used. A template for a project is available at [20]. Uploading a compiled code to the microcontroler is easy through the USB connector. However, debugging of the code is nearly impossible.

Control of the car from PC

Paralelly with the car's program C-sharp application was developed for a wireleslly change of car commands. The application works with the XBee Explorer USB and it serves very well to accelerate program development. The application was used lated in platoon experiments to send reference values for all cars. It can also serve as a limited debugging tool, since it can receive data sent by the car.

Control of the car from Android OS mobile phone

The car is also possible to control from a mobile phone with Android OS. This feature is mainly for demonstrative purposes. Since mobile phones usually do not have implemented XBee module, the phone communicates with the application on a computer via Bluetooth, which re-sends commands to the car. For now, only one directional communication is implemented, mainly the user of the phone sets the PWM duty cycle (see Fig.2.9). However, this functionality can easily be extended if needed.



Enables to control speed by tilting the phone

Selects car to control

Controls the connection to PC

Tapping the bar changes car speed

Buttons to change car speed

PWM: 2000

Figure 2.9: Screenshot of Android application controlling slot car.

2.6 Hardware modifications

The particular type of a vehicle is *Ford Capri RS Tuner 3* shown in Fig.2.10. Since the controller printed circuit board (PCB) provided by Freescale Semicondutor (see below) does out not to fit into the interior of the car (the car dimensions are not readily available before buying), the plastic corpus of the car had to be adjusted (cut). This leaves the car a bit ugly with holes on its sides. Different (larger) cars could be bought, but we wanted to use the available cars already purchased. Alternatively, the layout of the PCB board could also be modified so that it fits into the chosen car, but with several assembled PCB boards available for free from Freescale Semiconductor, it was decided to cut the plastic corpus. When purchasing another type of a car, it may be useful to consider the shape of the front fender for an easy installation of the infrared proximity sensor.

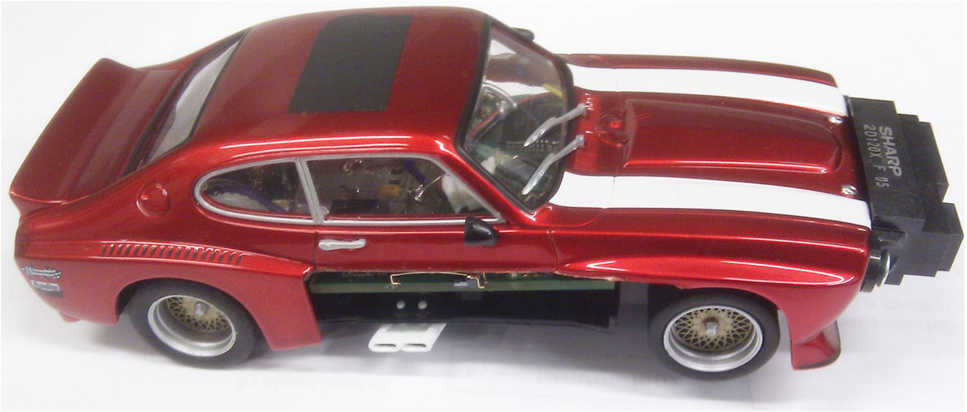


Figure 2.10: Carrera Ford Capri RS Tuner 3 with a hole cut in the side revealing the controller PCB inside. IR proximity sensor attached to the front.

It might be surprising but the assembly of one car took about 4 to 5 hours (see Fig.2.11 and Fig.2.12).

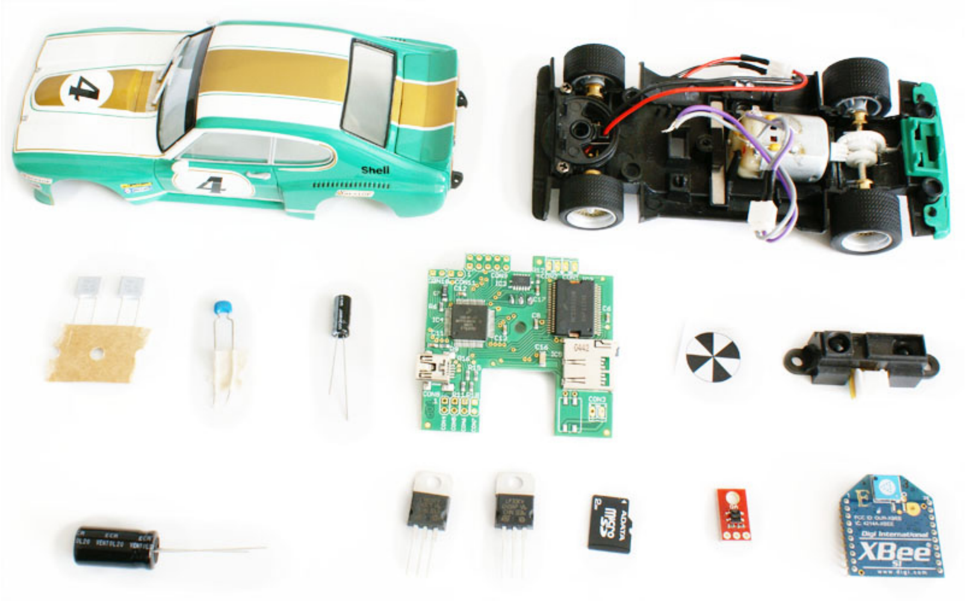


Figure 2.11: Fully disassembled car (of a different color). All car parts except connecting wires are shown.

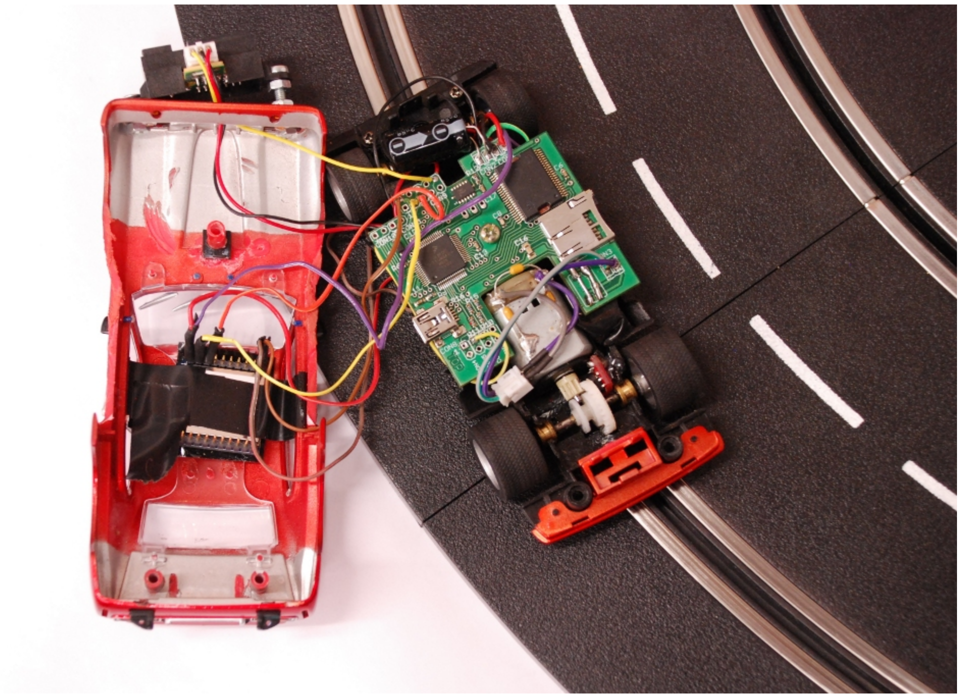


Figure 2.12: Closer look on the assembled PCB board provided by Freescale Semiconductor inserted into a car. XBee module is placed under the roof and the voltage controllers are hidden under the board.

2.7 Overall schematics

A conceptual scheme of measuring devices with all sampling frequencies and filters is shown in Fig.2.13. Design of PID controllers will be explained in the next chapter.

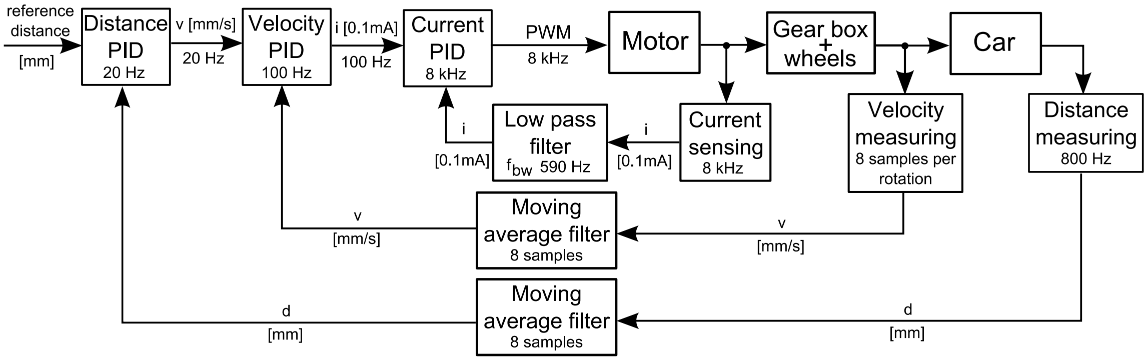


Figure 2.13: Conceptual scheme of measuring devices

The overall electrical schematic of the car is shown in Fig.2.14. All devices except the PCB were chosen, bought and assembled within the diploma project.

The analog output of the velocity sensor is directly connected to the digital input of the processor. It works fine in our case since the signal was of good quality. However, it would be better to add a capacitor to enhance the output signal and the Schmitt trigger to have a full control of the logical threshold boundaries.

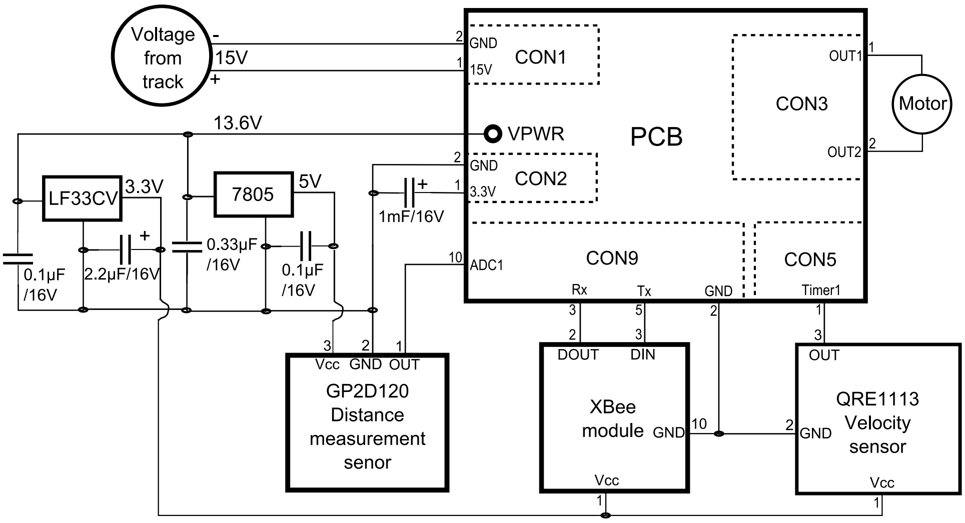


Figure 2.14: Electronic scheme with additional electronic components

Chapter 3

Modeling of the dynamics

3.1 Single car

3.1.1 Model of the car

The dynamics of an individual vehicle is modeled by a moving mass on wheels driven by a DC motor. Since the vehicle moves along a circular track in our experiments, only one-dimensional translational dynamics model is sufficient for modeling its dynamics. The well known equations for the electrical and mechanical parts of the model are

$$L\frac{di(t)}{dt} = -Ri(t) - k_e\omega(t) + u(t), \tag{3.1}$$

$$J\frac{d\omega(t)}{dt} = k_t i(t) - b_d\omega(t) - b_s sign(\omega(t)), \tag{3.2}$$

where $i(t)$ stands for an electric current, $u(t)$ is an electric voltage, $\omega(t)$ is an angular velocity of the motor, R is a resistivity, L is an inductance, J is a moment of inertia, k_e is a back emf constant, k_t is a motor torque constant, b_s is a static friction coefficient and b_d is a dynamic friction coefficient.

By neglecting static friction b_s a single input single output model of the car can be derived

$$G_{u\rightarrow\omega}(s) = \frac{k_t}{LJs^2 + (JR + bL)s + (bR + k_e k_t)}, \tag{3.3}$$

where applied voltage of the motor $u(t)$ and the angular velocity of the motor $\omega(t)$ are input and output of the model, respectively. Tge model for the translational velocity v(t) of the car is then

$$G_{u\to v}(s) = \frac{G_B r k_t}{LJs^2 + (JR + bL)s + (bR + k_e k_t)}, \tag{3.4}$$

where r is a radius of the wheels and G_B represents the gear box ratio from the DC motor to the wheels.

3.1.2 Identification of the constants

Back Electromotive Force k_e and torque k_t constants

A nonconventional way (see [22]) to measure parameter k_t is to attach an arm to the motor, push it on scales and measure the force generated by the motor. Knowing length of the arm l, generated force $m \cdot g$ and current flowing through the motor i, parameter k_t can be calculated by

$$k_t = \frac{l \cdot m \cdot g}{i}, \tag{3.5}$$

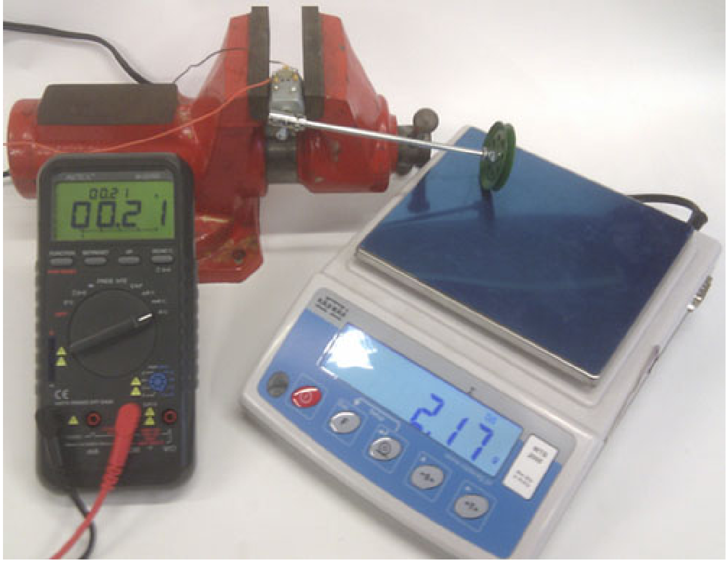
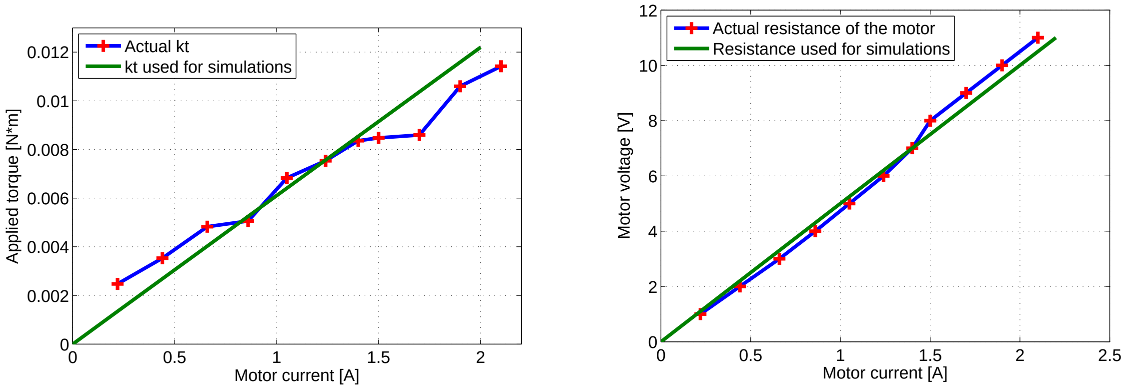


Figure 3.1: Experiment to measure constant k_t. Motor current and generated force are simultaneously measured.

Note, that this type of experiment can also be utilized to measure the electrical resistance of the motor R by assuming that we measure the input voltage. We can do that because the motor is stalled. In this case, no back-emf voltage is induced on the motor and the input voltage equals to the actual voltage on the motor. Data measured in this experiment are shown in Fig.3.2.



(a) k_t constant of the motor

(b) Resistance of the motor

Figure 3.2: Measurement of constants k_t and R

This linear approximation of the measurements yields $k_t = 0.0061\,\mathrm{N} \cdot \mathrm{m} \cdot \mathrm{A}^{-1}$ and $R = 5\,\Omega$.

We do not need to separately measure constants k_e since it is known fact that $k_e = k_t$. However, we verify this fact. To determine constant k_e the following experiment is performed. To minimize friction of the motor we turn the car upside down and apply a voltage on the motor u_{in}. Let us wait for the time when the angular velocity steady and measure current of the motor i (we took the average value of measurements) and the angular velocity of the motor ω. Constant k_e of a DC motor model can be computed by

$$k_e = \frac{u_{in} - R \cdot i}{\omega} = \frac{u_{emf}}{\omega}, \tag{3.6}$$

where R is the electrical resistance of the motor and u_{emf} is the voltage induced from back-emf.

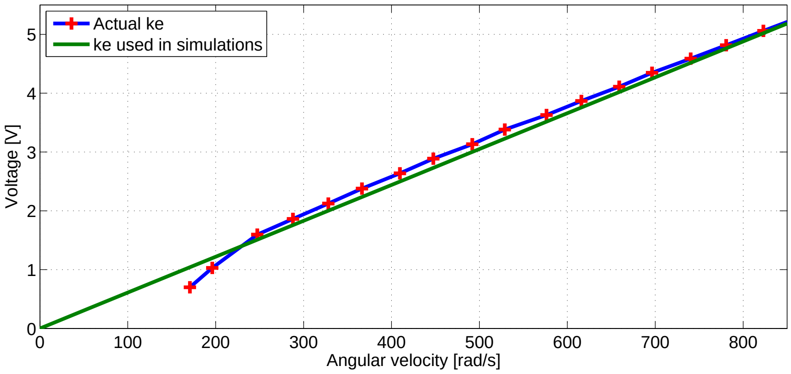


Figure 3.3: Measurement of k_e

By the measurement we obtain $k_e = 0.0061\,\mathrm{V} \cdot \mathrm{rad}^{-1} \cdot \mathrm{s}$, which confirms the fact $k_e = k_t$.

Inductance L and resistance R

The parameters L and R are measured by a special measuring device capable of measuring inductance, resistance and capacitance at various frequencies. By this measurements, we obtain $L = 2\,\mathrm{mH}$ and $R = 5\Omega$. We can see that the measured motor resistance is equal to the value obtained by the previous experiment.

Moment of inertia J and friction b

Moment of inertia was identified as $J = J_{\mathrm{motor}} + J_{\mathrm{car}}$. Since $J_{\mathrm{motor}} << J_{\mathrm{car}}$, can be approximated as $J \approx J_{\mathrm{car}}$. Moreover

$$J_{\mathrm{car}} = mr^2 G_B^2, \tag{3.7}$$

where m is mass of the vehicle.

Observations of the car on the track results in finding model of a friction as a combination of Coulomb (static) and viscous (dynamic) frictions, that is

$$T_f = b_s + b_d \cdot \omega, \tag{3.8}$$

where b_s represents static friction and b_d dynamic friction, respectively.

The identification of these parameters was done by the following experiment. A constant voltage was applied on the car on the track and the velocity and the motor current were recorded after the velocity becomes steady. This was repeated several times for various values of the voltage and for different cars. The torque caused by friction was calculated as

$$T_f = \frac{k_t \cdot i}{\omega} \tag{3.9}$$

The outcome of this experiment is depicted in Fig.3.4.

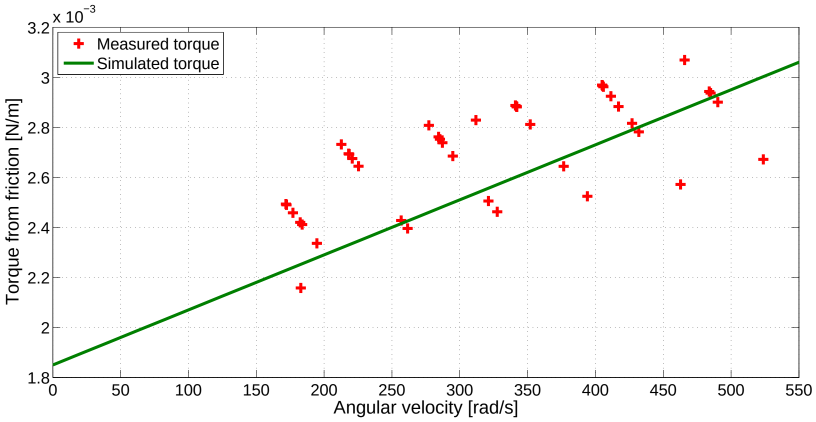


Figure 3.4: Torque caused by friction.

The last parameter needed for the model is the gearbox ratio between the motor and the wheels. For the used cars it is $G_B = \frac{1}{3}$, meaning that the motor is spinning faster than wheels.

All vehicle parameters are summarized in Table.3.1.

Table 3.1: Vehicle parameters obtained by identification.

Parameter	Value in SI units	Parameter	Value in SI units
R	$5\,\Omega$	L	2E-3 H
J	10E-6 $\mathrm{kg\cdot m^2}$	k_e	$0.0061\,\mathrm{V\cdot rad^{-1}\cdot s}$
k_t	$0.0061\,\mathrm{N\cdot m\cdot A^{-1}}$	r	9.2 mm
m	0.115 kg	b_d	$2.2\cdot10^{-6}\,\mathrm{rad\cdot s^{-1}\cdot N\cdot m}$
b_s	$1.85\cdot10^{-3}\,\mathrm{N\cdot m}$	G_B	1/3

Substituting the values of the physical parameters given in Table 3.1 into Eq.3.4 yields

$$G_{car} = \frac{8602}{s^2 + 2509s + 40000}. \tag{3.10}$$

This mathematical model is linear since static friction is neglected. A linear model is convenient for simulations of a very large platoons.

However, for simulations of ten vehicles described in Chapter 4 a nonlinear model described by Eq.3.2 is applied. To check this nonlinear model, several voltage values were applied on the motor. The velocity of the actual car and the velocity calculated from the mathematical model are compared in Fig.3.5. We can see a good agreement between modeled and measured velocities.

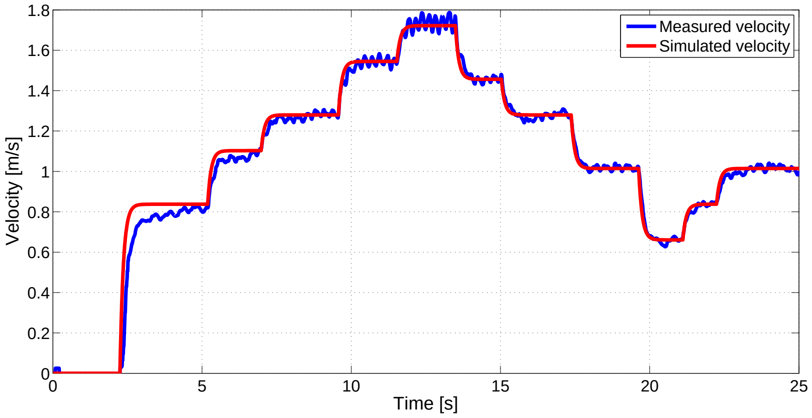


Figure 3.5: Comparison of the measured velocity and the velocity simulated by the model of the car.

3.1.3 Control of the single car

A cascade controller structure, shown in Fig.3.6, was chosen to control a single car. The control consists of three loops: current, velocity and distance, each of which is using a PI controller.

The controlled loops were designed in the following manner. The current controller was designed to have the largest possible bandwidth. The velocity loop was designed to have about ten times narrower bandwidth. The distance loop was designed to have ten times smaller loop than velocity loop.

The three controllers have the following transfer functions

$$\mathrm{PI}_{\text{current}} = \frac{14s + 6400}{s}, \tag{3.11}$$

$$\mathrm{PI}_{\text{velocity}} = \frac{1.5s + 0.5}{s}, \tag{3.12}$$

$$\mathrm{PI}_{\text{distance}} = \frac{2s + 1}{s}, \tag{3.13}$$

where the current, velocity and distance controller have 8 kHz, 100 Hz and 20 Hz sampling frequency, respectively.

The Bode characteristics of all three controlled loops with PI controllers are shown in Fig.3.7.

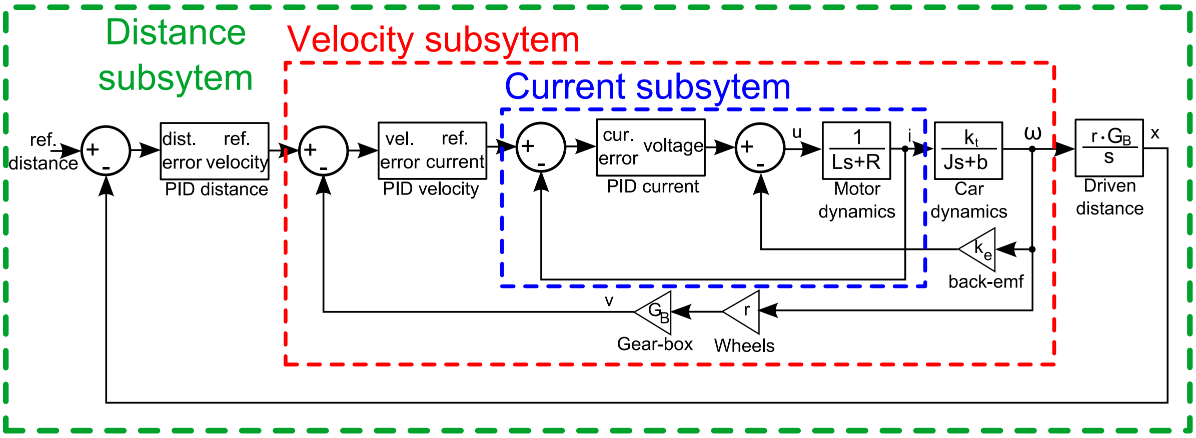


Figure 3.6: Cascade structure of the feedback controllers: current, velocity, position/distance

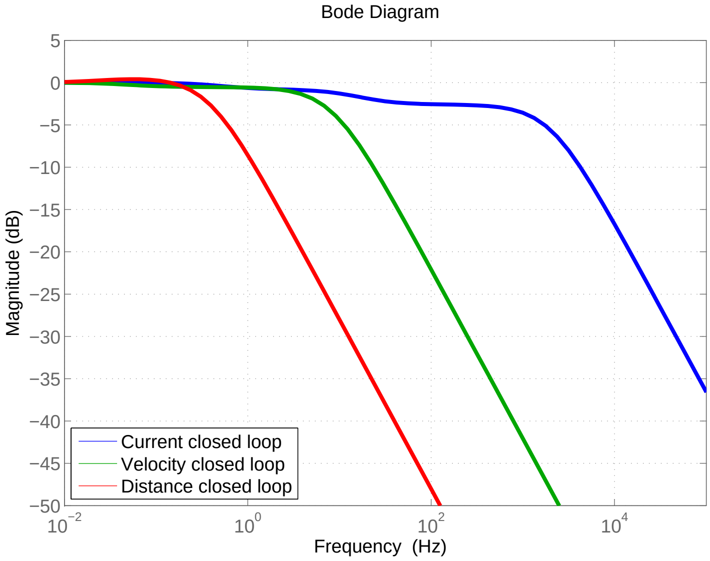


Figure 3.7: Bode characteristics of the three complementary sensitivity functions for the current, velocity and distance loops.

The performance of the innermost loop (current loop) is shown Fig.3.8. Although the current controller performs well in the situation when the wheels are stalled (panel on the right), its regulation is worse for spinning wheels, when the back-emf voltage cause oscillation of motor current as shown in Fig.2.6. These current oscillations should be compensated by the current controller. However, the PI controller was able only to partly compensate it. Fortunately it does not seem that current oscillations significantly affect velocity of the car.

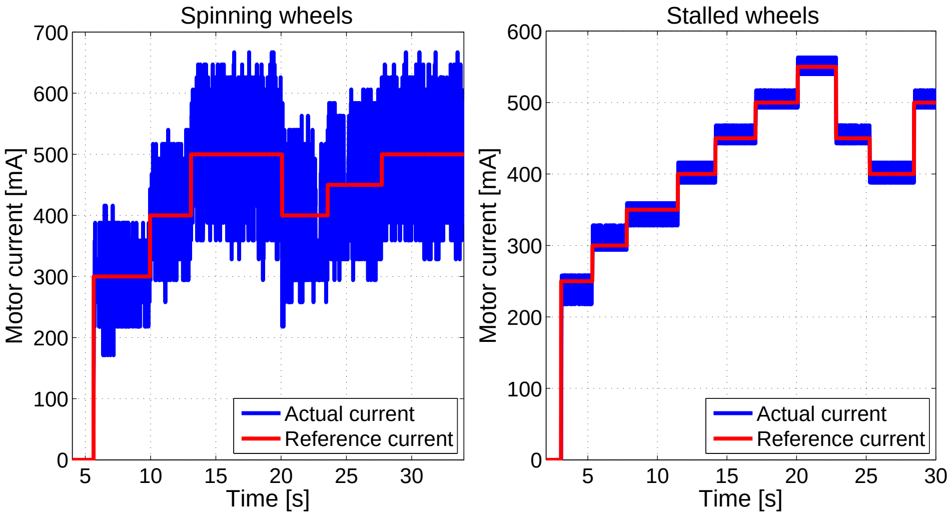


Figure 3.8: Performance of the current controller in two situations: (i) with spinning wheels on the left, (ii) with wheels stalled by hand on the right.

A single car is modeled as

$$T_{v_{ref}\to v} = \frac{25.8(s+0.33)}{(s+27.5)(s+0.31)} \approx \frac{27.5}{s+27.5} = \frac{\beta}{s+\alpha}, \tag{3.14}$$

where $\alpha = \beta = 27.5$.

The similar experiment as in Fig.3.5 was performed to validate the mathematical model of the car. Fig.3.9 shows a good agreement between simulations and observations.

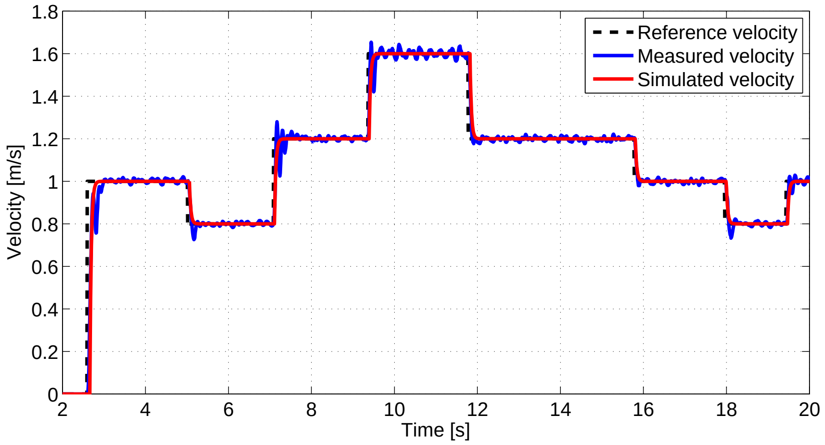


Figure 3.9: Reactions of the car on the steps in the reference signal. Simulation is compared with measurements.

3.2 Platoon

In this section several possible models of the platoon are discussed. They are separated into two categories according to the capability of describing infinite platoons. Platoon of vehicles with leader (indexed by 0) is depicted in Fig.3.10.

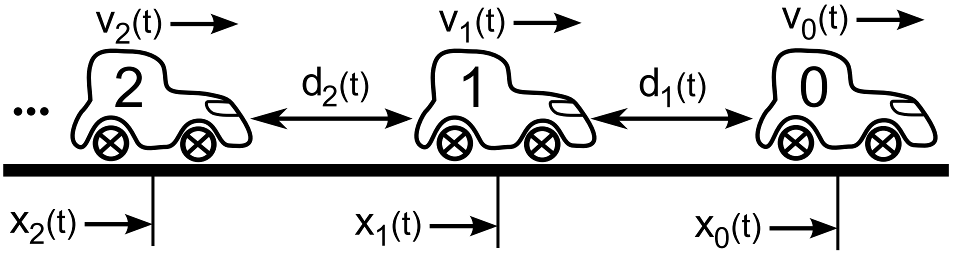


Figure 3.10: Platoon of vehicles with leader (indexed by 0).

3.2.1 Finite length

The state space formalism is a usual way of describing a system. This concept is popular in modeling of finite-length platoon. Here, the model is derived only for three vehicles since it is obvious how to extend the model for more vehicles.

Two state space configurations, mostly discussed in literature, will be shown hereafter.

Velocities and distances between vehicles

This model consists of $(2 \cdot N) - 1$ states, where N represents the number of vehicles in a platoon, is described as

$$\frac{d}{dt}\begin{bmatrix} v_0 \\ d_1 \\ v_1 \\ d_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} -\alpha & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -\alpha \end{bmatrix} \begin{bmatrix} v_0 \\ d_1 \\ v_1 \\ d_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} \beta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \beta \end{bmatrix} \begin{bmatrix} v_{ref_0} \\ v_{ref_1} \\ v_{ref_2} \end{bmatrix} \tag{3.15}$$

where d_n is distance between vehicles indexed n and $n-1$ and v_n is the velocity of the n th vehicle.

Driven distances and velocities

This model consists of $2 \cdot N$ states and its state space description is

$$\frac{d}{dt}\begin{bmatrix} x_0 \\ v_0 \\ x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\alpha \end{bmatrix} \begin{bmatrix} x_0 \\ v_0 \\ x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ \beta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \beta \end{bmatrix} \begin{bmatrix} v_{ref_0} \\ v_{ref_1} \\ v_{ref_2} \end{bmatrix}, \tag{3.16}$$

where x_n is the driven distance of the n th vehicle.

3.2.2 Infinite length

Both previous models are inconvenient for platoons with a large number of cars because of the large-size matrices. Therefore, there were introduced other ways of how to model large platoons.

Decentralized state space approach

A compact way of dealing with large state matrices was introduced by Chu [7]. He expressed the state description as

$$\frac{d}{dt}\begin{bmatrix} x_k(t) \\ \dot{x}_k(t) \end{bmatrix} = \sum_{j=-\infty}^{\infty} A_{k-j} \begin{bmatrix} x_k(t) \\ \dot{x}_k(t) \end{bmatrix} + Bu_k(t), \tag{3.17}$$

used the Z-transform and rewrote the state description into the following form

$$X(z,t) = A(z)X(z,t) + B \cdot U(z,t) \tag{3.18}$$

This formalism transforms Eq.3.15 into

$$X(z,t) = \begin{bmatrix} -\alpha & 0 \\ z^{-1} - 1 & 0 \end{bmatrix} X(z,t) + \begin{bmatrix} 0 \\ \beta \end{bmatrix} \cdot U(z,t), \tag{3.19}$$

where $X(z,t)$ and $U(z,t)$ are the Z-transforms of $\begin{bmatrix} v_k(t) \\ d_{k+1}(t) \end{bmatrix}$ and $\begin{bmatrix} v_{k_{ref}} \end{bmatrix}$, respectively.

The state space description in Eq.3.16 changes into

$$X(z,t) = \begin{bmatrix} 0 & 1 \\ 0 & -\alpha \end{bmatrix} X(z,t) + \begin{bmatrix} 0 \\ \beta \end{bmatrix} \cdot U(z,t), \tag{3.20}$$

where $X(z,t)$ and $U(z,t)$ are the Z-transforms of $\begin{bmatrix} x_k(t) \\ v_k(t) \end{bmatrix}$ and $\begin{bmatrix} v_{k_{ref}} \end{bmatrix}$, respectively.

Polynomial approach

Another compact way to describe a platoon is currently being examined by Šebek and Hurák in [23] and in [24]. A platoon is described as a 2D system using the joint Laplace and Z-transform denoted as $\mathcal{LZ}_1$ transform. The transform operates with variables s and z, which correspond to time and spatial index of the vehicle, respectively.

The platoon system is described as

$$y(s,z) = \frac{b(s,z)}{a(s,z)} u(s,z) + \frac{c(s,z)}{a(s,z)}, \tag{3.21}$$

and the controller as

$$u(s,z) = \frac{q(s,z)}{p(s,z)} e(s,z) + \frac{d(s,z)}{p(s,z)}, \tag{3.22}$$

where

$$e(s,z) = y_{ref}(s,z) - y(s,z). \tag{3.23}$$

and $y(s,z)$ and $u(s,z)$ represent plant output and input, respectively. $2D$ polynomials $a(s,z)$ and $b(s,z)$ represent a system transfer function, $c(s,z)$ is initial and boundary conditions.

Rewriting $y(s,z)$ and $u(s,z)$ into the power series of z^{-1} yields

$$y(s,z) = y_1(s)z^{-1} + y_2(s)z^{-2} + \ldots, \tag{3.24}$$

$$u(s,z) = u_1(s)z^{-1} + u_2(s)z^{-2} + \ldots, \tag{3.25}$$

where $y_k(s)$ and $u_k(s)$ are the output and input of the car at position labeled n.

The distance between vehicles is described as

$$d(s,z) = x(s,z)(z^{-1} - 1), \tag{3.26}$$

where $x(s,z)$ is $2D$ polynomial describing a driven distance of the vehicles. In the same manner, the transfer function from vehicle's velocity to distance between vehicles is

$$G_v(s,z) = \frac{d(s,z)}{v(s,z)} = T_{v_{ref}\to v}(s)\frac{z^{-1}-1}{s}. \tag{3.27}$$

More about the $\mathcal{LZ}_1$ transform and its properties is given in Appendix A.

Chapter 4

Control design

4.1 Control algorithms

In this section controlling algorithms used for the platoon of 10 cars are described. All presented algorithms (except Spatial IIR) were tested by the experimental cars and by mathematical-model simulations. Performances of the algorithms are shown in the next chapter.

The core of each controlling algorithm is a PI controller with transfer function $C(s) = \frac{k_p s + k_i}{s}$, regulating its own input $e(t)$ (error signal) to zero. The way how the input is composed is what distinguishes the individual algorithms. The output of the controller is denoted by $u_{c,k}(t)$, whereas $u_k(t)$ denotes a reference velocity for the velocity controller described in the previous chapter. Usually $u_{c,k}(t) = u_k(t)$ but in the algorithms discussed in this section it does not apply.

It is also presumed that leader of the platoon (indexed by $k = 0$) is controlled by some other means.

4.1.1 Predecessor following algorithm

The predecessor following algorithm is the most simple algorithm to control a platoon. The car only measures distance to the preceding vehicle, in principle, it does not need to communicate with other cars. The scheme in Fig.4.1 shows an alternative way of distance notation between cars, where x_{k-1} and x_k means total driven distance of vehicles with indexes $k - 1$ and k, respectively. Such a notation is useful for performance analysis of the algorithm.

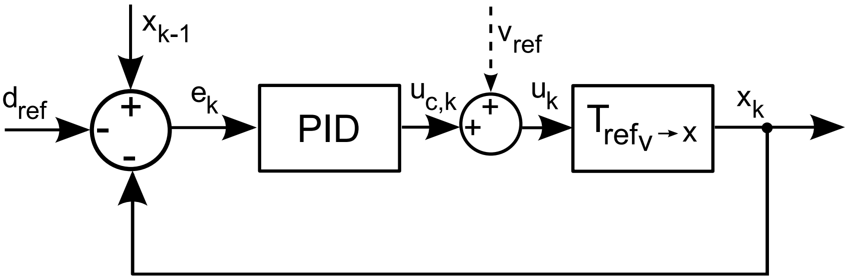


Figure 4.1: Predecessor following algorithm scheme

Fig.4.1 also shows an enhancement of the PI controller. The dashed line marked as v_{ref} acts as a feed-forward for the car subsystem and it represents a reference velocity of the whole platoon. The feed-forward is simply added to the output of the controller so the integration part does not need to integrate that much, when the platoon reference velocity changes (for example when the platoon starts). It does not effect the string stability, though it can improve the performance of the system. In our experiments the information about a platoon reference velocity is provided to all vehicles wirelessly through XBee from PC once reference velocity is changed for all vehicles (not just leader).

As suggested above, the input of the controller is inter-vehicle distance $x_{k-1}(t) - x_k(t)$ minus the reference distance $d_{ref}(t)$

$$e_{k_{PF}}(t) = x_{k-1}(t) - x_k(t) - d_{ref}(t) \tag{4.1}$$

In the $\mathcal{LZ}_1$ formalism the input of the controller described as

$$e(s,z)_{PF} = d(s,z) - d_{ref} = x(s,z)(z^{-1} - 1) - d_{ref}. \tag{4.2}$$

4.1.2 Leader following algorithm

As the name indicates the leader following algorithm regulates a distance to the leader as shown in Fig.4.2. There is no string created in the controlling loop in this type of algorithm, since only a distance to the leader is what matters. But, in our case, all cars have to cooperate since only inter-vehicular distances are measured. So, a communication string of vehicles is created. Each car measures a distance to the preceding vehicle, the increment of this value by its predecessor's distance to the leader and send this value to the following car. It should be noted that, for simplicity, the car physical length is neglected since all cars are identical.

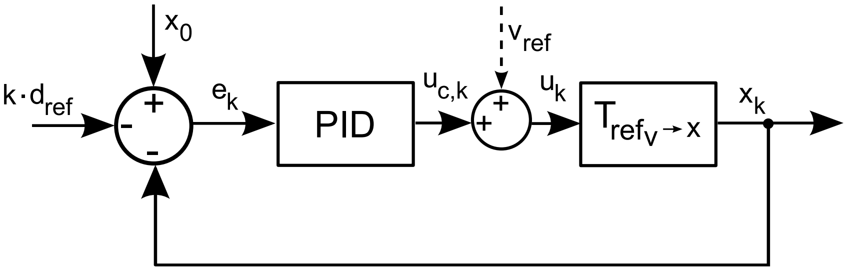


Figure 4.2: Leader following algorithm scheme

The input of controller is

$$e_{k_{LF}}(t) = x_0(t) - x_k(t) - k \cdot d_{ref}(t) \tag{4.3}$$

or, using $\mathcal{LZ}_1$

$$e(s,z)_{LF} = x(s,z)(z^{-k} - 1) - k \cdot d_{ref}. \tag{4.4}$$

4.1.3 Spatial IIR algorithm

The spatial IIR algorithm shown in Fig.4.3 combines above algorithms. The input of the controller is the same as for the predecessor following algorithm but controller's output u_{k-1} of the preceding car is added to the controllers output. There is no need for the feed-forward but wireless communication between cars is essential for this setting.

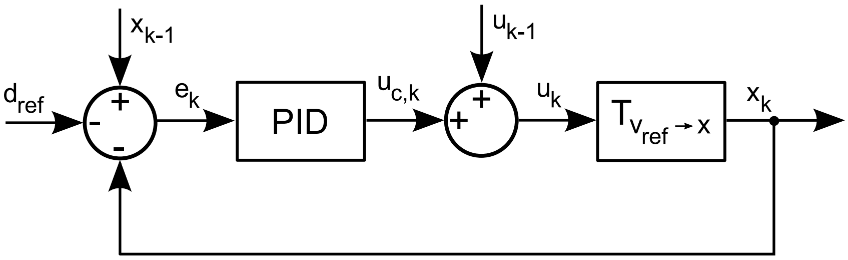


Figure 4.3: Spatial IIR algorithm scheme

The input of the controller is

$$e_k = x_{k-1}(t) - x_k(t) - d_{ref}(t). \tag{4.5}$$

or, in $\mathcal{LZ}_1$ transform

$$e(s,z)_{PF} = d(s,z) - d_{ref} = x(s,z)(z^{-1} - 1) - d_{ref} \tag{4.6}$$

whereas the input of the car subsystem (reference velocity) is

$$u_k(t) = u_{c,k}(t) + u_{k-1}(t) = u_{c,k}(t) + u_{c,k-1}(t) + u_{k-2}(t) = \ldots = \sum_{i=1}^{k} u_{c,i}(t) + u_0(t), \tag{4.7}$$

where

$$\sum_{i=1}^{k} u_{c,i}(s) = (x_0(s) - x_1(s))C(s) + (x_1(s) - x_2(s))C(s) + \ldots + (x_{k-1}(s) - x_k(s))C(s) \tag{4.8}$$

$$= (x_0(s) - x_k(s))C(s), \tag{4.9}$$

and $x_k(s)$ is the Laplace transform of signal $x_k(t)$.

In principle, the Spatial IIR algorithm is a leader following algorithm with the feed-forward equal to the reference velocity of the leader. It is important to point out the difference between this feed-forward and the feed-forward of the platoon reference velocity. The former is the actual reference velocity of the leader, so when the leader for some reason changes its velocity (for instance, it has to break due to an obstacle), the other vehicles are notified about it, while in the latter case an external command is sent to adjust the velocity of the whole platoon (for example to start the platoon).

4.1.4 Bidirectional algorithm

The bidirectional algorithm is yet another way to control a platoon. It is modification of the predecessor following algorithm. The controller operates with distances to the preceding and the following vehicles as shown in Fig.4.4. Theoretically, the vehicles do not need to communicate but cars in our experiments are capable of measuring only front distance (distance to predecessor). To make algorithm functioning, each car sends a measured distance to its predecessor. The last car in the platoon is controlled by a regular predecessor following algorithm.

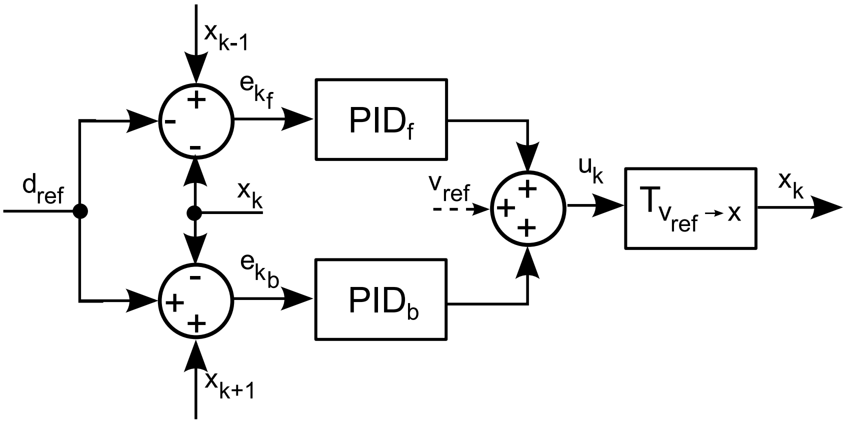


Figure 4.4: Bidirectional algorithm scheme

There are two separate PI controllers in this algorithm. One controls a distance to predecessor (noted as the *front* controller) and the other controls a distance to follower (noted as the *back* controller). The inputs of these controllers are

$$e_{f,k} = x_{k-1} - x_k - d_{ref}, \tag{4.10}$$

$$e_{b,k} = x_{k+1} - x_k + d_{ref}, \tag{4.11}$$

or, in $\mathcal{LZ}_1$ notation

$$e_f(s,z) = d(s,z) - d_{ref} = x(s,z)(z^{-1} - 1) - d_{ref}, \tag{4.12}$$

$$e_b(s,z) = d(s,z) - d_{ref} = x(s,z)(z - 1) + d_{ref}. \tag{4.13}$$

There are two possible configurations of this algorithm. First, a symmetric configuration where $C_f = C_b$. In our case, $C_f = C_b = \frac{2s+1}{s}$. Second, an asymmetric configuration, where $C_f \neq C_b$, $C_f = \frac{2s+1}{s}$ and $C_b = \frac{1s+1}{s}$ was chosen.

Due to time and space restrictions, only asymmetric configuration will be tested.

in other words controllers are not equal. It was chosen $C_f = \frac{2s+1}{s}$ and $C_b = \frac{1s+1}{s}$.

For time reasons and also to save space, it will be tested and simulated only asymmetric configuration.

4.2 String stability

This section introduces the term string stability as well as analytically derives whether presented controlling algorithms are stable.

4.2.1 Definition

String stability was introduced by [8] and it describes how an error signal propagates through a platoon. A necessary condition for the string stability is

$$\left\| \frac{e_i}{e_{i-1}} \right\|_\infty = \left\| \frac{x_i}{x_{i-1}} \right\|_\infty \leq 1. \tag{4.14}$$

In other words, a controlling algorithm is stable if an error signal is not amplified when propagating through a platoon. The concept of string stability was later extended by [9] for nonlinear systems.

It is important to note that even if the system is string unstable it can still be utilized to control the platoon. This situation occurs in case of the Predecessor following algorithm, when the system is string unstable but platoon of 10-slot cars did not crash. However, it is inevitable that for a longer platoon or more demanding reference signals they would crash.

4.2.2 String stability of controlling algorithms

Let us recall the model of our car,

$$T_v = \frac{x}{v_{ref}} = \frac{27.5}{s^2 + 27.5s} = \frac{\beta}{s^2 + \alpha \cdot s}, \tag{4.15}$$

and of distance controller,

$$C = \frac{k_p s + k_i}{s} = \frac{2s + 1}{s} \tag{4.16}$$

where x is the driven distance of the vehicle and v_{ref} is a reference velocity of the vehicle, which is a commanded velocity from the distance controller.

The feed-forward introduced in the previous section is not considered here since it does not affect string stability.

Predecessor following

The input of the car with index k in the predecessor following algorithm is

$$u_k = C(x_{k-1} - x_k - d_{ref}) = \frac{x_k}{T_v}. \tag{4.17}$$

By this, it can be derived

$$T_{PF} = \frac{x_k}{x_{k-1}} = \frac{C}{C + \dfrac{1}{T_v}} = \frac{(k_p s + k_i)\beta}{s^3 + \alpha \cdot s^2 + k_p \cdot \beta \cdot s + k_i \cdot \beta} \tag{4.18}$$

Evaluating $\mathcal{H}_\infty$ norm of T_{PF} yields

$$\|T_{PF}\|_\infty = 1.159 \tag{4.19}$$

We can conclude that the platoon is string unstable since the norm is larger than one. That is in agreement with the above mentioned results.

Leader following

Although the Leader following algorithm does not contain any topological string it will be analytically shown that it is string stable. Driven distance of the car with index k is

$$x_k = C\frac{1}{T_v}(x_0 - x_k - k \cdot d_{ref}). \tag{4.20}$$

This can be rewritten as

$$x_0 = x_k(C \cdot T_v + 1) + k \cdot d_{ref}, \tag{4.21}$$

or

$$x_0 = x_{k-1}(C \cdot T_v + 1) + (k-1) \cdot d_{ref}. \tag{4.22}$$

Therefore

$$x_k(C \cdot T_v + 1) + k \cdot d_{ref} = x_{k-1}(C \cdot T_v + 1) + (k-1) \cdot d_{ref} \tag{4.23}$$

and

$$T_{LF} = \frac{x_k}{x_{k-1}} = \frac{C \cdot T_v + 1}{C \cdot T_v + 1} = 1. \tag{4.24}$$

The norm, of T_{LF} is also less than one, proving that the leader following algorithm is string stable. However, this conclusion applies if it is possible to directly measure a distance to the leader, which is in practice challenging. It does not take into account delay in wireless transmission between vehicles. The effect of the delay is examined in the diploma thesis of Karel Němec.

Spatial IIR

It was already proven that Spatial IIR algorithm is the same as the Leader following algorithm with feed-forward from leader's reference velocities and that Leader following algorithm is stable. Since feed-forward does not affect string stability, we can directly say that Spatial IIR algorithm is also string stable.

Bidirectional algorithm

In the bidirectional algorithm there are two possible ways how the error can propagate in a platoon, namely either to the front or to the rear.

Driven distance of the vehicle with index k is described as

$$x_k = T_v \cdot C_f(x_{k-1} - x_k - d_{ref}) + T_v \cdot C_b(x_{k+1} - x_k + d_{ref}), \tag{4.25}$$

where C_f and C_b are PI controllers for the front and rear distance, respectively. From this, it can be derived that

$$T_{BD} \text{ front} = \frac{x_k}{x_{k-1}} = \frac{T_v C_f}{1 + T_v C_f + T_v C_b}, \tag{4.26}$$

$$T_{BD} \text{ back} = \frac{x_k}{x_{k+1}} = \frac{T_v C_b}{1 + T_v C_f + T_v C_b}. \tag{4.27}$$

We will investigate both possible configurations for this algorithm. The symmetric configuration where $C_f = C_b = \frac{2s+1}{s}$ are norm of complementary sensitivities

$$\|T_{BD} \text{ front}\|_\infty = \|T_{BD} \text{ back}\|_\infty = 0.546. \tag{4.28}$$

Hence, this means the platoon is string stable in this configuration. The asymmetric configuration is examined with controllers $C_f = \frac{2s+1}{s}$ and $C_b = \frac{1s+1}{s}$. Then

$$\|T_{BD} \text{ front}\|_\infty = \|\frac{145.2s + 23.23}{s^3 + 58s^2 + 145.2s + 46.46}\|_\infty = 0.725, \tag{4.29}$$

$$\|T_{BD} \text{ back}\|_\infty = \|\frac{23.23}{s^3 + 58s^2 + 145.2s + 46.46}\|_\infty = 0.5. \tag{4.30}$$

The platoon is string stable also in this case.

4.2.3 String stability and 2D BIBO stability

The second concept of stability that appears in literature is 2D BIBO stability. It is recently being investigated by Hurák and Šebek in [23]. 2D BIBO stability is more restrictive than string stability. Whereas string stability operates with disturbance impulses in the platoon (i.e. changing the reference distance or reference velocity for one vehicle without sending this information to other vehicles), the 2D BIBO stability allows any bounded disturbance signal to act, as for instance, a step signal. (i.e. changing reference distance for all vehicles).

The system is 2D BIBO stable if any signal in the platoon (output, error, input) remains bounded as a response on the bounded disturbance signal. A more rigorous definition is given in [25] (Theorem 4.3, pp. 126). A spatially distributed 2D system with a coprime transfer $f(s,z) = b(s,z)/a(s,z)$ is BIBO stable if

$$a(s, e^{j\omega}) \neq \quad \forall s \in \mathbb{C}, \omega \in \mathbb{R} : \mathbf{R}(s) \leq 0, \omega \in [0, 2\pi]. \tag{4.31}$$

In other words, if the polynomial in s is stable after substituting for z a complex number on the unit circle.

4.3 State-space LQR approach

This section examines another approach to stabilize a platoon. It employs a typical state space model description and uses the LQR approach to stabilize it. It was originally studied by Levine and Athans in [4] and, subsequently, by Melzer and Kuo in [5]. More than 30 years later Jovanovich and Bamieh showed in [17] that both formulations are effectively ill-posed and are incapable to stabilize large platoons. In this section, these problems are recalled and controllers used for stabilization of our platoon are proposed.

The demonstration is given only for 3 vehicles since it is obvious how to extend it for more vehicles. However, numerical simulations will be done for 50 vehicles.

Let us quickly recall the procedure of LQR controller design. For a system

$$\dot{x} = Ax + Bu, \tag{4.32}$$

state feedback law $u = -Kx$ is used, which minimizes the quadratic cost function

$$J(u) = \int_0^\infty (x^T Q x + u^T R u + 2x^T N u) dt. \tag{4.33}$$

Feedback gain K is calculated using the Algebraic Riccati equation (ARE)

$$A^T S + SA - (SB + N)R^{-1}(B^T S + N^T) + Q = 0 \tag{4.34}$$

as

$$K = R^{-1}(B^T S + N^T). \tag{4.35}$$

To simplify the problem we choose $N = 0$.

We will focus on the maximal and minimal eigenvalues of the S (the solution of ARE) and on maximal real part of the eigenvalues for $A - BK$ (closed loop).

4.3.1 Levine and Athans configuration

Levine and Athans [4] used a configuration of velocities and distances between vehicles. The system description of platoon with three vehicles for LQR design then looks as

$$\frac{d}{dt}\begin{bmatrix} \delta v_0 \\ \delta d_1 \\ \delta v_1 \\ \delta d_2 \\ \delta v_2 \end{bmatrix} = \begin{bmatrix} -\alpha & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -\alpha \end{bmatrix}\begin{bmatrix} \delta v_0 \\ \delta d_1 \\ \delta v_1 \\ \delta d_2 \\ \delta v_2 \end{bmatrix} + \begin{bmatrix} \beta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \beta \end{bmatrix}\begin{bmatrix} \delta v_{ref_0} \\ \delta v_{ref_1} \\ \delta v_{ref_2} \end{bmatrix}, \tag{4.36}$$

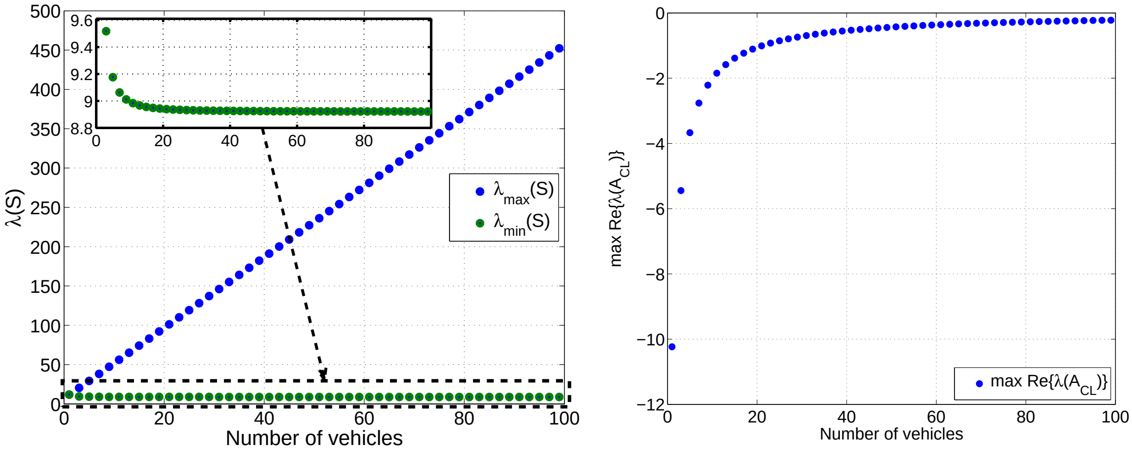
where δv_0 is velocity error of the car with index 0 (leader), δd_1 represents a distance error between cars with indices 0 and 1, and N is the total number of vehicles (platoon leader included).

The weighting matrices Q and R are diagonal of the form

$$Q = \begin{bmatrix} p & 0 & 0 & 0 & 0 \\ 0 & q & 0 & 0 & 0 \\ 0 & 0 & p & 0 & 0 \\ 0 & 0 & 0 & q & 0 \\ 0 & 0 & 0 & 0 & p \end{bmatrix}, \qquad R = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix}, \tag{4.37}$$

where p, q and r penalize velocity error, distance error and input to the system, respectively. For simulations $p = 100$ and $q = r = 1$ were chosen.

Values of $\lambda_{max}(S)$, $\lambda_{min}(S)$ and $max\mathbf{Re}\{\lambda(A_{CL}\}$ as a functions of the number of vehicles are shown in Fig.4.5. We can see that the maximal eigenvalue grows with the increasing number of vehicles, ultimately making impossible to stabilize the system. This results in the convergence of the closed loop dominant poles to the origin (stability margin).



(a) Maximal and minimal eigenvalues

(b) Eigenvalues of closed loop system

Figure 4.5: Eigenvalues of ARE solution for 100 vehicles for Levine and Athans configuration.

4.3.2 Melzer and Kuo

Melzer and Kuo [5] configuration consists of driven distance errors $\delta\epsilon_n$ and velocity errors δv_n of individual vehicles. A platoon is then described as

$$\frac{d}{dt}\begin{bmatrix}\delta\epsilon_0\\ \delta v_0\\ \delta\epsilon_1\\ \delta v_1\\ \delta\epsilon_2\\ \delta v_2\end{bmatrix} = \begin{bmatrix}0&1&0&0&0&0\\ 0&-\alpha&0&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&-\alpha&0&0\\ 0&0&0&0&0&1\\ 0&0&0&0&0&-\alpha\end{bmatrix}\begin{bmatrix}\delta\epsilon_0\\ \delta v_0\\ \delta\epsilon_1\\ \delta v_1\\ \delta\epsilon_2\\ \delta v_2\end{bmatrix} + \begin{bmatrix}0&0&0\\ \beta&0&0\\ 0&0&0\\ 0&\beta&0\\ 0&0&0\\ 0&0&\beta\end{bmatrix}\begin{bmatrix}\delta v_{ref_0}\\ \delta v_{ref_1}\\ \delta v_{ref_2}\end{bmatrix}. \tag{4.38}$$

The weighting matrices Q and R are following

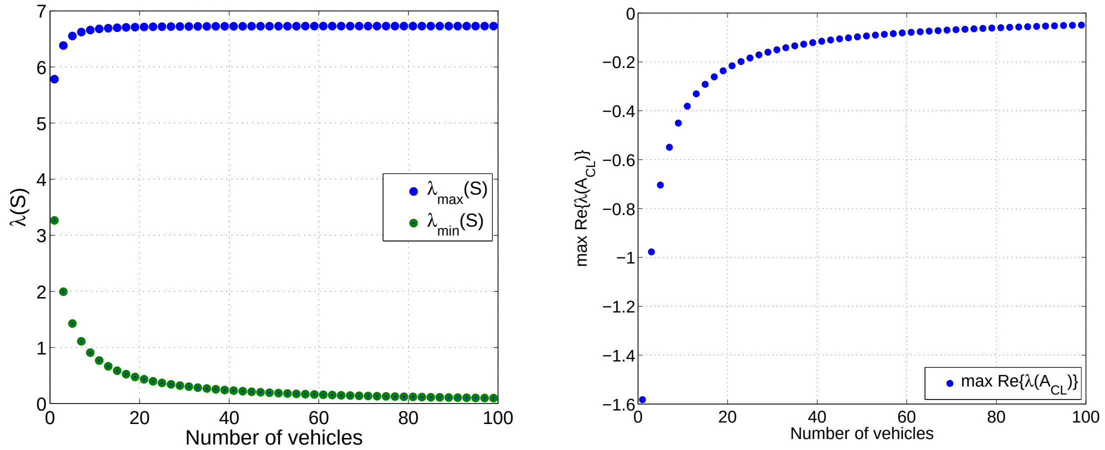
$$Q = \begin{bmatrix} q & 0 & -q & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 \\ 0 & 0 & q & 0 & -q & 0 \\ 0 & 0 & 0 & p & 0 & 0 \\ 0 & 0 & 0 & 0 & q & 0 \\ 0 & 0 & 0 & 0 & 0 & p \end{bmatrix}, \qquad R = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix}. \tag{4.39}$$

Since Q must be symmetric it is modified as

$$Q_{sym} = \frac{Q + Q^T}{2}. \tag{4.40}$$

The system was simulated with $p_{lq} = 10$ and $q_{lq} = r_{lq} = 1$.

The same analysis as in the previous case is depicted in Fig.4.6. In this case, the minimal eigenvalue converges to zero, making more and more difficult to stabilize large platoons. Dominant poles of the system are moving towards the stability margin, so it is impossible to stabilize very large platoons.



(a) Maximal and minimal eigenvalues

(b) Eigenvalues of closed loop system

Figure 4.6: Eigenvalues of ARE solution for 100 vehicles for Melzer and Kuo configuration.

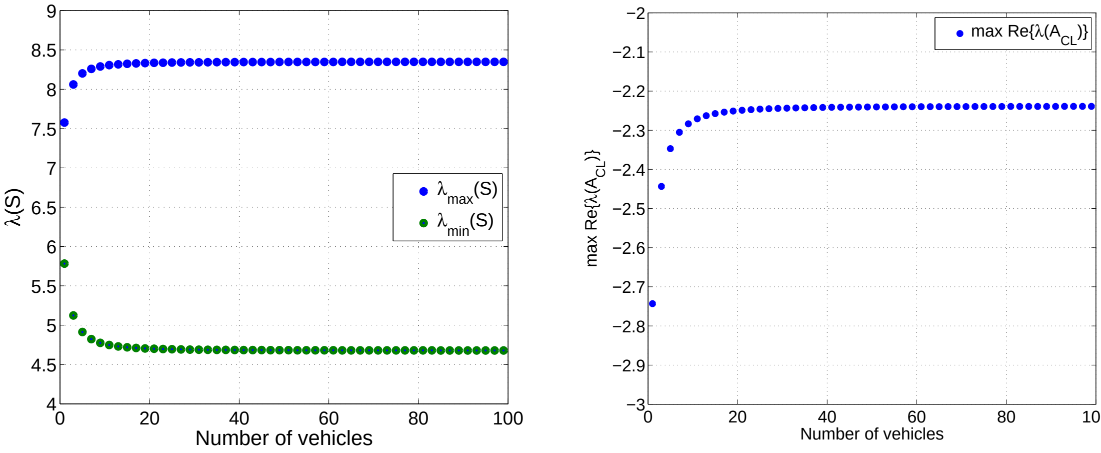
4.3.3 Jovanovich and Bamieh

Jovanovic and Bamieh [17] proposed the same model as (4.38) but they modified the weighting matrix Q as

$$Q = \begin{bmatrix} 2q & 0 & -q & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 \\ 0 & 0 & 2q & 0 & -q & 0 \\ 0 & 0 & 0 & p & 0 & 0 \\ 0 & 0 & 0 & 0 & 2q & 0 \\ 0 & 0 & 0 & 0 & 0 & p \end{bmatrix}. \tag{4.41}$$

Weighting matrix R is the same as in (4.39) as well as mentioned values for the simulations.

The maximal and minimal eigenvalues of the solution to ARE are shown in Fig.4.7. In this case, both minimal and maximal eigenvalues converge to non-zero value, resulting in a stabilizable system even for very large (infinite) platoons.



(a) Maximal and minimal eigenvalues

(b) Eigenvalues of closed loop system

Figure 4.7: Eigenvalues of ARE solution for 100 vehicles for Jovanovich and Bamieh configuration.

Not only numerical results but also a rigorous analysis of all the three configurations were done by Jovanovich and Bamieh in [17].

Chapter 5

Experiments and simulations

The measured data with the 10-vehicle platoon are now compared with simulations. To keep results transparent only situations with interesting phenomena are presented. Each algorithm is supplemented with 50-vehicle simulation to fully demonstrate characteristics of the control algorithms.

Two typical situations were experimented and simulated with the platoon: (i) the reference distance between cars (d_{ref}) is changed, (ii) the reference velocity of the platoon leader ($v_{0_{\mathrm{ref}}}$) is changed.

5.1 Reference-distance change

In this case, the reference distance (d_{ref}) between vehicles is showed via wireless communication. This experiment is partly related to the 2D-BIBO stability, but it does not verify the string stability.

This experiment is however quite unrealistic, since it is ill-conditioned. Changing the reference distance for all vehicles results in unrealistic demands on the cars in platoon, matter what algorithm is used. For example, if the first car has to get 10 meter closer to the leader, the car on the position 100 has to drive 1000 meters, preferably simultaneously. Nevertheless, it is an interesting way to stress the algorithms and compare their performances.

Let us note that changing the reference distance would be better done by sequential commands for individual cars, which is in principle equal to the change of the leader's velocity.

5.1.1 Predecessor following algorithm

The predecessor following algorithm is capable to stabilize small platoons such as ours as Fig.5.1 and Fig.5.2 show. Even though the last car significantly overshoots the desired inter-car distance, it does not collide with its predecessor.

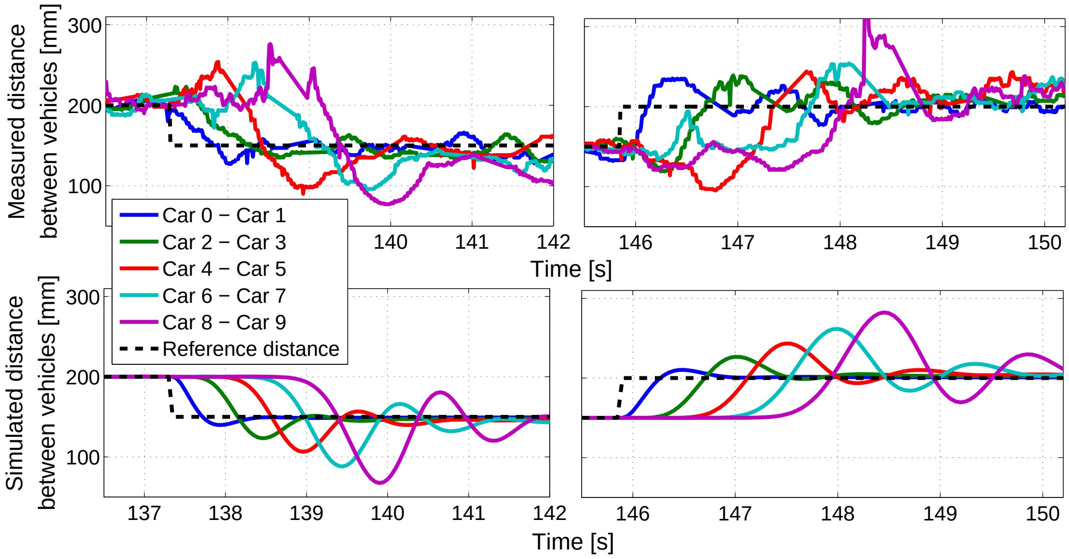


Figure 5.1: Inter-car distances in response to change in d_{ref} for the Predecessor following algorithm.

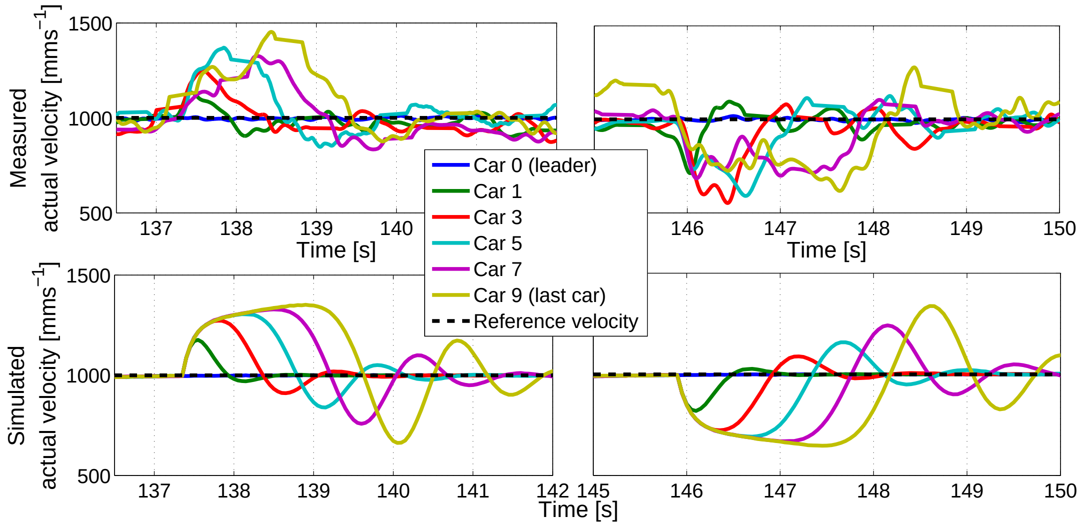


Figure 5.2: Velocities of cars in response to change in d_{ref} for the Predecessor following algorithm.

5.1.2 Leader following algorithm

There are two major differences between responses in the predecessor and leader following algorithms. The first is depicted in Fig.5.3. The inter-car distances is changing at the same time for all vehicles and not sequentially. A similar effect occurs in the velocity response in Fig.5.4, where cars have to drive faster but for a shorter time than in the previous case, though the total driven distance is the same.

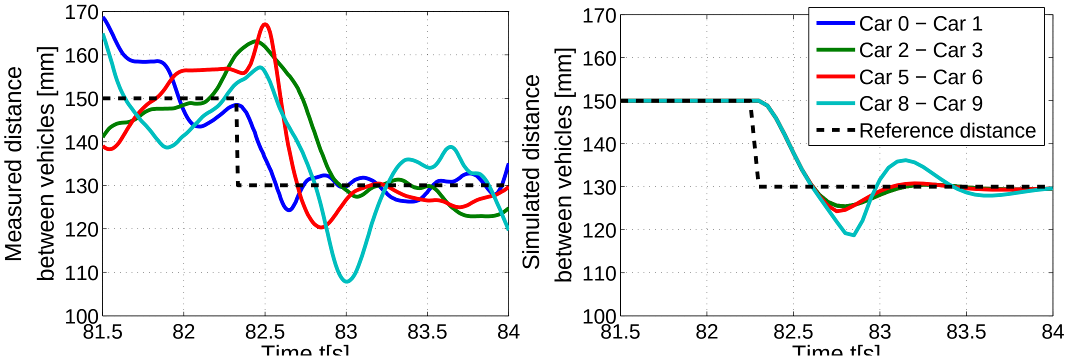


Figure 5.3: Inter-car distances in response to change in d_{ref} for the Leader following algorithm.

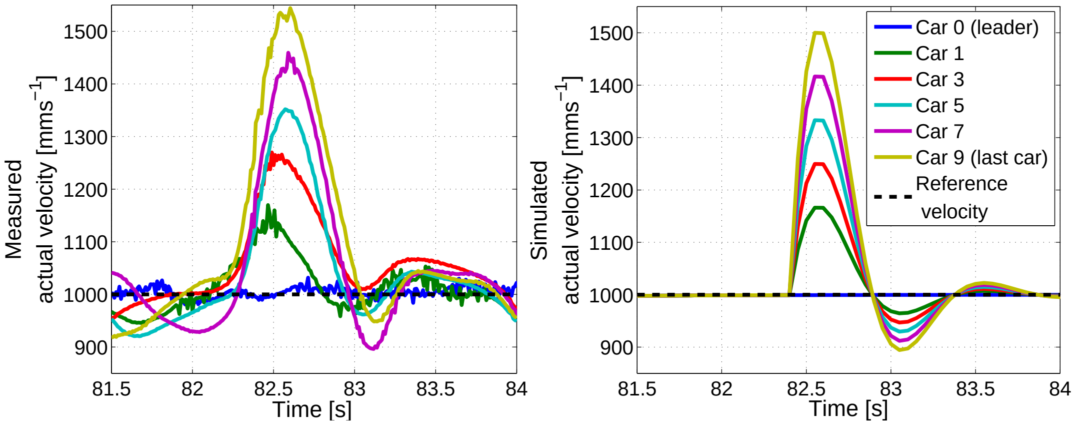


Figure 5.4: Velocities of cars in response to change in d_{ref} for the Leader following algorithm.

5.1.3 Bidirectional algorithm

The results of changes of the reference distance for the bidirectional algorithm are shown in Fig.5.5 and Fig.5.6. The performance of cars is less impulsive and smoother in time that that of the Predecessor following and even Leader following algorithms. Peaks of the velocity signals are smaller compared to the previous case despite the fact that the reference distances were nearly doubled. Almost no amplification of the distance error as Fig.5.7 shows is another important result.

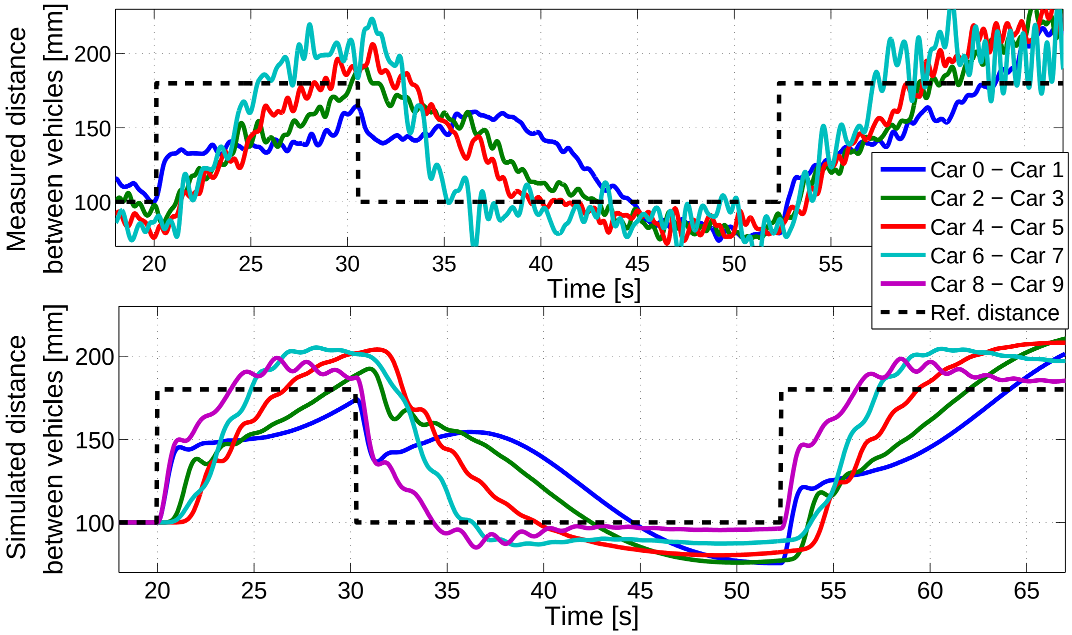


Figure 5.5: Inter-car distances in response to change in d_{ref} for the Bidirectional algorithm.

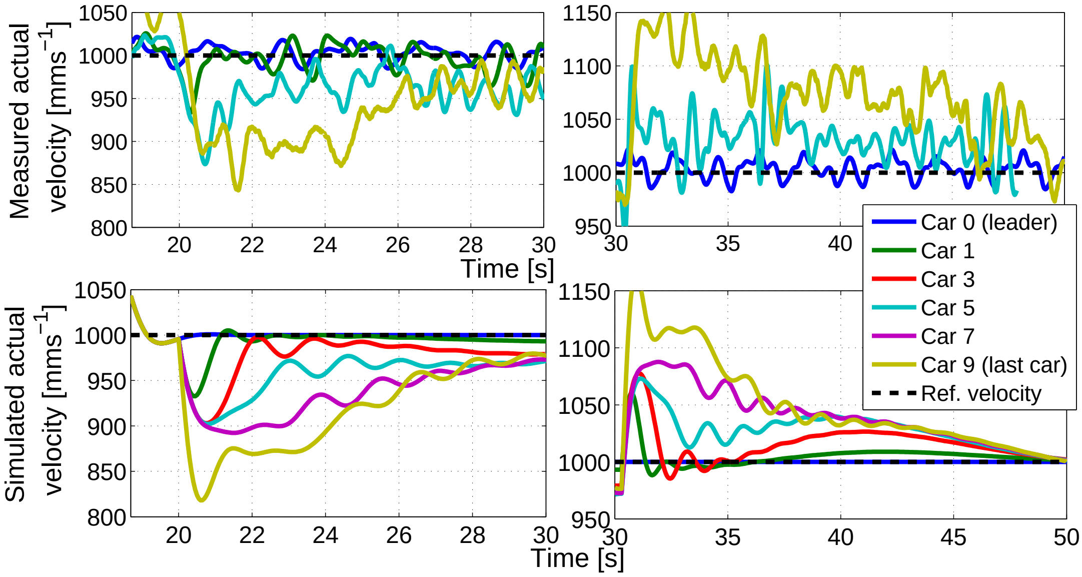


Figure 5.6: Velocities of cars in response to change in d_{ref} for the Bidirectional algorithm.

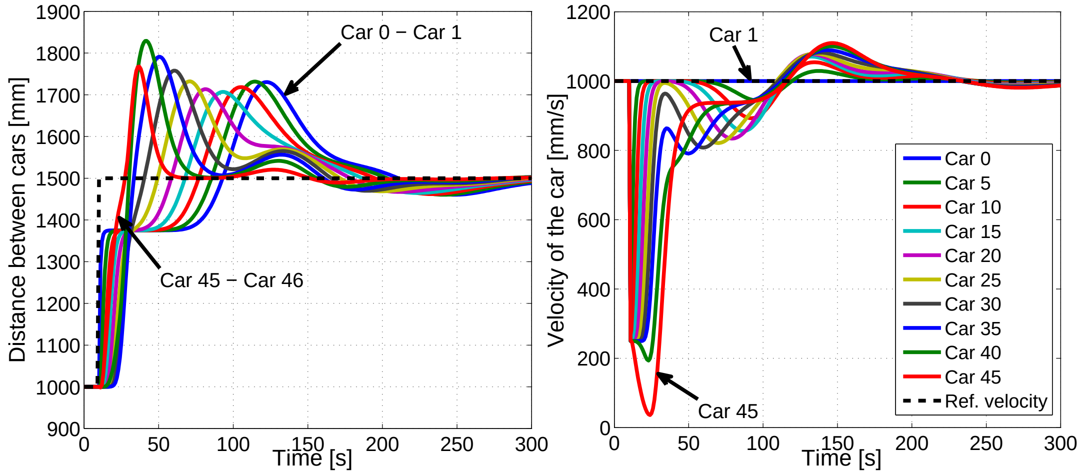


Figure 5.7: Simulation of distances and velocities of fifty cars in response to change in d_{ref} for the Bidirectional algorithm.

5.1.4 Spatial IIR algorithm

Simulation results of the Spatial IIR algorithm resemble those of the Leader following algorithm. A significant difference is that the distances between vehicles remain for all cars identically the same.

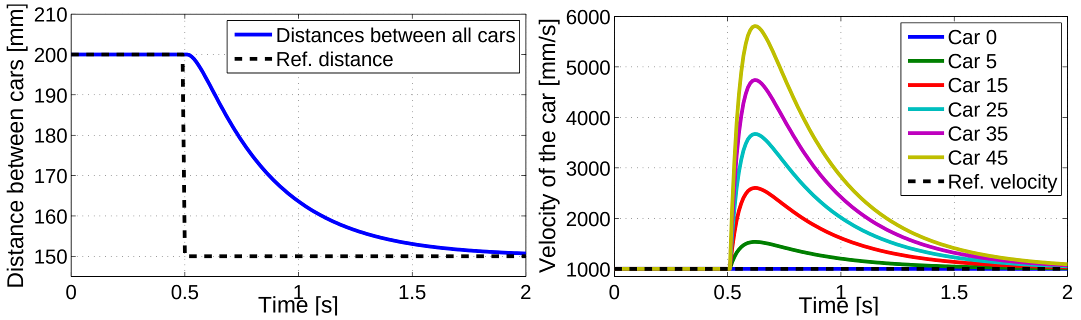


Figure 5.8: Simulation of distances and velocities of fifty cars in response to change in d_{ref} for the Spatial IIR algorithm.

5.2 Change of the platoon leader's velocity

In the second situation, the reference velocity of the platoon leader ($v_{0_{\mathrm{ref}}}$) is changed without notifying other cars. This will verify the string stability of the controllers. Note that it is a more realistic experiment than changing the reference distance for all vehicles.

5.2.1 Predecessor following algorithm

Figs.5.9 and 5.10 show an instability of the platoon. Inter-car distances and velocities of cars are amplified towards the end of a platoon. For a very large platoon without any saturation limits this would ultimately lead to crash as shown in Fig.5.11, where the distances between cars become negative. This result is in full agreement with the analytical result discussed above.

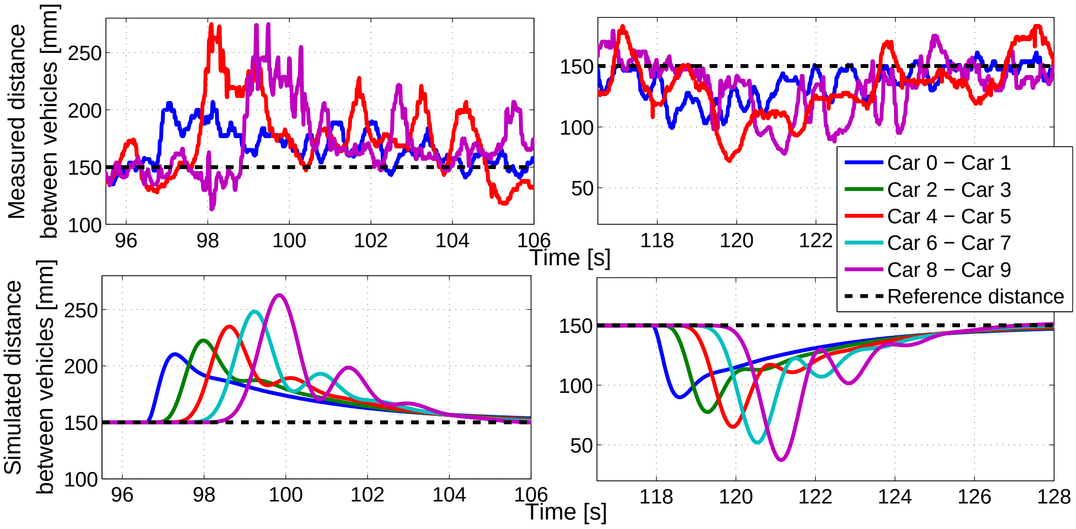


Figure 5.9: Inter-car distances in response to change in $v_{0_{ref}}$ for the Predecessor following algorithm.

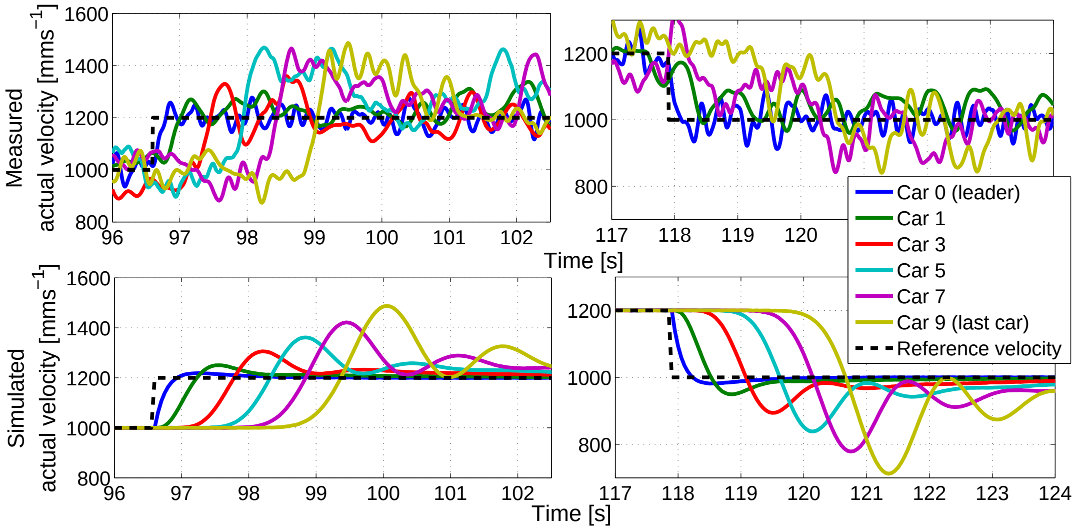


Figure 5.10: Velocities of cars in response to change in $v_{0_{ref}}$ for the Predecessor following algorithm.

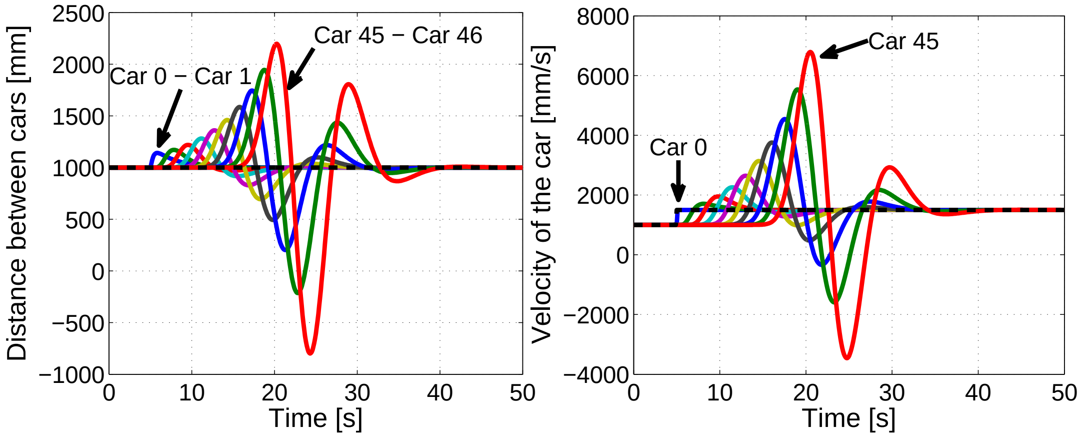


Figure 5.11: Simulation of distances and velocities of fifty cars in response to change in $v_{0_{ref}}$ for the Predecessor following algorithm.

5.2.2 Leader following algorithm

The string stability of the leader following algorithm is depicted in Fig.5.12 and Fig.5.13. The velocities of the cars further in the platoon are not amplified. Both figures nicely show a "synchronous movement" of the followers. Their velocity profiles are identical in the simulations and very similar in the real experiment. Therefore, only distance between the leader and the first follower is changed, while distances between rest of the cars remain the same. Simulations with a larger platoon were not done since they would be identical.

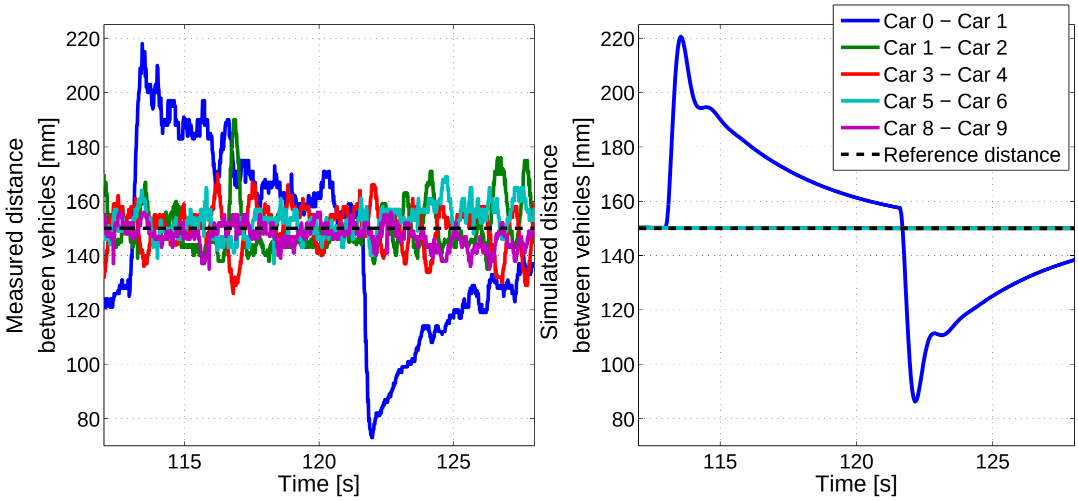


Figure 5.12: Inter-car distances in response to change in $v_{0_{ref}}$ for the Leader following algorithm.

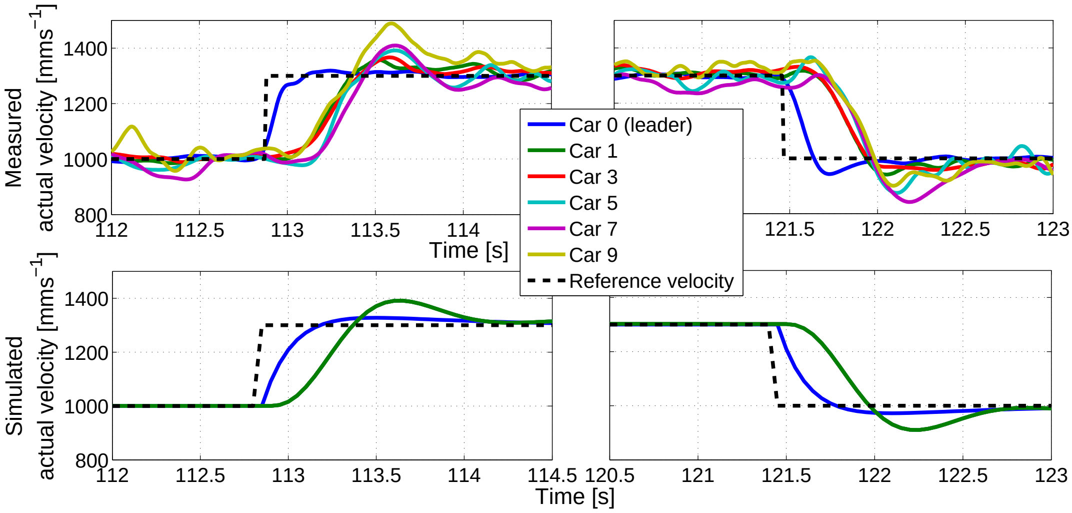


Figure 5.13: Velocities of cars in response to change in $v_{0_{ref}}$ for the Leader following algorithm.

5.2.3 Bidirectional algorithm

The bidirectional algorithm is string stable (at least from the leader → platoon end sense) as shown in Fig.5.15. The maximal peak velocities of the following cars are the same, only more distant cars have to drive with this velocity for a longer time as simulations in Fig.5.16 show.

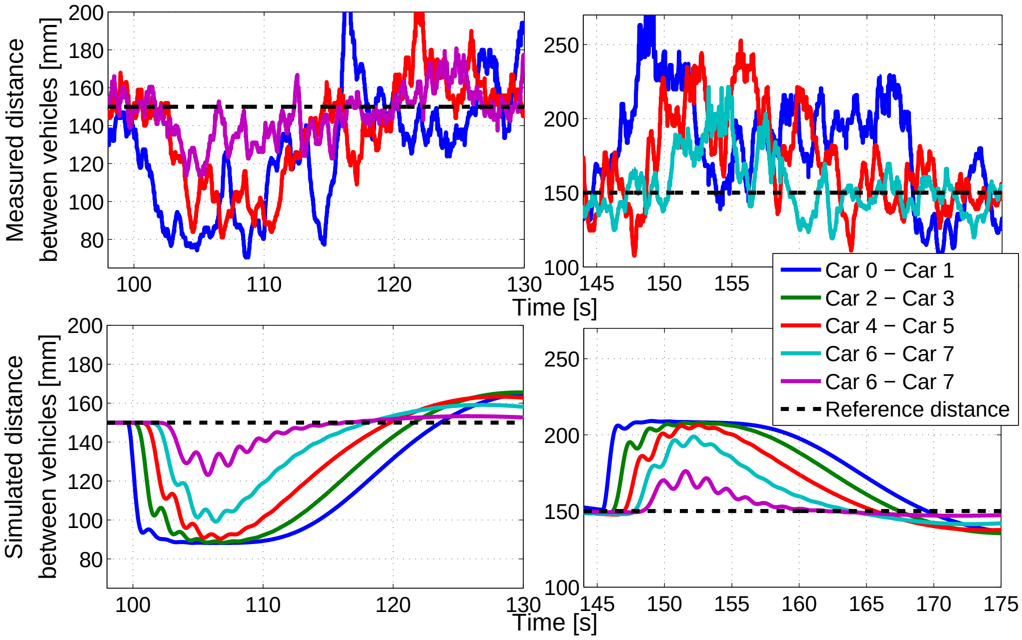


Figure 5.14: Inter-car distances in response to change in $v_{0_{ref}}$ for the Bidirectional algorithm.

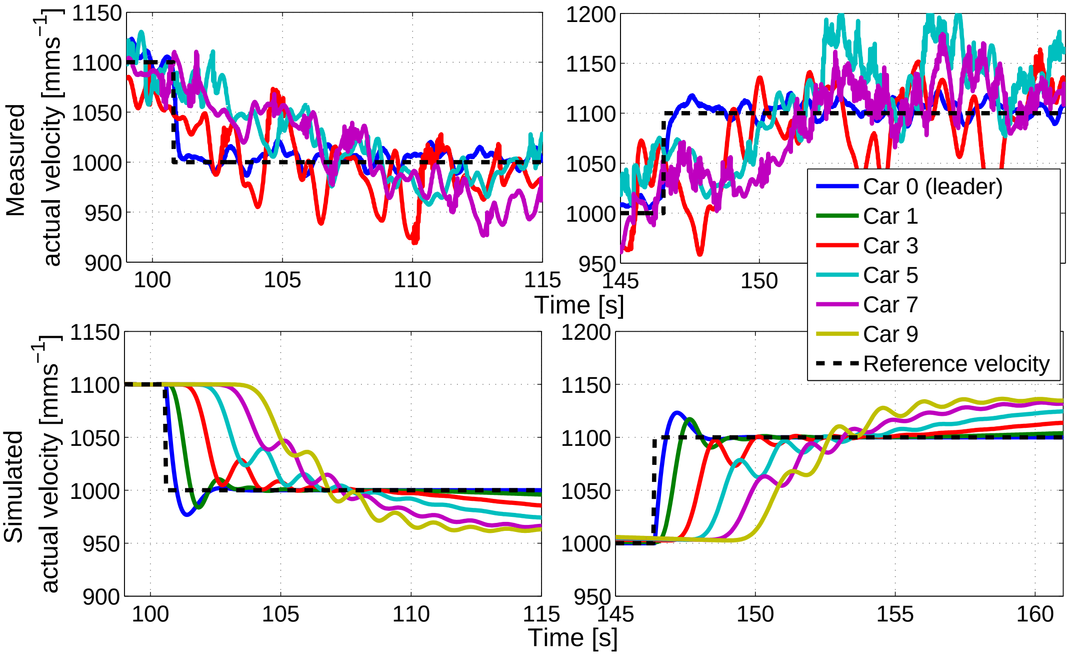


Figure 5.15: Velocities of cars in response to change in $v_{0_{ref}}$ for the Bidirectional algorithm.

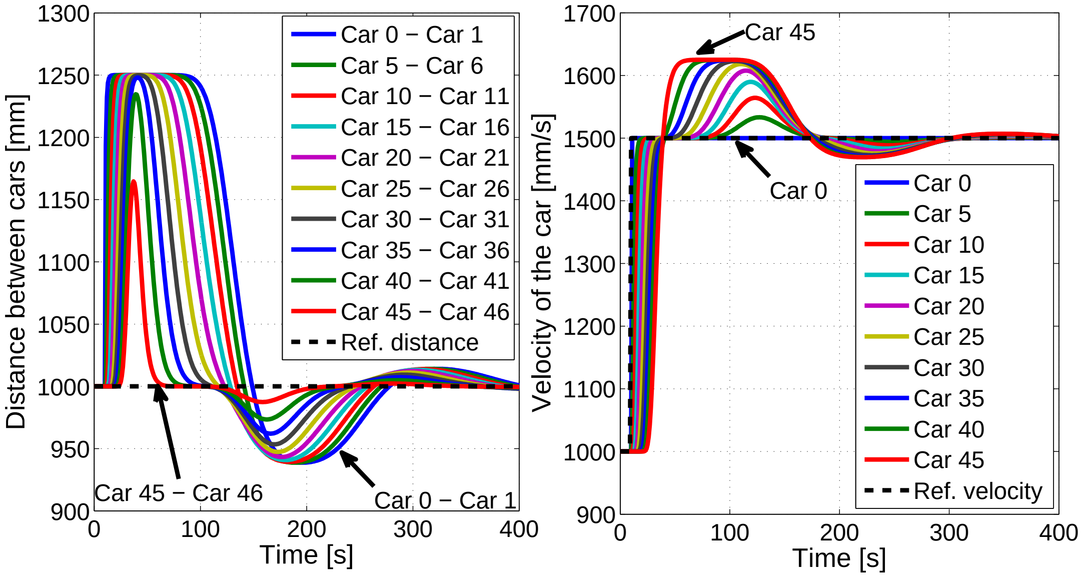


Figure 5.16: Simulation of distances and velocities of fifty cars in response to change in $v_{0_{ref}}$ for the Bidirectional algorithm.

5.2.4 Spatial IIR algorithm

Simulation results of the Spatial IIR algorithm again resembles those of the Leader following algorithm, though, this time results are significantly better. Since the leader automatically broadcasts its reference velocity, the other cars are notified about its change. Therefore, they all accelerate in the same manner and the distances between them remains unchanged. This type of algorithm is string stable. From the simulations in Fig.5.17 it is even impossible to determine a number of vehicles in the platoon since individual curves are overlapping. There were fifty cars in this simulation.

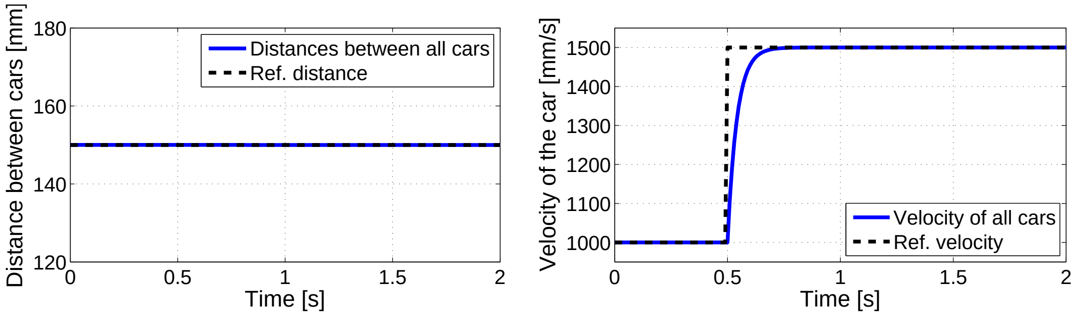


Figure 5.17: Simulation of distances and velocities of fifty cars in response to change in $v_{0_{ref}}$ for the Spatial IIR algorithm.

5.3 LQR controllers

LQR controller is based on the concept that every car in the platoon knows the velocity and distance of the other cars. This would require extensive wireless communication between cars and also a compensation of the communication delays. Such a task exceeds the original topic of this thesis, therefore only simulations on PC were made.

Two types of situations were simulated: (i) the reference distance (d_{ref}) is changed for all cars, as in previous case, (ii) the reference distance (d_{ref}) is changed only for the first car behind leader. The latter experiment imitates the situation when the leader changes its velocity. We followed this alternative to simplify a simulation task since the string instability exhibits in this case.

5.3.1 Reference distance change for all vehicles

Levine and Athans concept

The distance and velocity responses with Levine and Athans controller configuration are symmetric around the values for the car in the middle of the platoon (see Fig.5.18). The distance in front of the first car is the same as the distance in front of the last car for all time instances. Another distinction is that the middle car does not change the velocity while cars on the platoon edges need to go with the highest velocities.

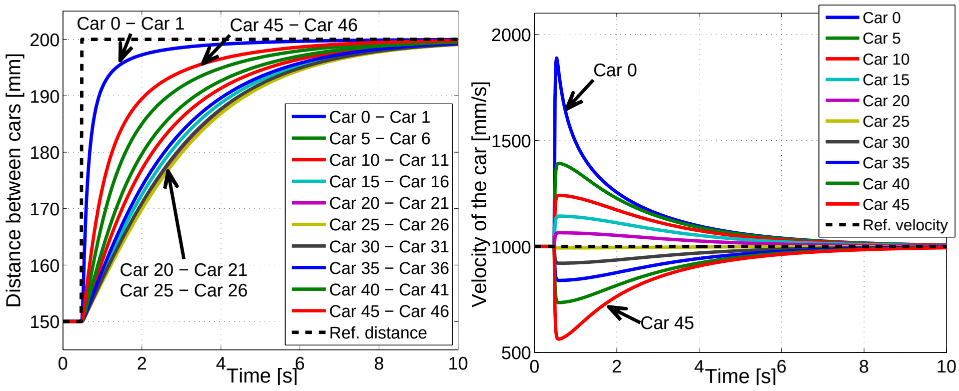


Figure 5.18: Simulation of distances and velocities of fifty cars in response to change in d_{ref} for the Levine and Athans controller configuration.

Melzer and Kuo concept

Melzer and Kuo controller concept forces cars in front of the platoon drive faster as seen from Fig.5.19. We can also see that the inter-vehicular error propagates from the back to the front of the platoon.

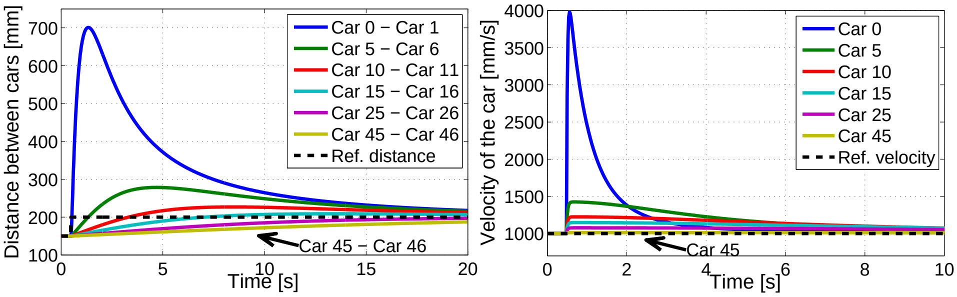


Figure 5.19: Simulation of distances and velocities of fifty cars in response to change in d_{ref} for the Melzer and Kuo controller configuration.

Jovanovich and Bamieh concept

Jovanovich and Bamieh concept results in a similar behavior of cars with the demands that velocities are more evenly distributed with respect to the front car. The most significant distance error is between the leader and the first car, while the error remains almost unchanged for the other cars.

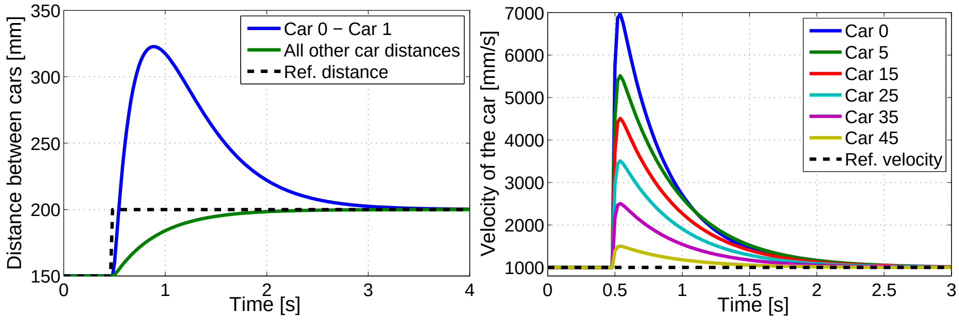


Figure 5.20: Simulation of distances and velocities of fifty cars in response to change in d_{ref} for the Jovanovich and Bamieh controller configuration.

5.3.2 Reference distance change for the first vehicle

The responses to the command on changing the reference distance of the first car are almost identical for all the three LQR controller concepts. Hence, only Fig.5.21 documents them. It is transparent that the system is string stable. The distance and velocity errors are attenuated towards the end of the platoon.

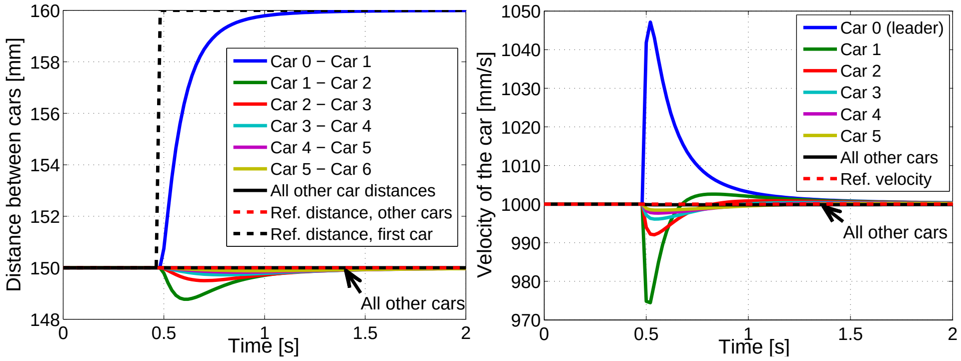


Figure 5.21: Simulation of distances and velocities of fifty cars in response to change in $d_{1_{ref}}$ for all the LQR controller configurations.

Chapter 6

Conclusion

Model of the car

The platform is as a promising tool with computational power capable of handling even more computationally demanding tasks. The platform is equipped with an easy-to-use and cheap Zigbee (Xbee) wireless communication module which makes possible to consider more complex control strategies such as a bidirectional distributed control, or, in the thesis called, the Spatial IIR algorithm. However, certain improvement in the distance-to-the--predecessor measurement is required to reach full potential of the platform. One possible solution has been designed and tested on a two car platoon but, due to lack of time, it has not been tested on a larger scale platoon. Another problem is caused by logging data on the μ-SD card. Occasionally only a part of data was stored for yet unknown reason. Even extensive examination has not fixed up the problem.

The comparison of experimental and simulation data shows the appropriateness of the mathematical model of the car. However, velocity oscillations occurred in measured data are not predicted by the mathematical model. Oscillations in distances between cars were apparently visible during measurements. This problem has not been solved despite extensive endeavor. Possibly, it is caused by small construction differences of individual cars since controller constants tuned for one car do not work as accurately for other cars as expected. Therefore, the constants providing an average performance of all cars were applied. Though, it is quite certain that our experiments reached the limit of car's technical capabilities.

Controllers comparison

The experiments confirm analytical results on string stability of the individual algorithms. However, we should emphasize that even a string unstable algorithm can drive a small platoon of vehicles in a control way. For example, the Previous following algorithm 10-slot cars (see sections 5.1.1 and 5.2.1.). But, error signals in the platoon must be treated carefully. And advantage of the Previous following algorithm is simplicity, since only the distance to the previous car is measured and no communication with other cars is required. However, this algorithm is only applicable on a small platoons due to string instability. Simulations of the platoon with mathematical model of cars are highly recommended to predict a magnitude of string instability, that may not be easy to foretell before experiments or simulations.

The Leader following algorithm is principially several one-to-one Previous following algorithms applied in one platoon. It works perfectly in simulations. However in practice there are a few technical difficulties complicating its performance. Measuring of the distance to the leader is a challenging task. Either one may implement an absolute distance measurement sensor, such as GPS but more enhanced, or one may rely on inter-vehicle distance measurements, such as in our case. The latter way additionally needs to implement a reliable wireless communication between vehicles and accounts for delays in wireless communication. This algorithm is, in principle, not very safe since each vehicle in the platoon relies on front-vehicle performances, therefore any pitfall in performances may result in platoon crash. This suggests that it may be advantageous to combine the Previous following algorithm with the Leader following algorithm. This concept was already studied in literature. In this thesis, the Spatial IIR algorithm is a modification of such a concept.

Bidirectional algorithm is yet another way to control the platoon. It performed well in simulations and experiments. Measuring the distance to the rear vehicle is recommended to exclude wireless communication from the control loop but this is a small price to pay. Even though wireless communication between vehicles is needed to enable increase and decrease the platoon size, it is not involved in the control loop. A drawback of this algorithm is the fact that events happening at the platoon's tail affect front cars, which is an unusual behavior. It means that string stability exhibits in both directions of the platoon.

Spatial IIR algorithm, as introduced, is a combination of Previous and Leader following algorithms. Unfortunately, due to lack of time this algorithm was not tested experimentally.

It combines information about the distance to the predecessor with the velocity of the leader. A reliable wireless communication is again essential, but in this case, communication imperfections are partly compensated by the distance measurement to the predecessor. This type of algorithm gives the best performance of all non LQR controllers, which looks promising for future applications. However, this algorithm relies on important assumption that all cars are identical. More specifically, all cars are assumed to have identical velocity closed loop, and thus the dynamics from a reference velocity to the actual velocity. The algorithm would probably work even if this condition is not exactly fulfilled. However, the distance and velocity responses would be different from our simulations.

As expected, LQR controllers gives the best performance of all applied control algorithms. It works with states (distance and velocity) of all cars implying the requirement on massive and reliable communication between vehicles. This is not a difficulty in simulations but it may be a significant difficulty in implementation. This problem can partly be resolved by using only a part of all states (e.g. only states of three preceding and three following vehicles).

Videos of the platooning slot cars are available at [3].

Bibliography

[1] I. Berkeley, “California path.” ”`http://www.path.berkeley.edu/`”, 2010.

[2] T. P. SARTRE, “Sartre road trains.” ”`http://www.sartre-project.eu/en/about/news/Sidor/roadtrains_video.aspx`”, 2012.

[3] AA4CC, “Distributed control of spatially distributed systems.” ”`http://aa4cc.dce.fel.cvut.cz/content/distributed-control-spatially-distributed-systems`”.

[4] W. Levine and M. Athans, “On the optimal error regulation of a string of moving vehicles,” *Automatic Control, IEEE Transactions on*, vol. 11, no. 3, pp. 355–361, 1966.

[5] S. Melzer and B. Kuo, “A closed-form solution for the optimal error regulation of a string of moving vehicles,” *Automatic Control, IEEE Transactions on*, vol. 16, no. 1, pp. 50–52, 1971.

[6] S. Melzer and B. Kuo, “Optimal regulation of systems described by a countably infinite number of objects,” *Automatica*, vol. 7, pp. 359–366, May 1971.

[7] K. Chu, “Decentralized control of High-Speed vehicular strings,” *Transportation science*, vol. 8, pp. 361–384, Nov. 1974.

[8] R. Cosgriff, “The asymptotic approach to traffic dynamics,” *Systems Science and Cybernetics, IEEE Transactions on*, vol. 5, no. 4, pp. 361–368, 1969.

[9] D. Swaroop and J. Hedrick, “String stability of interconnected systems,” *Automatic Control, IEEE Transactions on*, vol. 41, no. 3, pp. 349–357, 1996.

[10] L. Peppard, “String stability of relative-motion PID vehicle control systems,” *Automatic Control, IEEE Transactions on*, vol. 19, no. 5, pp. 579–581, 1974.

[11] P. Seiler, A. Pant, and K. Hedrick, “Disturbance propagation in vehicle strings,” *Automatic Control, IEEE Transactions on*, vol. 49, no. 10, pp. 1835–1842, 2004.

[12] B. Bamieh, F. Paganini, and M. Dahleh, “Distributed control of spatially invariant systems,” *Automatic Control, IEEE Transactions on*, vol. 47, no. 7, pp. 1091–1107, 2002.

[13] R. D’Andrea and G. Dullerud, “Distributed control design for spatially interconnected systems,” *Automatic Control, IEEE Transactions on*, vol. 48, no. 9, pp. 1478–1495, 2003.

[14] G. Stewart, D. Gorinevsky, and G. Dumont, “Feedback controller design for a spatially distributed system: the paper machine problem,” *Control Systems Technology, IEEE Transactions on*, vol. 11, no. 5, pp. 612–628, 2003.

[15] M. Jovanović and B. Bamieh, “Lyapunov-based distributed control of systems on lattices,” *Automatic Control, IEEE Transactions on*, vol. 50, no. 4, pp. 422–433, 2005.

[16] G. Stein and D. Gorinevsky, “Design of surface shape control for large two-dimensional arrays,” *Control Systems Technology, IEEE Transactions on*, vol. 13, pp. 422–433, May 2005.

[17] M. Jovanović and B. Bamieh, “On the ill-posedness of certain vehicular platoon control problems,” *Automatic Control, IEEE Transactions on*, vol. 50, no. 9, pp. 1307–1321, 2005.

[18] P. Barooah, P. Mehta, and J. Hespanha, “Mistuning-Based control design to improve Closed-Loop stability margin of vehicular platoons,” *Automatic Control, IEEE Transactions on*, vol. 54, no. 9, pp. 2100–2113, 2009.

[19] D. Martinec and Z. Hurák, “Vehicular platooning experiments with lego mindstorms nxt,” in *Control Applications (CCA), 2011 IEEE International Conference on*, pp. 927–932, Sept. 2011.

[20] Freescale, “Freescale race challenge.” ”`http://www.freescale.com/webapp/sps/site/overview.jsp?code=ROMANIA_RACE_CHALLENGE`”, 2004 - 2012.

[21] Digi, “Digi international.” ”`http://www.digi.com`”, 1996-2012.

[22] R. Livingston, "Slot car news motor list." "`http://slotcarnews.blogspot.com/2007/02/slot-car-news-motor-list.html`", Feb. 2007.

[23] Z. Hurák and M. Šebek, "2D polynomial approach to stability of platoons of vehicles," in *Proceedings of the 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, vol. 2, (Centre de Congrès de L'Impérial Palace, Annecy, France), 2010.

[24] M. Šebek and Z. Hurák, "2-D polynomial approach to control of leader following vehicular platoons," in *18th World Congress of the International Federation of Automatic Control (IFAC)*, (Milano, Italy), IFAC, 2011.

[25] E. Kamen, "Stabilization of Linear Spatially-Distributed Continuous-Time and Discrete-Time systems," in *Multidimensional Systems Theory: Progress, Directions and Open Problems in Multidimensional Systems*, Mathematics and Its Applications, D. Reidel Publishing Company, 1985.

[26] M. Šebek, "On 2-d pole placement," *Automatic Control, IEEE Transactions on*, vol. 30, pp. 819 – 822, aug 1985.

[27] M. Šebek, "Polynomial solution of 2D Kalman-Bucy filtering problem," *Automatic Control, IEEE Transactions on*, vol. 37, no. 10, pp. 1530–1533, 1992.

Appendix A

$\mathcal{LZ}_1$ transform

The properties of the $\mathcal{LZ}_1$ are taken from [24].

The joint unilateral Laplace and unilateral z-transform is defined as

$$\mathcal{LZ}_1\{f(t,i)\} = \int_{0^-}^{\infty} \left(\sum_{k=1}^{\infty} f(t,k) z^{-k} \right) e^{-st} dt = f(s,z), \tag{A.1}$$

which is different from the common z-transform definition since it begins with the car indexed by $k = 1$. Therefore, the description of the leader's movement (indexed by $k = 0$) is not included in the $\mathcal{LZ}_1$ transform and appears as a boundary condition.

Some z-transform properties:

$$\mathcal{LZ}_1\{\frac{\partial f}{\partial t}\} = sf(s,z) - f_{0^-}(z), \tag{A.2}$$

$$\mathcal{LZ}_1\{\frac{\partial^2 f}{\partial t^2}\} = s^2 f(s,z) - sf_{0^-}(z) - \dot{f}_{0^-}(z) \tag{A.3}$$

assuming that the derivatives exist, and

$$f_{0^-}(z) = \sum_{k=1}^{\infty} f(0^-,k) z^{-k} \tag{A.4}$$

$$\dot{f}_{0^-}(z) = \sum_{k=1}^{\infty} \dot{f}(0^-,k) z^{-k} \tag{A.5}$$

are $\mathcal{Z}_1$-transforms of the spatial sequences of initial conditions.

Another property is

$$\mathcal{LZ}_1\{f(t,k-1)\} = z^{-1} f(s,z) + z^{-1} f_0(s), \tag{A.6}$$

where

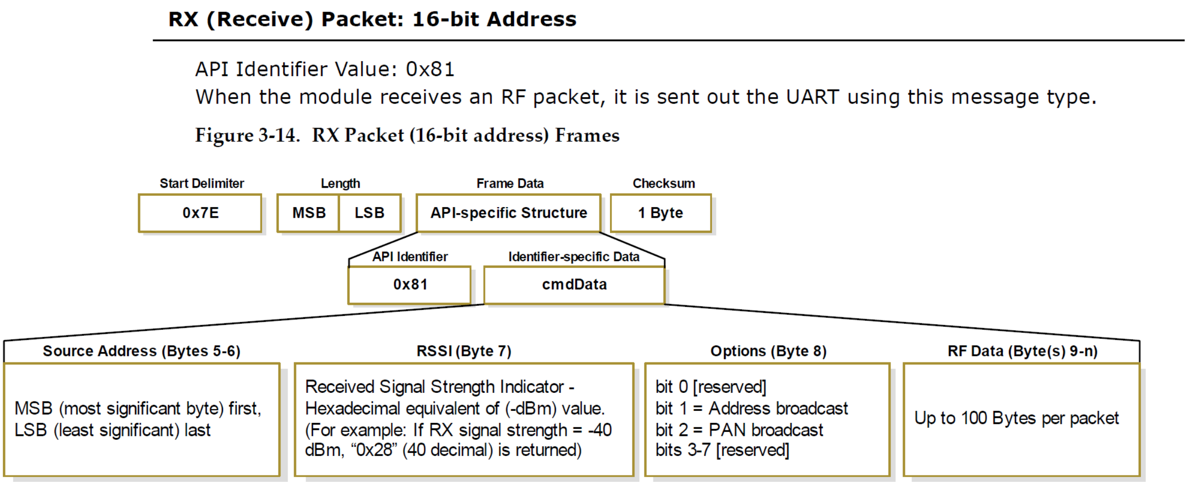
$$f_0(s) = \int_{0^-}^{\infty} f(t,0)e^{-st}dt \tag{A.7}$$

is the $\mathcal{L}$-transform of the function of the leader.

A more thorough analysis of $2D$ polynomial equations are given in [26] and [27].

Appendix B

XBee packets and PCB schematic



RX (Receive) Packet: 16-bit Address

API Identifier Value: 0x81

When the module receives an RF packet, it is sent out the UART using this message type.

Figure 3-14. RX Packet (16-bit address) Frames

Start Delimiter: 0x7E

Length: MSB | LSB

Frame Data: API-specific Structure

Checksum: 1 Byte

API Identifier: 0x81

Identifier-specific Data: cmdData

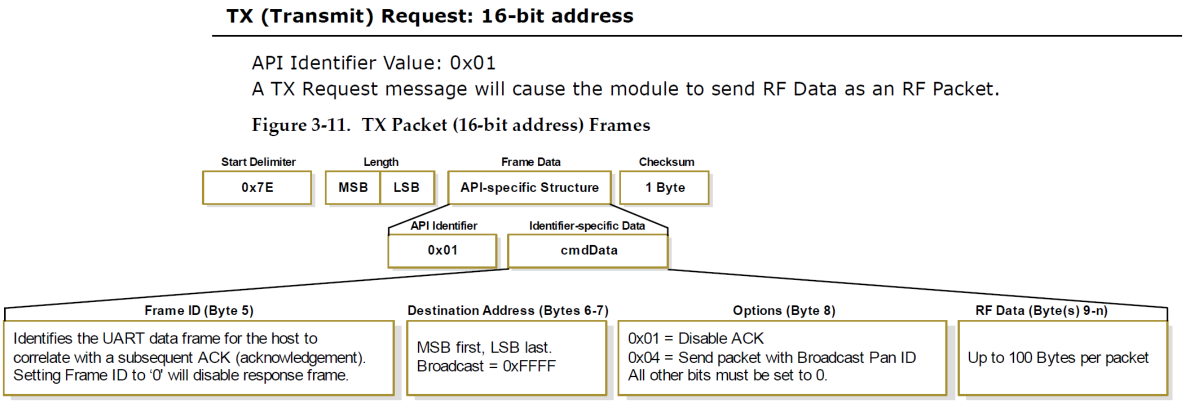
Source Address (Bytes 5-6): MSB (most significant byte) first, LSB (least significant) last

RSSI (Byte 7): Received Signal Strength Indicator - Hexadecimal equivalent of (-dBm) value. (For example: If RX signal strength = -40 dBm, “0x28” (40 decimal) is returned)

Options (Byte 8): bit 0 [reserved]
bit 1 = Address broadcast
bit 2 = PAN broadcast
bits 3-7 [reserved]

RF Data (Byte(s) 9-n): Up to 100 Bytes per packet

Figure B.1: Structure of a RX packet from documentation available at Digi website



TX (Transmit) Request: 16-bit address

API Identifier Value: 0x01

A TX Request message will cause the module to send RF Data as an RF Packet.

Figure 3-11. TX Packet (16-bit address) Frames

Start Delimiter: 0x7E

Length: MSB | LSB

Frame Data: API-specific Structure

Checksum: 1 Byte

API Identifier: 0x01

Identifier-specific Data: cmdData

Frame ID (Byte 5): Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). Setting Frame ID to ‘0' will disable response frame.

Destination Address (Bytes 6-7): MSB first, LSB last. Broadcast = 0xFFFF

Options (Byte 8): 0x01 = Disable ACK
0x04 = Send packet with Broadcast Pan ID
All other bits must be set to 0.

RF Data (Byte(s) 9-n): Up to 100 Bytes per packet

Figure B.2: Structure of a TX packet from documentation available at Digi website

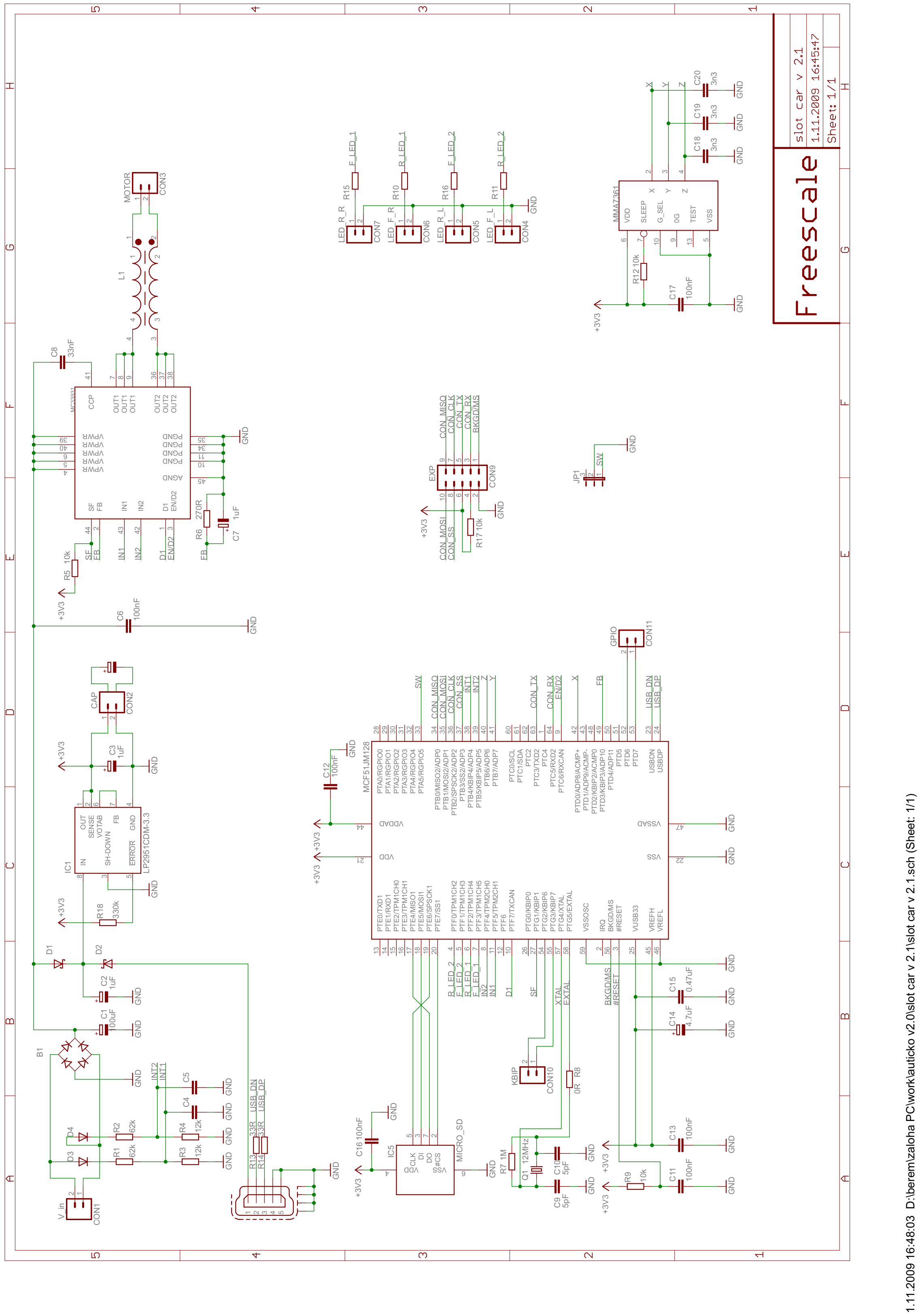


Figure B.3: Electronic scheme of the PCB designed by Freescale Semiconductor