BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING
FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS
ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

SEGWAY DRIVER PARAMETER ESTIMATION AND ITS USE FOR OPTIMIZING THE CONTROL ALGORITHM

ODHAD PARAMETRŮ JEZDCE NA VOZÍTKU SEGWAY A JEJICH POUŽITÍ PRO OPTIMALIZACI ŘÍDÍCÍHO ALGORITMU

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR
AUTOR PRÁCE

Bc. Barnabás Dobossy

SUPERVISOR
VEDOUCÍ PRÁCE

Ing. Martin Brablc

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání diplomové práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	**Bc. Barnabás Dobossy**
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Mechatronika
Vedoucí práce:	**Ing. Martin Brablc**
Akademický rok:	2018/19

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Odhad parametrů jezdce na vozítku segway a jejich použití pro optimalizaci řídícího algoritmu

Stručná charakteristika problematiky úkolu:

Cílem práce je vyvinout, otestovat a implementovat v praxi použitelnou metodu pro odhad parametrů jezdce na osobním balančním vozítku segway, která doplní standardní řídící algoritmus a umožní tak jeho adaptivitu na hmotnost a případně i výšku jezdce. Smyslem práce je ověřit myšlenku adaptace na různé jezdce a cílem dosažení lepší stability především pro začínající.
Práce navazuje na studentské projekty dokončené v letech 2011 a 2015.

Cíle diplomové práce:

1) Proveďte rešerši standardních i adaptivních řídících algoritmů vhodných pro řízení systémů podobných vozítku segway. Dále proveďte rešerši vhodných algoritmů pro odhad parametrů online i offline.

2) Vytvořte několik variant simulačních modelů soustavy a otestujte na nich vybrané algoritmy pro řízení, stabilizaci a odhad parametrů. Proveďte odhad ostatních parametrů vozítka.

3) Rozšiřte vybraný řídící algoritmus tak aby byl adaptivní a simulačně jej otestujte.

4) Přizpůsobte signálovou i výkonovou elektroniku vozítka vybraným algoritmům

5) Implementujte tyto vybrané řídící, stabilizační a estimační algoritmy pro řídící jednotku vozítka a otestujte jejich funkčnost. Při finální implementaci dbejte především na bezpečnost používání.

Seznam doporučené literatury:

NELLES, Oliver. Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Berlin: Springer, 2011. ISBN 978-364-2086-748.

LJUNG, Lennart. System identification: theory for the user. 2nd ed. Upper Saddle River, NJ: Prentice Hall PTR, 1999. ISBN 978-0136566953.

VALÁŠEK, Michael. Mechatronika. Dot. 1. vyd. Praha: České vysoké učení technické, 1996. ISBN 80-010-1276-X.

NOSKIEVIČ, Petr. Modelování a identifikace systémů. Ostrava: Montanex, 1999. ISBN 80-722-50-0-2.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc. ředitel ústavu	doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

Abstrakt

Táto práca sa zaoberá vývojom, testovaním a implementáciou adaptívneho riadiaceho systému pre dvojkolesové samobalancujúce vozidlo. Adaptácia parametrov vozidla sa uskutoční na základe parametrov vodiča. Parametre sústavy sa nemerajú priamo, ale sú odhadované na základe priebehu stavových premenných a odozvy sústavy. Medzi odhadované parametre patrí hmotnosť a poloha ťažistka vodiča. Cieľom práce je zabezpečiť adaptáciu jazdných vlastností vozidla k rôzným vodičom s rôznou hmotnosťou, kvôli zlepšeniu stability vozidla. Táto práca je pokračováním predchádzajúcich projektov z roku 2011 a 2015.

Summary

This thesis deals with development, verification and implementation of an adaptive control system on a two-wheeled self-balancing vehicle that based on the driver's parameters alters its behaviour. The parameters are not obtained by direct measurement but estimated based on the evolution of state variables and the system's response. The estimated parameters include driver's mass and height of his or her centre of gravity. The goal of this thesis is to verify the idea of an adaptation of the system's dynamical properties to different users while ensuring better stability. This thesis is the continuation of earlier projects finished in 2011 and 2015.

Klúčové slová

adaptivný regulátor, dvojkolesové samobalancujúce vozidlo, odhad parametrov, návrh DPS, SIL

Keywords

adaptive controller, two-wheeled self-balancing vehicle, parameter estimation, PCB design, SIL

Bibliographic citation

DOBOSSY , B. *Segway driver parameter estimation and its use for optimizing the control algorithm.* Brno: Brno University of Technology, Faculty of mechanical engineering, 2019. 78 pages, Master's thesis supervisor: Ing. Martin Brablc.

Rozšírený abstrakt

Úvod

Vozidlo s názvom Hummer je dvojkolové samobalancujúce vozidlo, ktoré vzniklo v rámci dvoch diplomových prác na Fakulte Strojného Inžinierstva VUT v roku 2011. V roku 2015 vozidlo prešlo výraznou obmenou. V rámci tejto obmeny bola nahradená celá elektronika, senzorická sústava a bol navrhnutý nový riadiaci algoritmus, ktorý obsahoval aj bezpečnostné prvky na detekciu chýb v elektronike.

Táto diplomová práca nadväzuje na predchádzajúce práce. Vytýčil sa nový cieľ a to implementácia nového adaptívneho riadiaceho algoritmu, ktorý bude schopný zaistiť zlepšenú stabilizáciu vozidla a zvýšenie bezpečnosti vodiča. Adaptácia parametrov vozidla sa uskutoční na základe parametrov vodiča (hmotnosť a výška ťažiska), ktoré budú odhadované na základe priebehu stavových premenných a dynamického chovania sústavy.

Podstata celej problematiky spočíva v tom, že dynamika sústavy skladajúca sa z vozidla a vodiča nie je stála, ale mení sa. Táto zmena je spôsobená zmenou vodičov a ich fyzikálnych vlastností (hmotnosť a výška). Pri návrhu regulátoru pre riadenie sústavy musíme brať do úvahy aj vyššie uvedené zmeny parametrov. Pri samotnom návrhu regulátoru si môžeme vybrať z nasledujúcich dvoch možností:

- navrhnúť regulátor pre špecifický pracovný bod
- navrhnúť regulátor pre celý interval pracovných bodov

Z praktického hľadiska ani jedna z týchto možnosti nie je ideálna . Nevýhoda regulátoru navrhnutého pre špecifický pracovný bod (napr. pre vodič s hmotnosťou 90kg) spočíva v tom, že regulátor funguje dobre v okolí pracovného bodu, ale mimo tento bod (napr. väčší rozdiel v hmotnosti vodiča) už jeho účinnosť výrazne degraduje.

Druhá možnosť ponúka čiastočné riešenie problému s tým, že sa nenavrhuje regulátor iba pre jeden bod, ale pre celú množinu takýchto bodov za cenu zhoršenia celkovej regulačnej schopnosti.

Adaptívny regulátor kombinuje výhody oboch predchádzajúcich riešení. Základná myšlienka regulátoru spočíva v tom, že máme viac regulátorov každý navrhnutý na iný pracovný bod a prepínaním medzi nimi sa zaisťuje kvalitná regulácia v ktoromkoľvek pracovnom bode. Prepínanie sa uskutoční na základe parametrov, ktoré sú schopné zaistiť popis zmeny dynamiky sústavy.

V nasledujúcej časti bude prezentované vybrané riešenie problému.

Popis riešenia

Vybrané riešenie bolo rozdelené na tri časti na základe vykonanej úlohy, ktoré sú nasledujúce: odhad parametrov, stavový automat, ktorý na základe získaných parametrov nastavuje zosilnenie regulátoru v spätnej väzbe a regulátor zaisťujúci stabilizáciu vozidla.

Odhad parametrov vozidla bola vyriešená implementáciou Newton-Raphsonovej metódy, ktorá je gradientná optimalizačná metóda. Pred implementáciou bol algoritmus simulačne otestovaný na umele vytvorených dátach získaných z modelu sústavy. V rámci testov sa

zistilo, že kvôli nemodelovateľným javom súvisiacich s riadiacimi zásahmi vodiča, odhad parametrov bude fungovať správne iba v prípade, že budú zaistené špeciálne podmienky. Tieto špeciálne podmienky môžu byť zaistené v prípade, že hodnota uhlového zrýchlenia uhlu náklonu vozidla sa bude pohybovať v rámci definovaného intervalu. Toto riešenie bolo vyskúšané aj na reálnom vozidle, kde bola overená jeho účinnosť a presnosť pri odhadovaní hmotnosti vodiča. Odhad polohy ťažiska vodiča sa nedala realizovať dostatočne presne na to, aby sa údaje dali využiť pri nastavovaní regulátoru.

Regulátor pre stabilizáciu vozidla sa skladá z podriadenej momentovej regulačnej slučky a z nadriadenej polohovej slučky. Momentová slučka, ktorá bola implementovaná v mikropočítači na výkonovej doske plošných spojov s H-mostom bola realizovaná ako regulátor typu PI. Polohový regulátor bol realizovaný adaptívnym stavovým regulátorom.

Adaptácia regulátoru bola realizovaná pomocou regulačnej štruktúry pripomínajúci stavový automat na základe hmotnosti vodiča. Boli vytvorené tri hmotnostné skupiny, každý disponujúci vlastným spätno-väzobným regulačným zosilnením. Zmienené hmotnostné skupiny boli nasledujúce:

- od 0 do 50 kg
- od 50 do 75kg
- od 75 do 120 kg

Realizácia takéhoto zložitého regulačného algoritmu požadovalo aj vylepšenie hardwarovej výbavy vozidla.

V rámci obnovy bol pridaný kapacitný enkodér za účelom merania polohy natočenia kolies. Pridanie enkodéru bolo nutné preto, lebo údaj z polohy enkodéru sa stal jedným zo vstupov novo pridaného algoritmu na odhad parametrov. Okrem priradenia enkodéru sa obnovila aj výkonová a signálová elektronika vozidla .

V rámci diplomovej práce boli vytvorené nasledujúce tri dosky plošných spojov:

- **výkonová DPS** s integrovaným H-mostom , meraním prúdu a mikropočítačom slúžiaci ako regulátor prúdu
- **signálová DPS** , ktoré zaisťuje vzájomné dátové prepojenie jednotlivých senzorov a riadiacich jednotiek
- **DPS na galvanické oddelenie riadiaceho signálu**

Zhrnutie a zhodnotenie výsledkov

V rámci tejto diplomovej práci bol vytvorený adaptívny regulátor pre riadenie dvojkolového samobalancujúceho vozidla s prezývkou Hummer. Potrebné softwarové riešenie sa skladalo z troch funkčných časti, ktoré boli nasledujúce:

- algoritmus pre odhad parametrov
- stavový automat pre zaistenie adaptáciu regulátoru
- samotný regulátor na stabilizáciu vozidla

Algoritmus pre odhad parametrov pracoval s presnosťou ±10 % v prípade odhadu hmotnosti vodiča, presnosť odhadu polohy ťažiska bola veľmi nízka, preto sa tento parameter nepoužil pre nastavenia regulátoru. Kvôli tomu, že boli pridané tri rôzne nastavenie regulátoru, každé pre jeden hmotnostný interval, sa výrazne zlepšila stabilita a celková ovládateľnosť vozidla.

Súčasťou práce bolo aj obnovenie a rozšírenie elektroniky a senzorickej sústavy vozidla. Boli navrhnuté a vyhotovené tri dosky plošných spojov a bol pridaný aj enkodér na meranie uhlu natočenia kolies.

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This master's thesis is my own work and contains nothing which is the outcome of work done in collaboration with others.

Barnabás Dobossy

Brno . . . .

First of all, I would like to express my sincere gratitude to my advisor Ing. Martin Brablc for his insightful comments, patience, motivation, and immense knowledge. I also would like to thank all of the members of the Mechatronics Laboratory for their help and for creating supportive environment. Last but not the least, I would like to thank my family for supporting me throughout writing this thesis and my life in general. This accomplishment would not have been possible without them. Thank you.

Barnabás Dobossy

Contents

List of Figures

List of Tables

1 Introduction

Nowadays traffic has become one of the biggest issues that challenges us on a daily bases, with constant traffic congestions and accidents making it a nightmare to commute to work, school, etc., or just to travel. These problems will be even more challenging in the future as the number of cars on the roads is supposed to grow by 36% by 2025 according to an article from the World Economic Forum [1]. Such issues give incentives to inventors and engineers to introduce new alternative ways of transportation that can solve the challenges of the future. Among the newest alternatives is Elon Musk's Hawthorne tunnel [2] for mid-range travel through Los Angeles, or the Segway Personal Transportations.

The Segway is a two-wheeled self-balancing vehicle invented by Dean Kamen in 1999. Since then it has risen to worldwide fame and inspired many universities and researchers to build their own version of this unique vehicle.

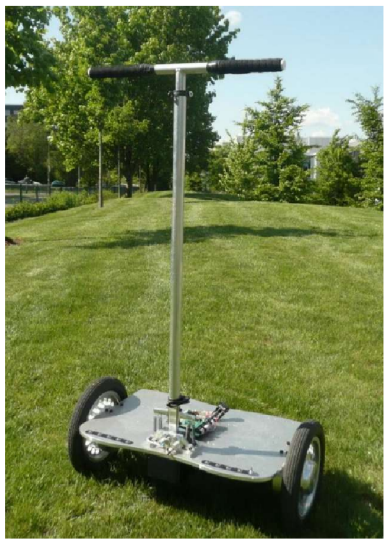


Figure 1.1: Hummer in 2011 [4]

Faculty of Mechanical Engineering of Brno University of Technology built its own version from scratch in 2011 as part of two master's theses and named it Hummer [3], [4]. In 2015 the Hummer received a complete overhaul. This included complete redesign of the original electronics, the control algorithm and it received safety features to detect failures of the sensors and the controller units [5], [6].

This thesis is the continuation of the above-mentioned work, while implementing a new control algorithm to make improvements in the fields of driving experience, stability and user safety. To achieve these demanding objectives a personalised solution will be used. The parameters of the driver's physical traits such as weight, height will be taken into consideration by a controller design approach called adaptive control. Adaptive control accommodates its behaviour to changes in the system's parameters, which will not be measured directly but obtained indirectly from the system's dynamical behaviour. The

utilized techniques are referred to as parameter estimation.

1.1 Motivation

By carefully studying the problematics presented in this thesis the following conclusion can be made. The two-wheeled self-balancing vehicle and its driver form a dynamical system whose parameters are varying. This is due to the fact that different driver's have different physical properties, such as body mass or height, which significantly affects the system's overall dynamical behaviour. Therefore, when designing a controller the changes in the system's dynamics can not be neglected. As solution to this problem the designer of the controller has two options:

1. design a regulator for a specific operating point
2. design a regulator for a whole interval of different operating points.

In general non of these alternatives are ideal when having just a single controller. In case of the first option the controller's performance in close vicinity of the specific operating point is exceptional; however, outside this area its performance degrades. The farther the system is operated from the working point the more its performance decreases. To put it simply, the controller, which was designed to perform well with the two-wheeled self-balancing vehicle without a driver, is insufficient for controlling it properly when a driver with mass of 80 kg and height 190 cm is standing on it. This is due to the fact that the driver's dynamics changes the system's dynamics as whole, while it increases its mass, which creates more resistance to acceleration, and also moves the position of its centre of mass higher, which makes the stabilization even more difficult.

The second option presents a solution to this problem by sacrificing the outstanding performance around one single point for a mediocre performance on a wider interval of operating states.

The combination of these two approaches is the adaptive controller. The idea behind the adaptive controller is quite simple. It combines the advantages both of the proposed solutions. The technique utilizes multiple controllers each designed to work around a specific operating point, which in this case might be a specific value of body mass, e.g., the above-mentioned 80 kg, that way covering broader interval of operating states. As a result the controller retains exceptional behaviour over the whole interval.

The adjustment of the regulator takes place as a response to changes in the system's dynamics, therefore to be able to capture these changes, parameters of the vehicle that describe the changes in behaviour have to be monitored. There are two ways to do that:

- by direct measurement of the parameters
- by estimation of the parameters

Estimation of the parameters was chosen over their direct measurement based on the following reasons:

- there are parameters that can not be directly measured, for instance the position of the centre of mass

– reduced requirements on the used hardware, e.g., sensors for the measurement of mass would otherwise require complex hardware solution consisting of multiple strain gauges, signal conditioning circuitry and data processing.

As it was mentioned above, the estimated parameters include the body mass and position of centre of mass of the driver.

In the following chapters the development procedure of the adaptive controller will be described in details.

1.2 The initial state of the vehicle

The vehicle received its last significant rebuild in 2015 as the result of two master's theses [5], [6]. This section is devoted to briefly recapture the state as to the summer of 2018, when this project started.

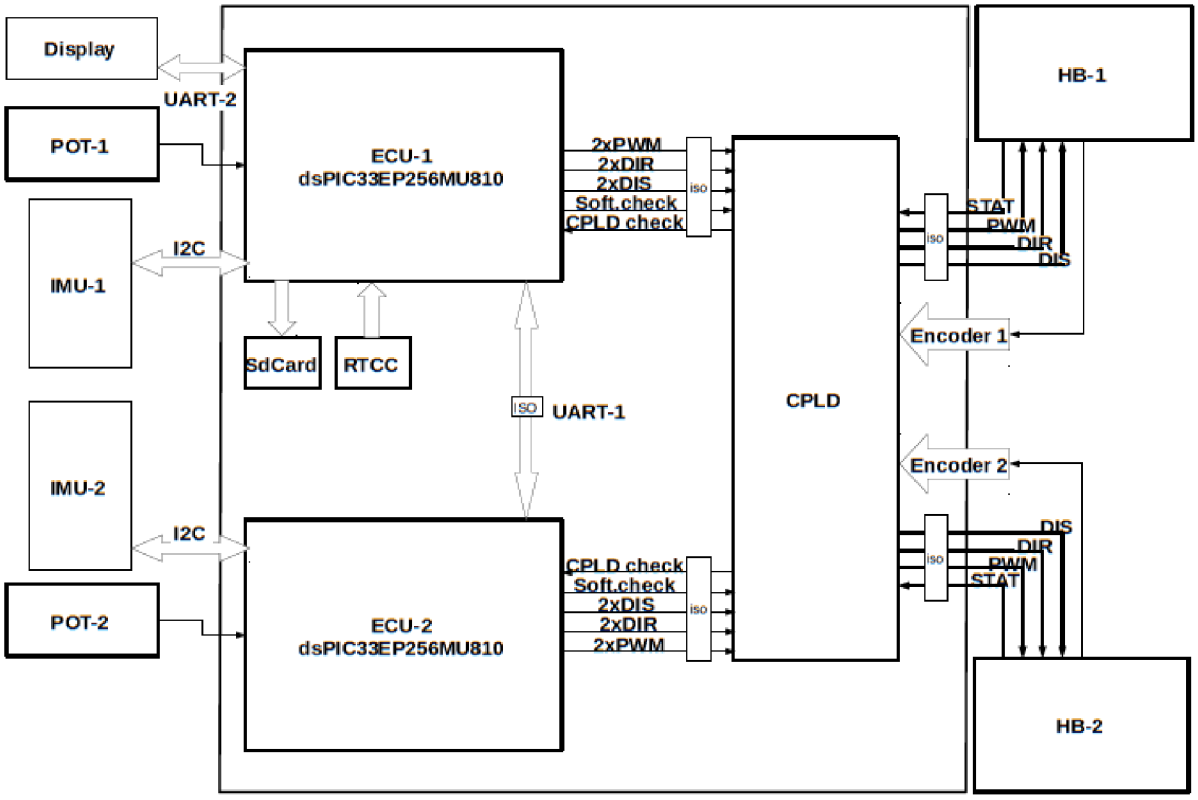


Figure 1.2: Block diagram of the control system after the rebuild in 2015 [5]

The body is made of aluminium alloy, which makes it durable while keeping the overall weight of the chassis low. The drive-train of the vehicle consists of two DC motors gr 80x80 from Dunker Motoren with nominal voltage of 40 V each of them equipped with gearbox with 9 : 1 gear ratio. The necessary energy for powering the motors is supplied from three lead-acid battery packs with nominal voltage of 12.5 V and capacity of 7.5 Ah. The batteries are connected in series which makes it 37.5 V and 22.5 Ah in total.

The vehicle is also equipped with IMU sensors for calculating the leaning angle of

the body and with potentiometer for measuring the sideways tilt of the handlebar which serves as user input in order to steer the vehicle. Complementary filter algorithm is used to calculate the leaning angle from the IMU's accelerometer and gyroscope readings. This algorithm together with the main PID controller algorithm, which is responsible for keeping the vehicle upright runs on the main ECU. For safety reasons the number of sensors and ECUs is doubled and only one of them is used for controlling the Hummer at a time. To switch between the two units a Xillinx XC95144 CPLD is used. The Figure 1.2 presents the layout of the hardware in its original state at the start of the project.

2 Research Study

This chapter presents the theoretical foundation of this work, as it will introduce and briefly explain some of the concepts from control system design and system identification theory that can be of practical use for our real vehicle.

The chapter can thematically be divided into two parts. The first describes some conventional and adaptive control algorithms that are suitable for controlling two-wheeled inverted pendulum-like vehicles. The second part examines methods appropriate for on-line and/or off-line parameter estimation.

2.1 Conventional Control Algorithms

The Hummer and similar systems are commonly referred to as two-wheeled inverted pendulums. The phrase inverted refers to the fact that in their equilibrium position the rod stands vertically. This equilibrium point is unstable, which means that the rod falls over unless a force is applied to the system. To find the precise amount of force that has to be exerted to keep the leaning angle of the rod zero a control algorithm needs to be applied.

When choosing the control algorithm there are a few aspects that have to be taken into consideration. One of them is the relationship between the input and output quantities of the system.

This vehicle is basically a non-linear system, where the non-linearities come from:

- relationship between the position of the point mass which represents the mass of the rod and chassis and the leaning angle.
- the Coulomb friction between the wheels and the surface of the road and also between the axles.

However, by setting some limitations, the system can be linearised around a specific working point by the use of the first order Taylor polynomial. These limitations are the following:

- the leaning angle is small and set around the upright position of the rod
- the Coulomb friction between the wheels and the surface of the road and also between the axles is negligible.

As a result of that, to stabilise the rod in the vertical position linear control algorithms can be used. The following part of this section is designated to make the reader acquainted with these techniques.

2.1.1 PID controller

The Proportinal-Integral-Derivative controller is a versatile control algorithm, which thanks to its robustness and functional simplicity became the most widely used control algorithm in industrial applications. According to [9] "*Based on a survey of over eleven thousand controllers in the refining, chemical and pulp and paper industry, 97% of regulatory controllers utilize PID feedback* ".

This method has some significant benefits. First of all it is a control method for linear systems, also capable of dealing with non-linear systems with slight non-linearities. Secondly it presents a black-box approach for control system design, therefore, it is convenient and easy to implement on any system.

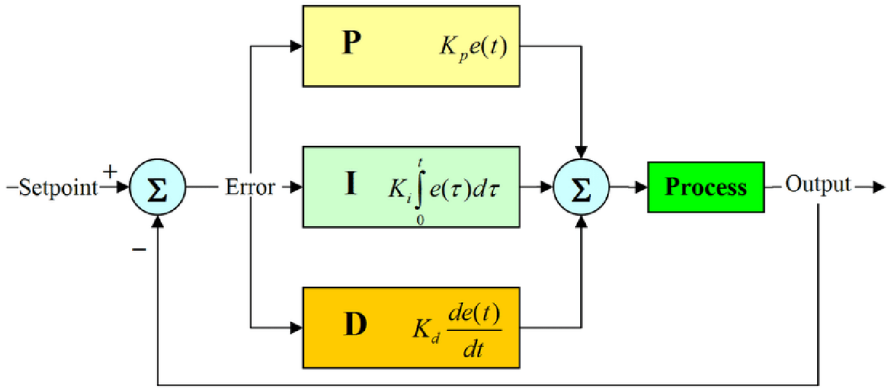


Figure 2.1: Basic block diagram of PID controller [10]

In the past years many master's theses and articles have been published on successful practical implementation of this control approach on two-wheeled inverted pendulum-like vehicles [3], [6], [11], [12]. According to [11] the PID controller had faster rise-time and smaller overshoot compared to LQR controller as a reaction to step-like change in the input.

As PID controller is considered as a well known control algorithm, therefore more detailed description is out of the scope of this thesis, for more information the reader may visit the following websites [7], [8].

2.1.2 Modern Control Approach [13]

Contrary to the previous method the following section presents a distinct approach to control system design, which is referred to as [13] "modern control theory".

Techniques described in this section represent the knowledge about the system's internal structure by the use of the LTI state space model, which is a set of n first order differential equations, where the number n is the order of the system. To simplify the work with these equations vector matrix notation is used.

$$\begin{aligned} \dot{x} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{aligned} \tag{2.1}$$

where $\dot{\mathbf{x}}$ = state vector
$\mathbf{y}$ = output vector
$\mathbf{A}$ = state matrix
$\mathbf{B}$ = input matrix
$\mathbf{C}$ = output matrix
$\mathbf{D}$ = feed-through matrix

In dynamic system analysis the solution of the homogeneous part or in mathematical terms the eigenvalues of the differential equation have significant importance as they carry useful information regarding the system's stability and time-domain behaviour. From control system design perspective to ensure system stability the eigenvalues of the system should have negative real part and their position on the negative axis determines the system's dynamics.

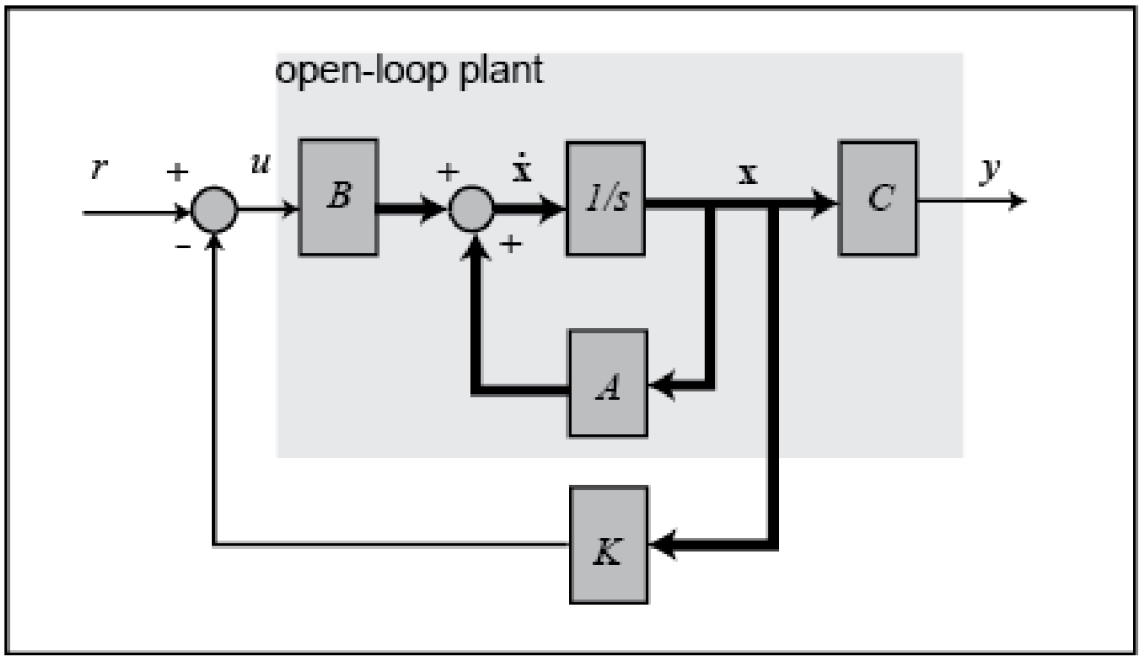


Figure 2.2: Block diagram full state feedback system [14]

All of the following algorithms are based on altering the system's eigenvalues to match the desired response by choosing the gains of the state-feedback gain matrix. The only major difference between them is in the way, how the gain matrix is obtained. The modified state-feedback equations with the gain matrix are the following:

$$\begin{aligned} \dot{x} &= (\boldsymbol{A} - \boldsymbol{BK})\boldsymbol{x} + \boldsymbol{Bu} \\ \boldsymbol{y} &= \boldsymbol{Cx} \end{aligned} \tag{2.2}$$

where $\mathbf{K}$ = state-feedback gain matrix.

These methods can only be applied if the state space model meets the requirement of *controllability*. According to [15]: "*The system is controllable if there exists an input sequence that can transfer the system from the initial state x_0 to any arbitrary state x_k in finite-time.*"

Ackermann's Formula [13]

The Ackermann's formula is a commonly used practical implementation of the *pole placement method*. Pole placement is a novel concept introduced by modern control theory. The basic idea behind this method is the following: the designer of the controller has access to all of the poles of the system, not just to the dominant ones (classical control theory) and can move them to arbitrary position to meet the desired control goals. One of the biggest advantages of this method is the analytical expression for calculating the state-feedback gain matrix.

$$\boldsymbol{K} = [0 \quad 0 \quad \cdots \quad 0 \quad 1][\boldsymbol{B} \quad \boldsymbol{AB} \quad \cdots \quad \boldsymbol{A}^{(n-1)}\boldsymbol{B}]^{-1}\phi(\boldsymbol{A}) \tag{2.3}$$

where $\phi(\mathbf{A})$ = desired characteristic polynomial evaluated at matrix $\mathbf{A}$

In Matlab the function *place()* is used to calculate the state-feedback gain matrix for the desired placement of poles. The function utilizes the Ackermann's formula.

Linear Quadratic Regulator [13]

Linear Quadratic Regulator or LQR is another method for obtaining the elements of the state-feedback gain matrix. In this case the calculated gain matrix is optimal as it was acquired by minimizing a quadratic performance index.

$$J = \int_0^\infty (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})dt \tag{2.4}$$

where $\mathbf{Q}$ = weight matrix of the states
$\mathbf{R}$ = weight matrix of the inputs

Matrices $\mathbf{Q}$ and $\mathbf{R}$ make it possible to set the weights individually for each state and input, respectively. The higher the weight the more precisely the specific state follows the reference input. Both of these matrices should be positive definite Hermitian matrices. An advantage of state-feedback gain matrix designed by this method is that the system will be always stable, unless the controllability condition is not met.

The minimization problem of the quadratic performance index leads to the solution of algebraic Riccatti equation, which has no closed form solution. The two most commonly used numerical methods to solve this problem are the Schur Decomposition and the Sign Function Method. The Matlab command *lqr()* utilizes the Schur method.

The applicability of the LQR design technique on two-wheeled self-balancing vehicles was verified in the article [16], where the author successfully applied it to stabilize and control a two-wheeled self-balancing robot.

2.2 Adaptive Control Algorithms

The idea of Adaptive Control comes from the 1950's, when American engineers tried to refine aeroplane's flight control by deploying autopilot.[18]

Adaptive Control Algorithms are based on the simple idea that the output vector *y(t)* carries information about the states *x(t)* and parameters of the system and by learning

to interpret the output signal the change in the system's parameters can be acquired. Finally, based on the newly obtained parameters the feedback gains can be updated [17].

As can be seen from Figure 2.3 the basic structure of the adaptive controller does not differ much from the one introduced in the previous chapter, except the fact that the feedback gains in this case are changing while the system is working. There exists a handful of adaptive control methods the only difference is in the way of devising update strategy for the controller's gain vector in response to changes in parameters, also referred to as the (*control law*) [17].

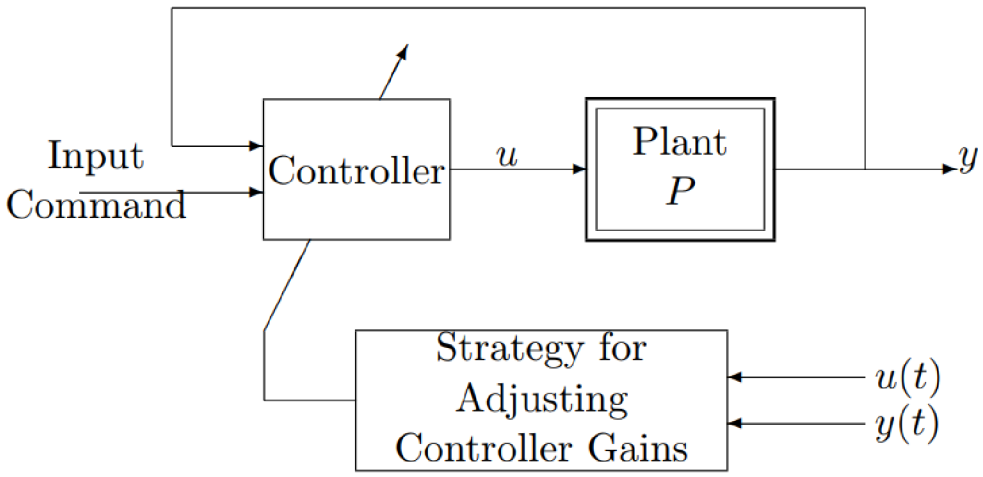


Figure 2.3: Block diagram of Adaptive Control scheme [17]

A common approach to classify adaptive algorithms is based on the way the on-line parameter estimator (*adaptive law*) is used with the control law [17]:

- *Indirect Adaptive Control*: first the new parameters are estimated on-line by the use of the plant's model, followed by the update of the controller
- *Direct Adaptive Control*: parameter estimation and regulator calculation is done simultaneously within the same step.

In the following adaptive control methods will be introduced with brief explanation of their working principle.

2.2.1 Gain Scheduling [17]

In this case the controller works with a precomputed set of constant gains implemented as a look-up table. The gains are calculated off-line on the linearised model of the plant. This method works as a state machine, change from one operating point represented with set of feedback gains to another operating point happens as a response to measurement of state variables. According to [17], in case of an aeroplane the measured variables can be "*the Mach number and the dynamic pressure*" .

When there is no direct match between the measurement and operating points, interpolation is used to compute the new gains.

Among the advantages of this method are quick gain transition, as it uses no adaptive or control law. This method has its own drawback, too. Frequent gain changes may lead to unstable behaviour. Therefore, some sort of hysteresis has to be implemented to limit the frequency of gain changes.

Among the possible fields of applications of this method are aeronautics [19], automation and automotive [20].

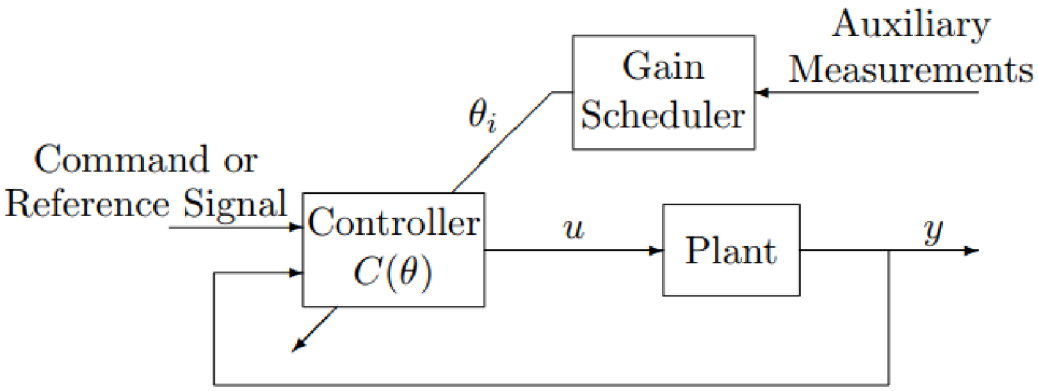


Figure 2.4: Block diagram of Gain Scheduling control scheme [17]

2.2.2 Model Reference Adaptive Control [17]

Model Reference Adaptive Control or *MRAC* is one of the most commonly used techniques when it comes to adaptive controller design.

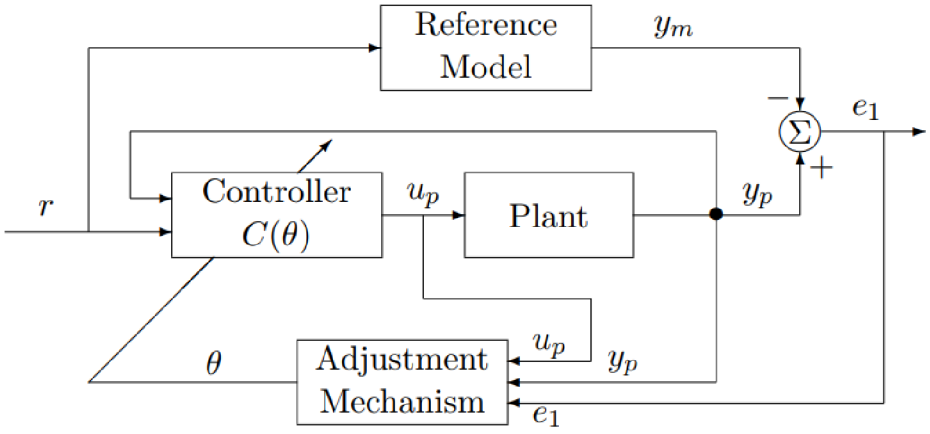


Figure 2.5: Block diagram of MRAC scheme [17]

A model of the plant is used to generate reference output based on the desired input and the system's parameters. This reference output is then used to compute the tracking error, which is the difference between the plant's output and the reference output.

$$e = y_P - y_M \tag{2.5}$$

where e = tracking error
y_P = plant's output
y_M = model reference output

The adjustment mechanism sets the controller parameters so, that the tracking error is minimized, i.e., the plant's output follows the reference output.

There are different available approaches to design the adjustment mechanism. For instance: *the MIT Rule* or the *SPR-Lyapunov Design Approach*.

The *MIT rule* is a gradient descend method, which uses a quadratic cost function with the parameter θ :

$$J(\theta) = \frac{e^2}{2} \tag{2.6}$$

The control law is derived by applying the gradient rule and the product rule on the above introduced cost function, which results in the subsequent equation:

$$\frac{\partial \theta}{\partial t} = -\lambda \frac{\partial J}{\partial t} = -\lambda e \frac{\partial \theta}{\partial t} \tag{2.7}$$

where λ is a positive constant, which sets the speed of convergence, in literature it is often referred to as the *adaptation gain of the controller*. The negative gradient ensures that the tracking error decreases to zero [21].

The *SPR-Lyapunov* approach on the other hand uses a strictly positive real (*SPR*) transfer function to describe the tracking error. Then a Lyapunov function is chosen based on some criteria, which are defined by the *Meyer-Kalman-Yakubovich* and the *Kalman-Yakubovich-Popov* lemmas. The Lyapunov function's time derivative is used to derive the adaptive law. An important property of this time derivative is its non-positive nature, which is used to minimize the tracking error.[17]

The MRAC schemes can be *direct* or *indirect*. The direct MRAC as the name suggests sets the controller parameter directly by the use of the adaptive law, while the indirect techniques calculate the new controller gain values by solving an algebraic equation which links the gains with the estimated system parameters.[17]

2.2.3 Adaptive Pole Placement Control [17]

Adaptive Pole Placement Control or *APPC* presents a group of methods that is applicable on wide range of control problems, with common feature that they do not involve cancellation of poles and zeros to adjust the poles of the plant. It utilizes the combination of pole placement and parameter estimation algorithms.

Direct APPC schemes generate the control law without any intermediate steps. Their application is restricted only to special control problems and they are more complex compared to *indirect APPC schemes*.

Indirect APPC schemes on the other hand, first employ algorithms for on-line parameter estimation and based on the resulting coefficients the controller is updated. Enormous advantage of the indirect scheme is in the separation of the two design steps. Therefore the designer can independently choose the applied methodology to solve the parameter

estimation and controller design problem.

Among the available controller design techniques are the already introduced pole placement and LQR. The adaptive law is commonly realized by least-squares or gradient descend techniques, which will be explained in more details in the following section.

2.3 Algorithms for Parameter Estimation

According to [22] "*Parameter estimation is a technique for determining the best value of certain model parameters through data assimilation or similar techniques* "

The use of parameter estimation improves the accuracy of the used models, therefore, their response more accurately approximates the behaviour of the original system. It plays a critical role in today's engineering practice, where the *Model-Based Design*, or*MBD* is a widely used approach to address complex problems from control system design to signal processing.

In the following, basic parameter estimation techniques will be introduced that could be of practical use in determining the parameters of our two-wheeled inverted pendulum-like vehicle. In the choice of the algorithms the following criteria were considered:

- adaptability on linear and non-linear systems
- ease of implementation
- fast convergence, the number of iteration steps required to be as low as possible (it is related to the time constraints and computational requirements of the algorithm)
- adaptability of convergence rate (safeguard to ensure convergence even for poorly conditioned problems)
- feasibility and availability of boundary conditions

2.3.1 Gradient Methods [24]

Gradient-based methods are among the most commonly used estimation techniques. As it was mentioned before the goal is to find the optimal value of some chosen parameters θ_k by iterative minimization of a loss function. Gradient serves as a tool to search the multi-dimensional parameter space and find the direction of steepest descend. Basic proposition of this method is that the gradient of the loss function has to be known either analytically or as a finite difference. The parameter update is given by the following formula:

$$\boldsymbol{\theta_k} = \boldsymbol{\theta_{k-1}} - \eta_{k-1}\boldsymbol{p_{k-1}} \tag{2.8}$$

with

$$\boldsymbol{p_{k-1}} = \boldsymbol{R_{k-1}}\boldsymbol{g_{k-1}} \tag{2.9}$$

where θ_{k-1} = parameter vector at *k-1* iteration
g_{k-1} = gradient of the loss function
R_{k-1} = direction matrix
η_{k-1} = step size

The difference between the existing techniques is in the setting of the coefficients η_{k-1} and R_{k-1}.

Steepest Descent [24]

The case when $\boldsymbol{R_{k-1}} = \boldsymbol{I}$ is called th *steepest descent* and it is a special case of the 2.8 equation. The new equation has the following form:

$$\boldsymbol{\theta_k} = \boldsymbol{\theta_{k-1}} - \eta_{k-1}\boldsymbol{g_{k-1}} \tag{2.10}$$

As its name suggests the method ensures that the loss function is decreasing and its value is smaller with each iteration. However it has its shortcomings. In case of a large eigenvalue spread of the Hessian matrix the optimization of the parameters will only affect the dominant eigenvalues and as a result its convergence will be slower ([24] refers to it as "zig-zagging") or it will not converge at all. Among the key advantages of this method are:

- the Hessian is not required
- ease of implementation and comprehension
- low computational complexity

Newton's Method [24]

Newton's method is another modification of the upper mentioned original gradient method, in this case the direction matrix $\boldsymbol{R_{k-1}} = \boldsymbol{H_{k-1}^{-1}}$, the resulting equation:

$$\boldsymbol{\theta_k} = \boldsymbol{\theta_{k-1}} - \eta_{k-1}\boldsymbol{H_{k-1}^{-1}}\boldsymbol{g_{k-1}} \tag{2.11}$$

where θ_{k-1} = parameter vector at *k-1* iteration
g_{k-1} = gradient of the loss function
R_{k-1} = direction matrix
η_{k-1} = step size

The usage of second order derivatives makes the parameters converge at quadratic rate. In case of linear problems the algorithm converges within one iteration, however, for non-linear problems it takes more than one iteration. Improvement can be achieved by tuning the coefficient η. The fast convergence comes at cost of increased requirements related to the use of the Hessian. One of them is the requirement on the Hessian to be positive definite, which is a necessary condition ensuring the convergence of the algorithm. The usage of the Hessian also demands increased computational resources as its computation and inversion can be also challenging, mainly for hard real-time applications (on-line parameter estimation). These deficiencies make its application limited only to small parameter estimation problems.

In the following part techniques that were designed to eliminate the shortcomings of the basic Newton's method will be presented.

Quasi-Newton Methods [24]

These algorithms are an improved version of the basic Newton's method introduced in the previous subsection as it ensures fast convergence of the solution, while it also solves almost all the problems related to the use of the second order derivatives, which are the following:

- matrix inversion
- calculation of the second order derivatives by finite difference techniques

The idea behind these methods is simple, they use approximated Hessian $\hat{H}_{k-1}$ to overcome the above-mentioned shortfalls. The solution can be outlined by the following mathematical expression:

$$\boldsymbol{\theta_k} = \boldsymbol{\theta_{k-1}} - \eta_{k-1}\hat{\boldsymbol{H}}_{\boldsymbol{k-1}}^{-1}\boldsymbol{g_{k-1}} \tag{2.12}$$

with

$$\hat{\boldsymbol{H}}_{\boldsymbol{k}} = \hat{\boldsymbol{H}}_{\boldsymbol{k-1}} + \boldsymbol{Q_{k-1}} \tag{2.13}$$

or

$$\hat{\boldsymbol{H}}_{\boldsymbol{k}}^{-}\mathbf{1} = \hat{\boldsymbol{H}}_{\boldsymbol{k-1}}^{-1} + \widetilde{\boldsymbol{Q}}_{\boldsymbol{k-1}} \tag{2.14}$$

where $\boldsymbol{Q_{k-1}}$ and $\widetilde{\boldsymbol{Q}}_{\boldsymbol{k-1}}$ are used to update the approximate Hessian at each iteration, their implementation differs from method to method. The two most commonly used are the *Broyden-Fletcher-Goldfarb-Schanno* or the *Davidon-Fletcher-Powell* formula.

The equation 2.14 is more commonly utilized in practical applications as it is a straightforward formula for obtaining the inverted Hessian. In practice identity matrix is used as the initial guess of the Hessian ($\boldsymbol{H}_0 = \boldsymbol{I}$).

Although the presented algorithm solves the matrix inversion and the finite difference problem, its shortcomings include high memory requirements and computational complexity, due to the extensive use of matrix multiplication.

Conjugate Gradient Methods [24]

The *conjugate gradient methods* contrary to the previous techniques circumvent direct approximation of the Hessian matrix at the expense of slower convergence. Nonetheless its performance in the other aspects outweighs this deficiency. Among its advantages are:

- memory requirements grow linearly with the matrix dimensions
- lower computation time
- fast convergence
- computation of the Hessian is not required

The algorithm can be described by the following mathematical expression:

$$\boldsymbol{\theta_k} = \boldsymbol{\theta_{k-1}} - \eta_{k-1}\boldsymbol{p_{k-1}} \tag{2.15}$$

with

$$\boldsymbol{p_{k-1}} = \boldsymbol{g_{k-1}} - \beta_{k-1}\boldsymbol{p_{k-2}} \tag{2.16}$$

where β is a scalar weighting factor, which serves as a measure of incorporation of knowledge from previous iterations. In case of linear problem the speed of convergence is independent of the parameter β. For finding the optimal value of this weighting parameter given a non-linear problem different approaches are used. One common choice is the formula proposed by *Fletcher* and *Reeves*.

$$\beta_{k-1} = \frac{\boldsymbol{g}_{k-1}^T\boldsymbol{g}_{k-1}}{\boldsymbol{g}_{k-2}^T\boldsymbol{g}_{k-2}} \tag{2.17}$$

One of the favourable properties of incorporating information about preceding search directions is the elimination of the "zig-zagging" effect, which is common for the steepest descent algorithm. From the perspective of implementation of past knowledge this method is a compromise between the steepest descent and the quasi-Newton techniques.

Newton-Raphson Method [25]

This method is the modification of the above-mentioned Newton's method. This approach is primarily used to solve system's of equations by finding their roots. However, this method can be tailored to solve parameter estimation problems. The general form of this method is given by:

$$\boldsymbol{\theta_k} = \boldsymbol{\theta_{k-1}} - \eta_{k-1}\boldsymbol{J_{k-1}^{-1}}\boldsymbol{f_{k-1}} \tag{2.18}$$

where $\boldsymbol{f_{k-1}}$ = system of equtions reorganized to the form $\boldsymbol{f(x)} = \boldsymbol{0}$
$\boldsymbol{J_{k-1}}$ = Jacobian of the equations $\boldsymbol{f(x)} = \boldsymbol{0}$

The advantages of this method include:

- fast convergence
- the computation of the Hessian is not required
- ease of implementation

On the other hand the method's performance can degrade when the value of the elements of the Jacobian is close to zero, as a result of that the correction term $\boldsymbol{J_{k-1}^{-1}}\boldsymbol{f_{k-1}}$ becomes large.

2.3.2 Non-linear Least Squares Problems [24]

Non-linear Least Squares, or *NLS* represent another methodology commonly utilized in parameter estimation. Contrary to the previous methods it minimizes a quadratic loss

function of the following form:

$$I(\boldsymbol{\theta}) = \sum_{i=1}^{N} q_i e^2(i, \boldsymbol{\theta}) = \sum_{i=1}^{N} f^2(i, \boldsymbol{\theta}) \tag{2.19}$$

with

$$e(i, \boldsymbol{\theta}) = y(i) - \hat{y}(i, \boldsymbol{\theta}) \tag{2.20}$$

where y_i = measured output
$\hat{y}(i, \boldsymbol{\theta})$ = model output
q_i = weighting term

Among the advantages of this particular choice of loss function are:

– all the errors add up regardless of their sign

– many small errors outweigh a few large errors

– convex shape of the state space, with one extremum

– analytical expression for gradient and Hessian computation

The gradient and the Hessian can be calculated according to the following mathematical expressions:

$$\boldsymbol{g} = 2\boldsymbol{J}^T\boldsymbol{f} \tag{2.21}$$

$$\boldsymbol{H} = 2\boldsymbol{J}^T\boldsymbol{J} + 2\boldsymbol{S} \tag{2.22}$$

with

$$\boldsymbol{S} = \sum_{i=1}^{N} f(i)\boldsymbol{T}(i) = \tag{2.23}$$

where $\boldsymbol{J}$ = Jacobian of the loss function f
$\boldsymbol{T}$ = matrix of second order partial derivatives of the loss function

Based on the inclusion or exclusion of the $\boldsymbol{S}$ term from equation 2.22, two different approaches of the NLS are distinguished.

The *small residual algorithms* neglect the $\boldsymbol{S}$ term, which means that to evaluate the Hessian only the Jacobian matrix is needed. Due to this fact the computational cost of the Hessian remains low. The *Gauss-Newton* and *Levenberg-Marquardt* are two of the most popular implementations of the approximated Hessian. These techniques will be explained in details in the upcoming part.

The second bunch of algorithms are called *large residual algorithms*, while they utilize the $\boldsymbol{S}$ term. These methods are computationally expensive as they require the computation of the matrix $\boldsymbol{T}$, therefore, they are not widely used as the NLS method would loose its edge over Newton's method.

Gauss-Newton Method [24]

It is the non-linear least squares extension of the basic Newton method. This technique is one of the representations of the small residual algorithms, therefore it neglects the $\boldsymbol{S}$ term whilst approximating the matrix of second derivatives. The mathematical formula is the following:

$$\boldsymbol{\theta_k} = \boldsymbol{\theta_{k-1}} - \eta_{k-1}(\boldsymbol{J}_{k-1}^T \boldsymbol{J}_{k-1})^{-1} \boldsymbol{J}_{k-1}^T \boldsymbol{f}_{k-1} \tag{2.24}$$

By carefully setting the coefficient η the robustness and convergence properties of this approach can be improved. The deficiencies of the Gauss-Newton method surface when the system of equations is poorly conditioned or singular. The eigenvalues of the approximated Hessian term influence the rate of convergence. The smaller they are, the slower it converges.

The fact that the inverse term of the 2.24 equation has analytical solution is a vast advantage.

Levenberg-Marquardt Method [24]

The Levenberg-Marquardt algorithm is an improved version of the previously introduced Gauss-Newton method. As it presents solution to the poorly conditioned matrix problem.

The next equation presents mathematical formula of this approach:

$$\boldsymbol{\theta_k} = \boldsymbol{\theta_{k-1}} - \eta_{k-1}(\boldsymbol{J}_{k-1}^T \boldsymbol{J}_{k-1} + \alpha_{k-1}\boldsymbol{I})^{-1} \boldsymbol{J}_{k-1}^T \boldsymbol{f}_{k-1} \tag{2.25}$$

where α_{k-1} is a scaling factor. For small values of α_{k-1} the behaviour of the method is similar to Gauss-Newton method, however for large values it resembles the steepest descent algorithm. Ideally the scaling coefficient α_{k-1} should change during the execution of this algorithm. At the beginning, the algorithm should start off with large value of α_{k-1}, which ensures that the loss function decreases by keeping the term $\boldsymbol{J}_{k-1}^T \boldsymbol{J}_{k-1} + \alpha_{k-1}\boldsymbol{I}$ positive. At each iteration this value should decrease by a small amount to make sure good convergence properties of the Gauss-Newton method in the vicinity of the optimum are used.

A practical heritage from the Gauss-Newton method is the existence of analytical formula for calculating the matrix inverse by solving a system of linear equations.

Constrained Parameter Estimation [24]

Put simply, the constraints represent prior knowledge about the possible value of the estimated parameters. Extending the already introduced methods with constraints can improve the speed of convergence and reduce the number of iterations required to reach the desired precision of the sought parameters. All this is achieved by reduction of the parameter space.

The most effective approaches utilize *Lagrange multipliers*, approximate the *Kuhn-Tucker equations* or use *sequential quadratic programming*. Among their disadvantage is the complexity of their application. More detailed description of these methods is out of the scope of this thesis, more information can be found on the website of Mathworks [26].

Another approach incorporates the constraints by modifying the loss function to in-

clude the constraints. The modified cost function has the following form:

$$I_p(\boldsymbol{\theta}, p) = I(\boldsymbol{\theta}) + pP(\boldsymbol{\theta}) \tag{2.26}$$

where $P(\boldsymbol{\theta})$ = penalty function constraint violation
p = weighting factor

To find the optimal value of the weighting factor fine-tuning is required. The penalty function can be of two different types:

An *exterior* penalty function punishes the violation of boundaries by giving $P(\boldsymbol{\theta})$ large value, otherwise $P(\boldsymbol{\theta}) = 0$. Usually quadratic penalty function is used.

The *interior* penalty function grows as the estimated parameters approach their boundaries, therefore their value is forced to stay between the feasible limits.

3 Mathematical models of the vehicle

This chapter is the first of two chapters that presents the steps of the preparation procedure necessary to perform *software-in-the-loop*, or *SIL* tests on the different controller design and parameter estimation techniques introduced in the previous chapter. The information collected during the SIL simulations will be used to choose the suitable algorithms for the real-life application.

The central topic of this first chapter is the derivation of the mathematical models of the vehicle Hummer by the use of the *Newton's method*.

In the following part, two mathematical models will be presented. The first model will be a simpler two degree of freedom model, while the second will be a more complex one with one additional degree of freedom. To derive the models the analogy between the vehicle Hummer and an inverted pendulum will be used. Except of the non-linear models their linearised version will also be introduced, which will be later used for controller synthesis.

3.1 The 2DoF model

This section presents a two degree of freedom model, the two independent parameters are the forward motion and the rotation of the point mass.

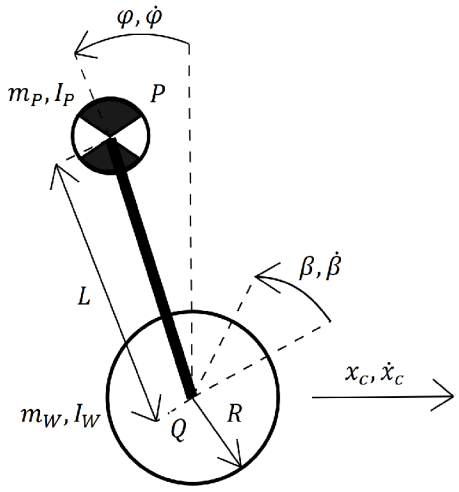


Figure 3.1: 2DOF model of the Hummer

In the presented model the non-conservative forces will be neglected. Among the neglected force are the rolling friction between the surface of the road and the wheels, viscous and the Coulomb friction in the bearings of the motor.

3.1.1 Kinematic equations of the 2DoF model

Before the equations of motion were derived the kinematic foundation had to be laid down. It is assumed that the wheel performs rolling motion without slipping, therefore between the translational forward motion and the rotation of the wheel the following expression holds:

$$\dot{x_c} + \dot{\beta}R = 0 \tag{3.1}$$

where $\dot{x_c}$ = translational velocity of the wheel
$\dot{\beta}$ = angular velocity of the wheel
R = radius of the wheel

The Equation 3.1 is often referred to as the *no slip condition*. After taking the second derivative of the Equation 3.1 with respect to time and reorganizing it for $\ddot{x_c}$ the following equation for the acceleration is obtained:

$$\ddot{x}_c = -\ddot{\beta}R \tag{3.2}$$

The kinematic description of the motion performed by the point mass on the rod is a little more complex, while it is the superposition of translational motion of the wheel and rotation of the point mass. The position of the point mass is given by the following expression:

$$x_p = x_c - Lsin\varphi \tag{3.3}$$

and

$$z_p = R + Lcos\varphi \tag{3.4}$$

where x_p = position of the point mass along the x axis
z_p = position of the point mass along the z axis
φ = leaning angle
L = distance of the centre of gravity of the mass point from the axis of the wheel

Acceleration of the point mass is given by:

$$\ddot{x}_p = \ddot{x}_c - L\ddot{\varphi}cos\varphi + L\dot{\varphi}^2 sin\varphi \tag{3.5}$$

$$\ddot{z}_p = -L\ddot{\varphi}sin\varphi - L\dot{\varphi}^2 cos\varphi \tag{3.6}$$

3.1.2 Equations of motion

The free body diagram in Figure 3.2 visualizes all the force and moments that will be taken into consideration for the derivation of the two degree of freedom mathematical model of the two-wheeled inverted pendulum.

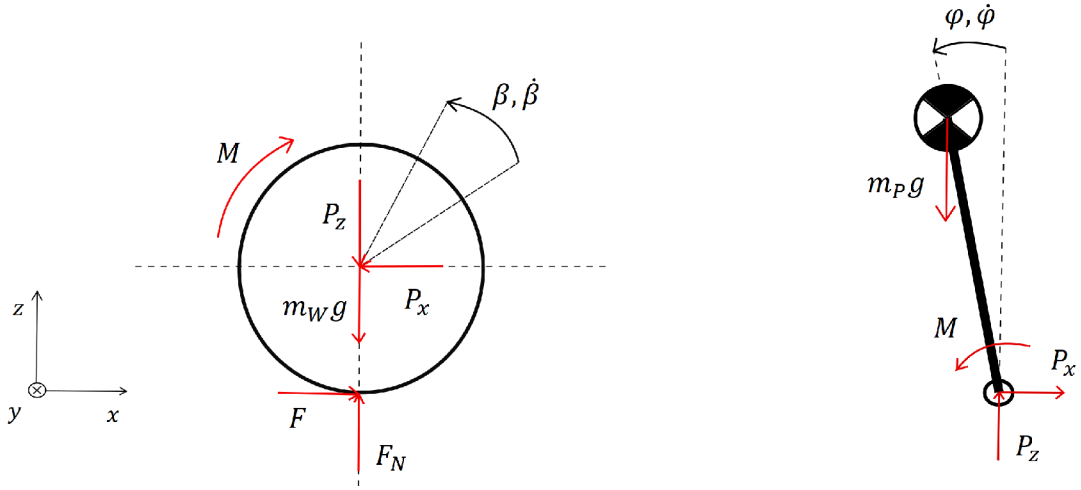


Figure 3.2: Free body diagram of the 2DOF model

The wheel performs rolling motion, therefore except of the two dynamic equations for the applied forces in the x and z directions a third equation for moment has to be included. Based on *Newton's second law* the following equations of motion can be written:

$$m_w \ddot{x}_c = F - P_x \tag{3.7}$$

$$0 = F_N - m_w g - P_z \tag{3.8}$$

$$I_w \ddot{\beta} = FR - M \tag{3.9}$$

The equation of motion for the point mass are the following:

$$m_p \ddot{x}_p = P_x \tag{3.10}$$

$$m_p \ddot{z}_p = P_z - m_p g \tag{3.11}$$

$$I_{p,y} \ddot{\varphi} = P_z L sin\varphi + P_x L cos\varphi + M \tag{3.12}$$

where m_w = mass of the wheel
I_w = moment of inertia of the wheel
m_p = mass of the point mass
$I_{p,y}$ = moment of inertia of the point mass
F = reaction force between the surface and the wheel
P_x = contact force between the wheel and the point mass in x direction
P_z = contact force between the wheel and the point mass in z direction
F_N = normal force

M = induced torque by the motor

By substituting the Equations 3.9 and 3.8 into Equation 3.10 and also applying the kinematic Equation 3.5 and the no slip condition from Equation 3.2 the following expression is obtained:

$$\ddot{\beta}(m_w R^2 + m_p R^2 + I_w) + \ddot{\varphi}(m_p L R cos\varphi) = \dot{\varphi}^2 L m_p sin\varphi - M \tag{3.13}$$

The second equation of the model is given by the following expression.

$$\ddot{\beta}(m_p L R cos\varphi) + \ddot{\varphi}(L^2 m_p + I_{p,y}) = m_p g L sin\varphi + M \tag{3.14}$$

The Equation 3.14 was acquired after adding Equations 3.10 and 3.11 together and substituting it into Equation 3.12. The final form was obtained after substituting the acceleration terms $\ddot{x}_p$ and $\ddot{z}_p$ with the previously derived kinematic expressions from the subsection 3.1.1.

3.1.3 Linearised 2DoF model

In Chapter 2 controller design techniques were presented that can only be applied to LTI models. Therefore to design regulators for the SIL simulations the derived equations from subsection 3.1.2 had to be linearised.

To linearise mathematical model of the two-wheeled inverted pendulum-like vehicle the following approximation formulas were used:

$$\begin{aligned} sin\varphi &\approx \varphi \\ cos\varphi &\approx 1 \\ \dot{\varphi}^2 &\approx 0 \end{aligned} \tag{3.15}$$

As point of interest the position where $\varphi = 0$ was chosen. This position was logical choice as the motion of the point mass is centered around this point. The linearised equations are given by:

$$\ddot{\beta}(m_w R^2 + m_p R^2 + I_w) + \ddot{\varphi}(m_p L R) = -M \tag{3.16}$$

$$\ddot{\beta}(m_p L R) + \ddot{\varphi}(L^2 m_p + I_{p,y}) = m_p g L \varphi + M \tag{3.17}$$

3.2 The 3DoF model

This section presents a three degree of freedom model, the three independent parameters include the forward motion and the forward or backward tilt of the point mass and the rotation of the vehicle around its vertical axis.

Just like in the case of the two degree of freedom model non-conservative forces will be neglected. Among the neglected forces are the rolling friction between the surface of the road and the wheels, the viscous and the Coulomb friction in the bearings of the motor.

These forces were neglected by the sake of simplicity of the derivation.

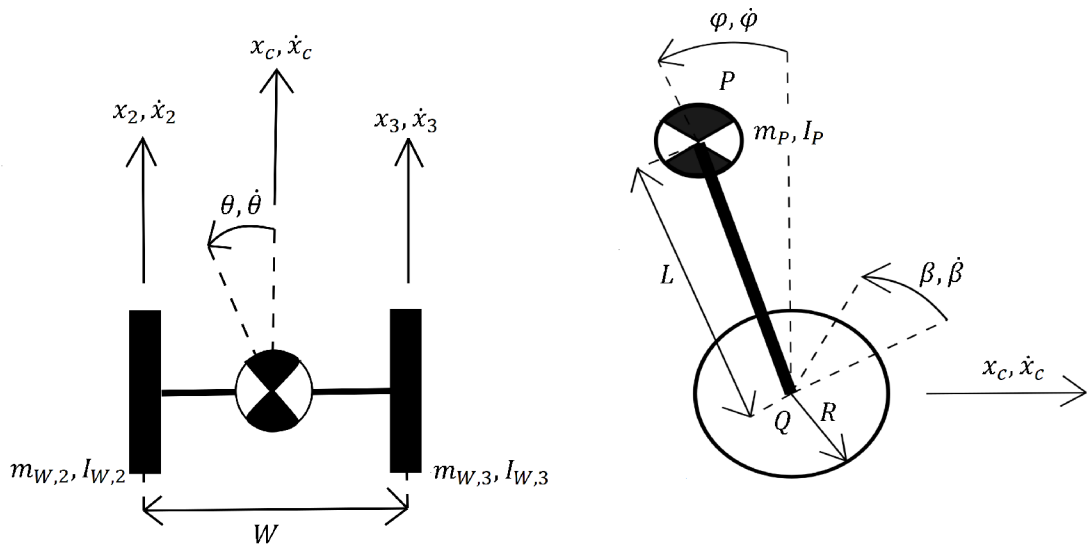


Figure 3.3: 3DOF model of the vehicle

3.2.1 Kinematic equations of the 3DoF model

The kinematics description in case of the three degree of freedom model is little more complex compared to the model with two degree of freedom. The complexity comes from the fact that in case of the *3DoF* model the non-holonomic character of the vehicle can not be neglected. Therefore to ensure punctual kinematic description the following two non-holonomic constraints have to be used. The first constraint formulates the mathematical relationship between the wheels' angular velocity and the resulting translational motion.

$$\dot{x_c} = \frac{-(\dot{\beta_3} + \dot{\beta_2})R}{2} \tag{3.18}$$

The second non-holonomic constraint is used to describe the speed of rotation of the vehicle around its vertical axis as a result of inequality of the wheels' angular velocity.

$$\dot{\theta_c} = \frac{(\dot{\beta_2} - \dot{\beta_3})R}{W} \tag{3.19}$$

where $\dot{x_c}$ = velocity of the vehicle
$\dot{\theta_c}$ = velocity of the vehicle's rotation around the vertical axis
R = radius of the wheels
W = distance between the wheels
$\dot{\beta_2}$ = angular velocity of the left wheel
$\dot{\beta_3}$ = angular velocity of the right wheel

The first derivatives of the expressions 3.18 and 3.19 are the following:

$$\ddot{x}_c = \frac{-(\ddot{\beta}_3 + \ddot{\beta}_2)R}{2} \tag{3.20}$$

and

$$\ddot{\theta}_c = \frac{(\ddot{\beta}_2 - \ddot{\beta}_3)R}{W} \tag{3.21}$$

The previously derived equations together with the kinematic equations 3.5 and 3.6 and also the no slip condition Equation 3.2 from Subsection 3.1.1, which are also valid for the *3DoF* case, will be used later in this chapter to derive the mathematical model.

3.2.2 Equations of motion of the 3Dof model

The equations of motion for the *3DoF* model were derived based on the forces and moments from the free body diagram in Figure 3.4.

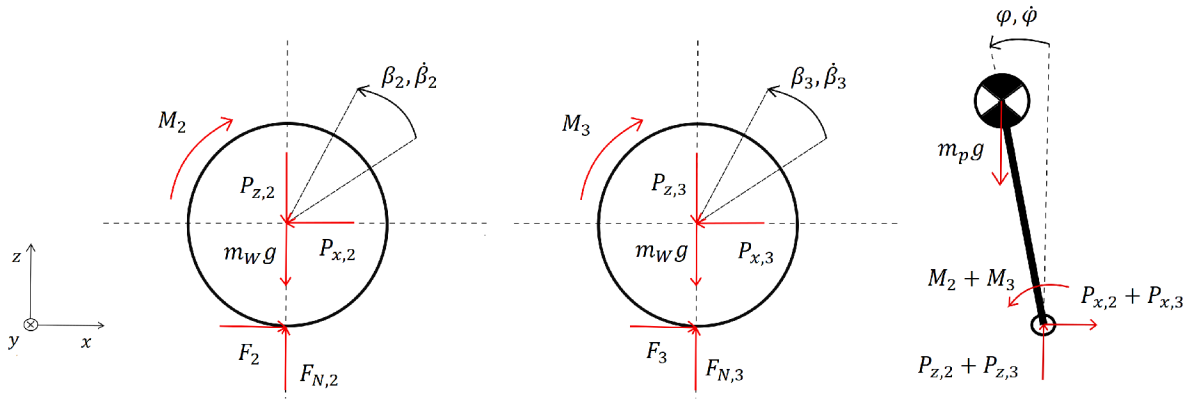


Figure 3.4: Free body diagram of the 3DOF model

Both wheels perform rolling motion the equations of motion are similar to the equations derived in subsection 3.1.2. The equations for the left wheel are the following:

$$m_w \ddot{x}_2 = F_2 - P_{x,2} \tag{3.22}$$

$$0 = F_{N,2} - m_w g - P_{z,2} \tag{3.23}$$

$$I_w \ddot{\beta}_2 = F_2 R - M_2 \tag{3.24}$$

The equations of motion for the right wheel are given by :

$$m_w \ddot{x}_3 = F_3 - P_{x,3} \tag{3.25}$$

$$0 = F_{N,3} - m_w g - P_{z,3} \tag{3.26}$$

$$I_w \ddot{\beta}_3 = F_3 R - M_3 \tag{3.27}$$

The performed motion by the point mass consists of translation, two rotations. around the axis of the wheels (leaning forward/backward) and rotation around axis z.

$$m_p \ddot{x}_p = P_{x,2} + P_{x,3} \tag{3.28}$$

$$m_p \ddot{z}_p = P_{z2} + P_{z,3} - m_p g \tag{3.29}$$

$$I_{y,p} \ddot{\varphi} = P_{z2} L sin\varphi + P_{z3} L sin\varphi + P_{x,2} L cos\varphi + P_{x,3} L cos\varphi + M_2 + M_3 \tag{3.30}$$

$$I_{p,z} \ddot{\theta}_c = -P_{x,2} \frac{W}{2} + P_{x,3} \frac{W}{2} \tag{3.31}$$

where m_w = mass of the wheel
I_w = moment of inertia of the wheel
m_p = mass of the point mass
$I_{p,y}$ = moment of inertia of the point mass for rotation around y axis
$I_{p,z}$ = moment of vehicle for rotation around z axis
F = reaction force between the surface and the wheel
P_x = contact force between the wheel and the point mass in x direction
P_z = contact force between the wheel and the point mass in z direction
F_N = normal force
M = induced torque by the motor

The numbers 2 and 3 in the lower indexes refer to the left and right wheel, respectively.

The first equation of the *3DoF* model is obtained after the equations of motion for left and right wheel, namely Equation 3.22 , 3.24 ,3.25 and 3.27 are substituted into Equation 3.28. The final form of this expression is obtained after applying the kinematic Equation 3.5 and the Equation 3.20 which was derived from the non-holonomic constraints. The resulting expression is given by:

$$\ddot{\beta}_2(m_w R^2 + \frac{m_p R^2}{2} + I_w) + \ddot{\beta}_3(m_w R^2 + \frac{m_p R^2}{2} + I_w) + \ddot{\varphi}(m_p L R) cos\varphi = \dot{\varphi}^2 L R m_p sin\varphi - M_2 - M_3 \tag{3.32}$$

The second equation of motion was obtained after adding Equations 3.28 and 3.29 together and substituting it into Equation 3.30. The final form was obtained after replacing the acceleration terms $\ddot{x}_p$ and $\ddot{z}_p$ with the previously derived kinematic expressions from the subsections 3.1.1 and 3.2.1. The following equation was acquired:

$$\ddot{\beta}_2 \frac{1}{2}(m_p L R cos\varphi) + \ddot{\beta}_3 \frac{1}{2}(m_p L R cos\varphi) + \ddot{\varphi}(L^2 m_p + I_{p,y}) = m_p g L sin\varphi + M_2 + M_3 \tag{3.33}$$

By substituting Equations 3.22, 3.24 , 3.25 and 3.27 into Equation 3.31 and also using the Equation 3.21, the third equation of the model is derived. It is given by:

$$\ddot{\beta}_2(I_w - m_w R^2 - \frac{2I_{p,z}R^2}{W^2}) + \ddot{\beta}_3(I_w + m_w R^2 + \frac{2I_{p,z}R^2}{W^2}) = M_2 - M_3 \tag{3.34}$$

3.2.3 Linearised 3DoF model

To linearise the *3DoF* model the same linearisation formula was used as in the case of the simpler *2DoF* model. This formula is given by the Equations 3.15. The resulting equations of the model are the following:

$$\ddot{\beta}_2(m_w R^2 + \frac{m_p R^2}{2} + I_w) + \ddot{\beta}_3(m_w R^2 + \frac{m_p R^2}{2} + I_w) + \ddot{\varphi}(m_p LR) = -M_2 - M_3 \tag{3.35}$$

$$\ddot{\beta}_2\frac{1}{2}(m_p LR) + \ddot{\beta}_3\frac{1}{2}(m_p LR) + \ddot{\varphi}(L^2 m_p + I_{p,y}) = m_p gL\varphi + M_2 + M_3 \tag{3.36}$$

$$\ddot{\beta}_2(I_w - m_w R^2 - \frac{2I_{p,z}R^2}{W^2}) + \ddot{\beta}_3(I_w + m_w R^2 + \frac{2I_{p,z}R^2}{W^2}) = M_2 - M_3 \tag{3.37}$$

4 Parameters of the vehicle

This chapter is devoted to describe the procedure of parameter attainment of the two-wheeled inverted pendulum-like vehicle, called Hummer. The parameters together with the mathematical models from the previous chapter will be used to create model in the loop simulations of the vehicle for development and testing of control and parameter estimation algorithms that will be later implemented on the real vehicle.

The sought parameters can be divided into electrical (armature inductance , motor armature resistance), mechanical (mass,moment of inertia) and physical dimensions (radius of the wheels, width of the chassis).

Direct measurement was used to obtain all the quantities related to physical dimensions and the mass of the individual parts. The other parameters were either estimated by the deployment of simulation tool called the *Parameter Estimation Toolbox* from Mathworks or calculated by the usage of 3D model in *Inventor*.

Quantity	**Description**	**Value**	**Unit**
m_{wh}	mass of the wheel	2.11	kg
m_p	mass of the body	33.05	kg
R	radius of the wheel	0.15	m
W	distance between the wheels	0.5	m

Table 4.1: The parameters obtained by direct measurement

In the following section the parameter estimation procedure will be described in details for both the electrical and mechanical parts. Next an alternative approach for calculation of moment of inertia will be presented.

4.1 Parameter Estimation

Parameter estimation was utilized to obtain the unknown parameters of the DC motor both electrical and mechanical parts and of the mechanical construction. The Parameter Estimation Toolbox requires the following two entries:

- a *Simulink* model of the system, where the sought parameters appear as variables of the model
- measured data on the real system

Next in the subsections the two cases will be treated separately and a detailed description will be given.

4.1.1 DC Motor

The complete mathematical description of DC motor requires two equations, each of them describing a sub-process, namely *electrical* and *mechanical* . To the equations of the mechanical sub-process non-conservative forces were also incorporated which include viscous and Coulomb friction acting in the bearings. In the following the mathematical equations will be presented, first the equation for the electrical sub-process, next for the mechanical sub-process.

$$u(t) = Ri(t) + L\frac{di(t)}{dt} + c\phi_{el}\omega(t) \tag{4.1}$$

$$c\phi_{tor}i(t) = J\frac{d\omega(t)}{dt} + b\omega(t) + ksign\omega(t) \tag{4.2}$$

where R = motor armature resistance
L = armature inductance
$c\phi_{el}$ = electrical constant of the motor
$c\phi_{tor}$ = torque constant of the motor
J = moment of inertia of the rotor
b = coefficient of viscous friction
k = coefficient of Coulomb friction
$u(t)$ = armature voltage
$i(t)$ = armature current
$\omega(t)$ = angular velocity

To use the software tool real data on the DC motor had to be measured about the course of physical quantities, e.g. armature voltage and current, angular velocity. The motor was placed on a test bench and different transducers were connected to it to measure the above-mentioned quantities. For data acquisition and for producing the control signals a *MF-624* I/O card from Humusoft was used, which was interfaced from Simulinnk.

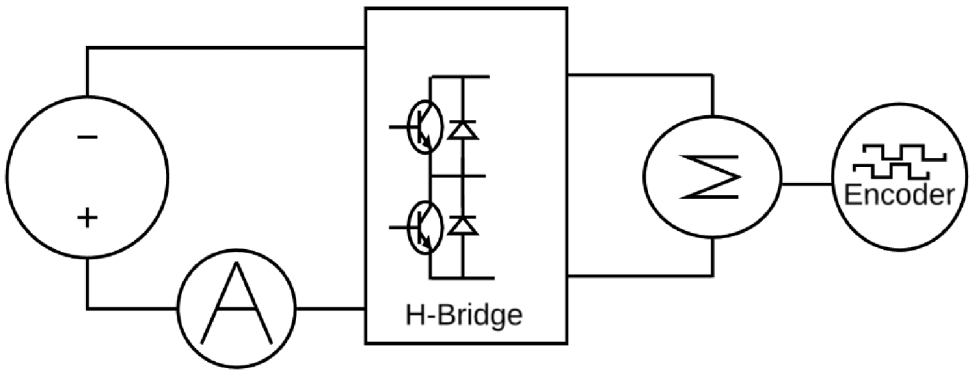


Figure 4.1: Block digram of the data acquisition circuit

For current measurement a *LTS 25-NP* transducer with operational amplifier was used

to improve the signal to noise ratio. The angular velocity was obtained after taking the derivative of the position signal measured by an encoder *AMT112Q-0256*. The armature voltage contrary to the other two quantities was not measured but calculated from the known value of the duty cycle and supply voltage according to the following formula:

$$u(t) = sU \tag{4.3}$$

where s = duty cycle
U = DC value of the voltage supplied to the motor

Parameters describing the motor's dynamical behaviour such as induction or the rotor's inertia could only be estimated from the transient response of the current and angular velocity, which could be only evoked by periodical changes of the duty cycle. Therefore a Simulink block called *Repeating Sequence Stair* was used to provide the changing duty cycle value.

The estimation procedure

The process of parameter estimation was divided into two steps. In the first step the unknown parameters from the electrical equation were estimated. As initial values of the unknown parameters, the coefficients presented in the motor's data sheet were used [27]. The input data to the model were armature voltage and angular velocity and as the output current data was used.

In the second step the remainder parameters were obtained. For this purpose the model included also the mechanical sub-process. Armature voltage, angular velocity figured among the entries as an input and output, respectively. During the estimation all parameters were estimated. The parameters obtained in the previous step had tighter constraint settings (around ±5% of their estimated value value) compared to the parameters of the mechanical equation. The limits for the electrical parameters were readjusted as their value approached the set value, this procedure was repeated until the whole process converged.

Quantity	**Value**	**Unit**
R	2.22	Ω
L	4.5	mH
$c\phi_{el}$	0.11	$V \cdot s$
$c\phi_{tor}$	0.12	Nm/A
J	2860	g/cm^2
b	1.22×10^{-4}	$Nm \cdot s$
k	2.55×10^{-2}	Nm

Table 4.2: Estimated parameters of the DC motor

4.1.2 Parameters of the Chassis

Similar methodology was used to estimate the position of the centre of gravity of the chassis. Segway-like vehicles are often referred to as inverted-pendulums, therefore this analogy was used to create the mathematical model of the chassis.

$$I_p \frac{d^2\varphi}{dt^2} = -m_p g L_p \sin\varphi - b\frac{d\varphi}{dt} - ksign\varphi \tag{4.4}$$

where I_p = moment of inertia of the chassis around the axis of the wheels
m_p = mass of the chassis
L_p = position of the centre of gravity
g = constant of the Earth's gravitational field
φ = yaw angle of the chassis

For b and k the values from Table. 4.2 were used. The moment of inertia of the pendulum was obtained based on the following analytical formula:

$$I_p = L_p^2 m_p \tag{4.5}$$

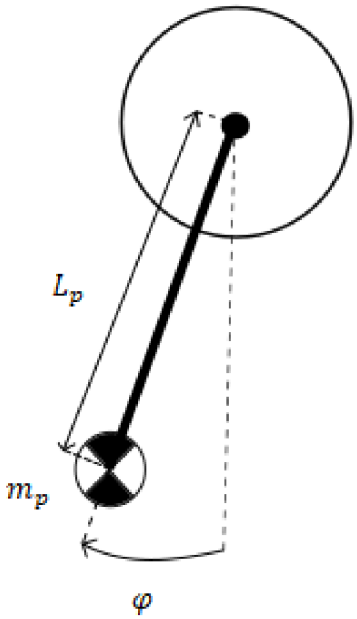


Figure 4.2: Schematics of the vehicle

Measurement and Estimation Procedure

The data acquisition was executed in the following way. The vehicle was hanged by its wheels and the handle bar pointed to the ground. At the beginning of the measurement the chassis was shifted by an angle φ_0 from its stable position and then it was let to

oscillate until it again settled in the stable position. The change in the yaw angle was measured by an encoder.

Then the parameter estimation was carried out. Among the known entries to the model were the mass of the chassis and the friction coefficients b and k, the measured yaw angle was set as output data. The constraints were set the same way as described in the estimation subsection for the DC motor. The following parameters were obtained:

Quantity	**Value**	**Unit**
L_p	58.6	mm

Table 4.3: The position of the centre of mass of the vehicle

4.2 Moment of inertia calculation

The mathematical model with 3 degree of freedom requires, except of the previously obtained parameters, also the moment of inertia around the vertical coordinate axis z. An alternative approach was used to obtain this parameter. A 3D model of the vehicle's body was created based on the physical dimensions and weight of the chassis, the motors and the battery packs. For this purpose a computer-aided design software called *Inventor* was used. To obtain the sought parameter a special feature of the software was utilized, which can calculate the moment of inertia around arbitrary axis.

Quantity	**Description**	**Value**	**Unit**
$I_{p,z}$	moment of inertia around the z axis	0.93	$kg \cdot m^2$

Table 4.4: The moment of inertia of the chassis

5 Software-in-the-Loop Testing

In the previous chapters different models and algorithms for controller synthesis and for parameter estimation were discussed that can be of practical use to achieve the goal of this thesis which is to implement an adaptive controller on the two-wheeled self-balancing vehicle to improve its stability and user safety. This chapter builds on the previous chapters, while it seeks to present a suitable solution as a combination of controller synthesis and parameter estimation algorithms and familiarize the reader with the process that has led to their selection.

To find the satisfactory algorithms software-in-the-loop tests were carried out on both types of algorithms on the two types of mathematical models that were formerly derived.

The outline of this chapter is the following. The first two sections will present the SIL tests of the estimation and synthesis subtask separately. In the third section the final software-in-the-loop test will be presented, which aimed at finding out whether the best individual solution of these subtasks could be combined together to form an adaptive controller which could be later applied to the real system. The fourth and final section of this chapter will present the plan of implementation of the chosen solution on the real vehicle.

5.1 Parameter Estimation

In this section a variety of algorithms for obtaining unknown parameters of the system will be tested. Among the sought parameters are the passenger's mass, centre of mass and moment of inertia.

The algorithms to be tested are the Newton-Raphson method (NR), the Gauss-Newton(GN) and the Levenberg-Marqurdt (LM) method. These methods were chosen based on the following advantageous features:

- fast convergence
- the Hessian is not required
- computational requirements are relatively small

Even though these methods are computationally inexpensive, the test on the ECU showed that the maximal size of the matrices in the estimation problem, that can be computed without jeopardizing the safety of the passenger (hard real-time system), is a *2 by 2* matrix. With this limitation in mind during the testing of the estimation techniques the maximal problem's size was restricted to *4 by 2* matrices. This decision which contradicts the limitation of the practical problem was made to see how additional measurement data improves the precision of the estimated parameters.

Due to the above-mentioned restriction the number of parameters that would be estimated was reduced to two, namely the passenger's mass and centre of mass.

The purpose of this testing is to select the most suitable algorithm from the above mentioned candidates that can be eventually applied to the real vehicle. The decision will be made based on the following criteria:

- accuracy of the estimation
- the required number of iterations
- the number of computationally demanding operations with matrices

In the upcoming part the testing procedure will be described.

5.1.1 Description of the Test and Results

All of the algorithms were given the same task which was to estimate the mass and the centre of mass of an average sized male passenger who was driving the vehicle. The test was carried out in simulation. The simulated model consisted of the *2DoF* model of the vehicle derived in the previous section and the passenger who was modelled as point mass. During the simulation, ride data about the quantities necessary to perform the parameter estimation was collected. The data included the leaning angle, the angular displacement of the wheels and the regulator output (moment).

Then from the dataset samples were drawn, and on the samples the parameter estimation was performed and eventually the results were evaluated by statistical methods. This procedure was repeated for all of the above-mentioned techniques. In total six experiments were carried out, two modifications of the three estimation techniques. The modification differed only in the number of samples used to estimate the sought parameters.

As it was mentioned before these methods do not require the computation of the Hessian, instead they approximate it with the Jacobian matrix. The Jacobian matrix for the *2 by 2* sized problem is the following:

$$J = \begin{bmatrix} J_{1,1} & J_{1,2} \\ J_{2,1} & J_{2,2} \end{bmatrix} \tag{5.1}$$

where $J_{1,1} = \ddot{\beta}R^2 + \ddot{\varphi}LRcos\varphi - \dot{\varphi}^2 LRsin\varphi$
$J_{1,2} = \ddot{\varphi}m_p Rcos\varphi - \dot{\varphi}^2 m_p Rsin\varphi$
$J_{2,1} = \ddot{\beta}LRcos\varphi + \ddot{\varphi}L^2 - gLsin\varphi$
$J_{2,2} = \ddot{\beta}m_p Rcos\varphi + 2\ddot{\varphi}Lm_p - m_p gsin\varphi$

From now on the methods utilizing the Equation 5.1 with samples from one specific time instant will be referred to as the basic variant of the given estimation approach. The extended variant deploys data from two successive time instants, therefore in the calculations *4 by 2* Jacobian was used.

Conclusion

In Table 5.1 the results of the testing are presented. The abbreviations m_H , L_H mark the mass and the height of the passenger, respectively, while the letter N is the number of iteration that was required to compute the estimated values. As it was mentioned above

the passenger was an average sized male, who weighed $m_H = 70\ kg$ and his centre of gravity was $L_H = 0.875\ m$ above the axis of the wheels.

Method's name	$\overline{m}_H\ [kg]$	$\sigma(m_H)\ [kg]$	$\overline{L}_H\ [m]$	$\sigma(L_H)\ [m]$	$\overline{N}$
basic NR	68.84	7.45	0.77	0.86	5.45
extended NR	70.76	6.55	0.77	0.85	4.67
basic GN	68.90	7.50	0.74	0.98	5.48
extended GN	70.76	6.55	0.78	0.85	4.67
basic LM	68.88	7.51	0.73	1.00	11.06
extended LM	70.74	6.55	0.78	0.85	7.85

Table 5.1: Evaluation of the parameter estimation techniques

Based on the introduced criteria the *Newton-Raphson* method was selected as it estimated the parameters with the highest accuracy while requiring the least number of iterations.

5.2 Controller Synthesis

This section analyses the problematics of finding a suitable controller synthesis approach which ensures passenger's safety and comfortable handling of the vehicle. The task itself can be divided into two subtasks that have to be addressed, which are the following:

- stabilizing the vehicle in the vertical position with and without a driver
- controlling the cornering of the vehicle as a response to driver input (sideways tilt of the handlebar)

From now on these subtasks will be treated separately. For stabilizing the vehicle the controller was designed by the LQR and the Ackermann's formula. This choice was based on the following two facts:

1. the feed-back gain matrix is computed iteratively, respectively closed-loop solution is available, which makes them ideal for adaptive controllers
2. the availability of only two parameters from the total three, that are necessary to fully describe the dynamics of the human body

To solve the cornering subtask a *P controller* with constant gains will be used, similarly to the solution implemented by [6] during the previous works on the vehicle.

Just like in case of the parameter estimation methods the goal was to find suitable algorithms to control the vehicle, therefore two experiments were carried out. The first experiment aimed at identifying the right stabilization algorithm that can be updated on-line, while the second was used to test the controller for cornering.

For safety reasons the algorithms were first tested in simulation by an approach called software-in-the-loop testing.

In the upcoming section the testing procedure will be described and based on its result a suitable solution will be selected.

5.2.1 Controller for Stabilizing the vehicle

This subsection focuses on the SIL testing of the stabilization algorithm. The main goal of this test was to find out, whether the upper-mentioned algorithms can handle two different operating conditions without changing the gains of the state-feedback gain matrix. The following operating conditions were of interest from the application point of view:

- the vehicle is empty
- the vehicle is carrying an average sized male passenger (height: 175cm weight: 70kg)

Among the other objectives were finding the right poles for the controller and also obtaining basic knowledge on how changing the closed-loop poles would affect the system's behaviour, which might prove useful when it comes to fine-tuning of the regulator on the real vehicle.

The test itself was relatively simple, it consisted of two stages. In the first stage the vehicle was not carrying any passenger, while in the second an average sized passenger was standing on it. At the start the vehicle was standing still in the stable upright position and user input simulated by a Gaussian shaped force pulse was applied to the handles to set it in motion. After the vehicle stopped and the passenger got on the whole process was repeated once again. During the whole test the vehicle was performing motion along straight line, therefore the testing was carried out on *2DoF* model of the system.

The testing was carried out in Simulink, where the model was structured according to the Figure 5.1.

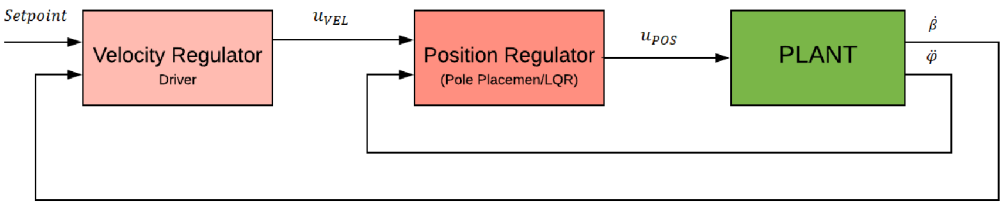


Figure 5.1: Block diagram of the model in Simulink

The position controller, which stabilized the vehicle in the vertical position was designed either by LQR or Ackermann's method. To use these methods the linearised model from Subsection 3.1.3 was reorganized for the quantities related to the leaning angle which are necessary for balancing the vehicle. The following expression was obtained:

$$\ddot{\varphi}(H_3H_1 - H_2^2) - H1m_pgL\varphi = M(H_1 - 1) \tag{5.2}$$

where $H_1 = m_wR^2 + m_pR^2 + I_w$
$H_2 = m_pLR$

$$H_3 = L^2 m_p + I_{p,y}$$

From the equation above the following LTI model was derived:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ \frac{H1 m_p g L}{H_3 H_2 - H_2^2} & 0 \end{bmatrix} \begin{bmatrix} \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{H_1 - 1}{H_3 H_1 - H_2^2} \end{bmatrix} [M] \tag{5.3}$$

In the model an additional velocity controller was implemented to account for the driver's control efforts. Based on the supposed nature of the driver's control actions a *P type controller* was proposed to model it. The addition of this controller was necessary to eliminate the steady-state leaning angle error, which was caused by the inclusion of the non-conservative forces, that where neglected during the derivation of the model. During the SIL tests viscous and the Coulomb friction in the bearings of the motor were taken into consideration.

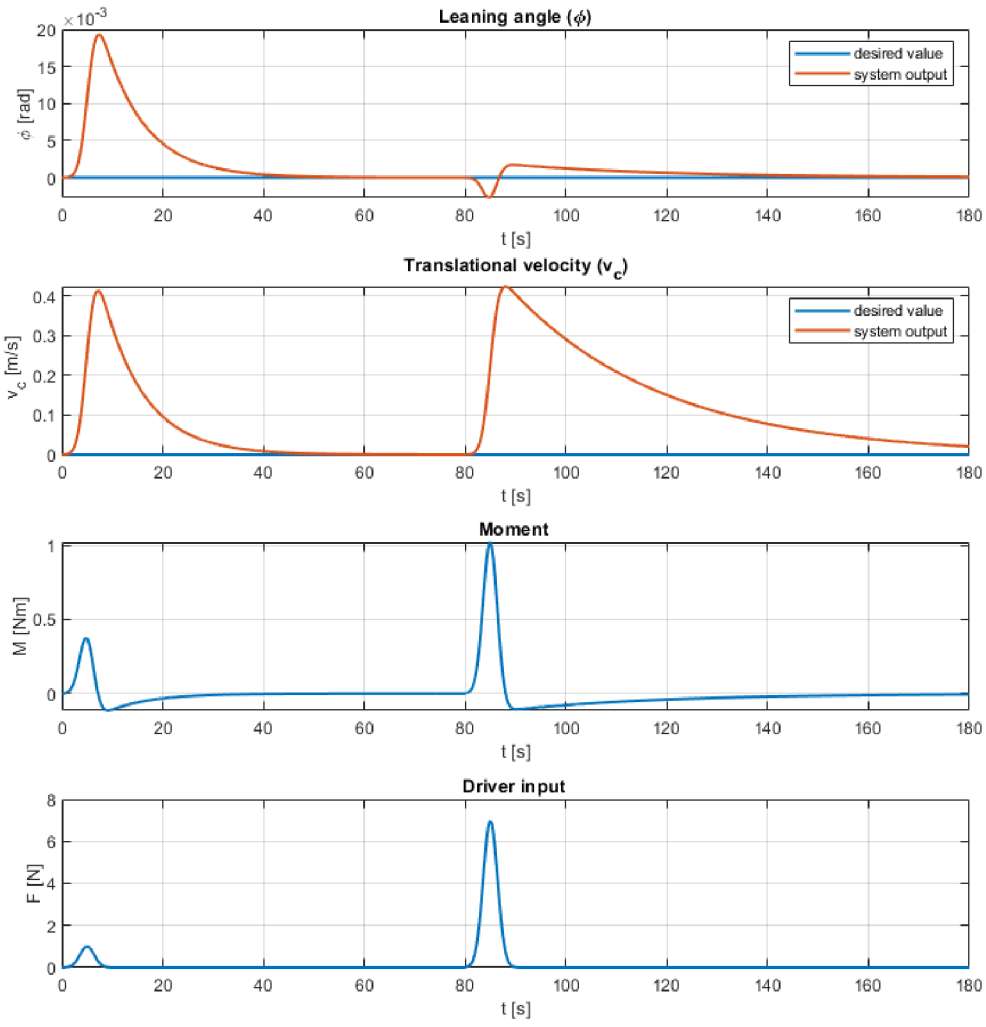


Figure 5.2: Result of the test of the pole placement by Ackermann's method

Pole Placement Method

In the following the test results for the pole placement by Ackermann's formula will be presented. The suitable poles for handling the two operating state were the following:

$$P = [-0.43, -318]$$

In Figure 5.2 the results of the simulation is presented.

Linear Quadratic Control Approach

Usage of the LQR approach requires the definition of the weight matrix of states Q and inputs R, which were chosen as follows:

$$Q = \begin{bmatrix} 50 & 0 \\ 0 & 50000 \end{bmatrix} \quad R = [1]$$

The relatively high requirements on the precision of the second state ($\dot{\varphi}$) were necessary due to the system's oscillatory behaviour after the user input was applied to it.

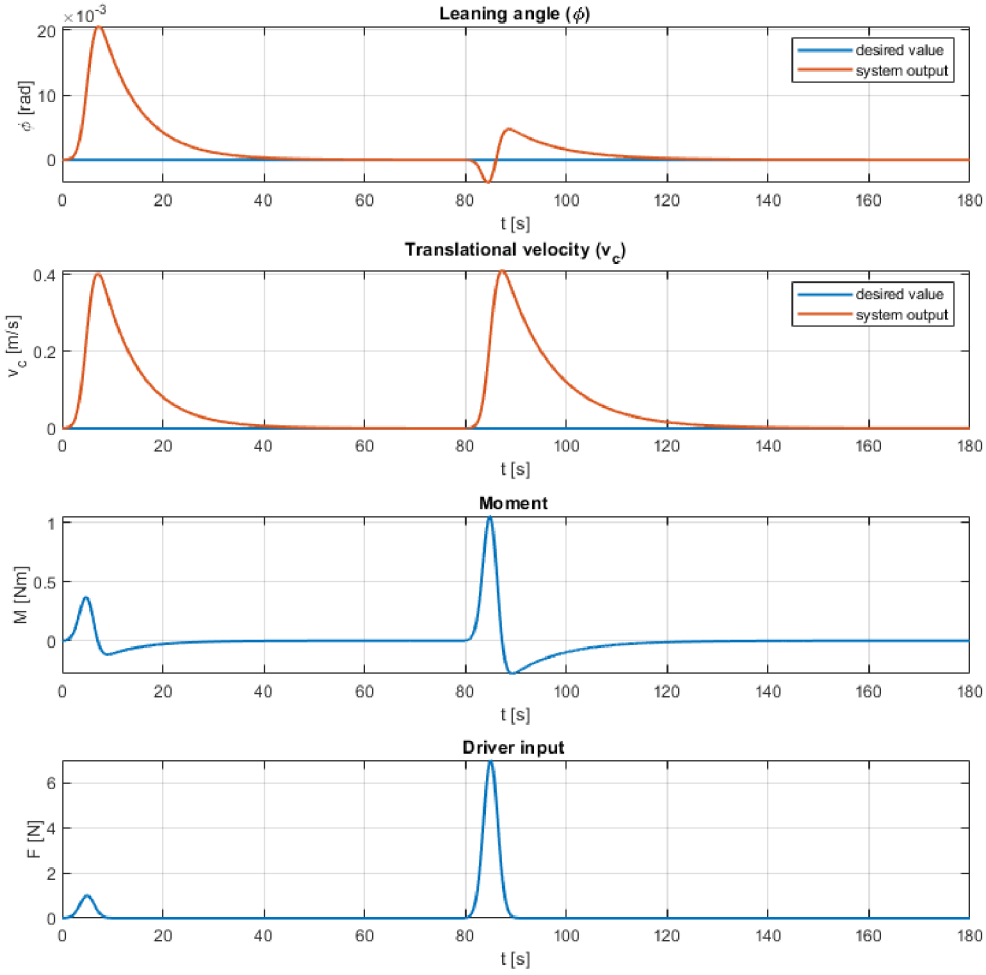


Figure 5.3: Result of the test of the LQR method

Conclusion

Based on the results displayed in Figure 5.2 and 5.3 the following conclusion can be made. Both methods can effectively handle the two tested states, even though the leaning angle controller designed by the LQR method offered slightly better positional control, the computational complexity of this method outweigh its advantages, therefore for the on-line regulator synthesis the pole placement by Ackermann's method will be used, which contrary to the LQR problem has a closed form solution.

As it was mentioned above the presented test results were obtained on the *2DoF* model of the vehicle. These results also apply for the *3DoF* model as the two models behaviour for the given circumstances (motion along a straight line) is very similar.

5.2.2 Regulator for Cornering

This subsection proposes solution for the cornering subtask. As it was mentioned above, the fact that the moment of inertia for the vertical axis can not be estimated on-line, the usage of controller with constant gain was the only reasonable option.

In case of the control algorithm implemented during the previous work [6], the turning was controlled by a *proportional controller*. This regulator worked as follows. The user input derived from the sideways tilt of the handlebar, which was fed through the P controller, and the resulting controller output was then directly subtracted from one wheel's duty cycle added to the second wheel's duty cycle, this way speed difference between the wheels was induced, as a result of this the vehicle turned.

Due to the fact that in the main loop the regulator's output quantity is not duty cycle of the PWM but moment the controller's gain of the controller had to be tuned to obtain the desired behaviour.

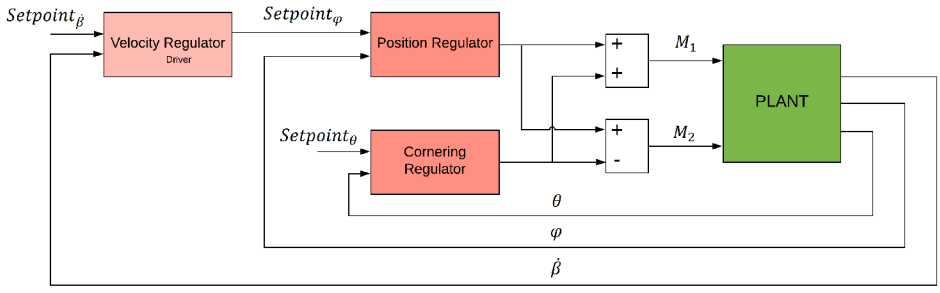


Figure 5.4: Block diagram of the cornering regulator and regulator for stabilization

Due to the fact that this solution had proved to be efficient throughout the years, its SIL testing was unnecessary. Further testing of the modified controller gain was carried out on the real vehicle.

5.3 Adaptive Controller

Until now the parameter estimation and the controller synthesis was treated separately, but now the two techniques will be combined to form an indirect adaptive pole placement

controller. As it was mentioned in the research study the indirect APPC consists of an adaptive law and a control law. Based on the experiment from the previous sections the suitable algorithms for the individual parts were selected.

- **adaptive law**: conjugate gradient method
- **control law**: Ackermann's formula (pole placement)

This section presents the last SIL test of the control algorithm before it would be implemented on the real vehicle. This test is almost identical to the one carried out to examine the controller algorithm. Contrary to that experiment, where the regulator's feed-back gains were kept constant throughout the whole test, this time the driver's mass and position of his or her centre of gravity will be estimated and then used to update the feed-back gain vector. The Figure 5.7 displays the block diagram of the adaptive controller used for the simulations in Simulink environment.

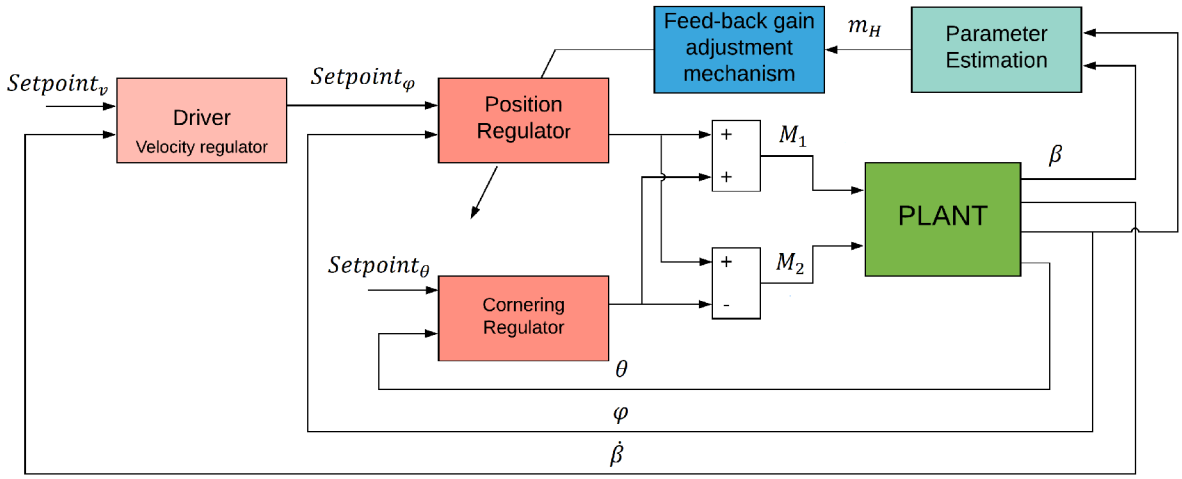


Figure 5.5: Block diagram of the APPC model in Simulink

The direct implementation of the adaptive and control law led to a complete failure. Therefore gradual amendments had to be performed in the adaptive algorithm to obtain the required performance. During these amendments the following observations were made:

- **Adaptive Law**
  - the parameter estimation procedure produces incorrect estimates due to model imperfections, therefore the estimation should be limited to instants when the disturbances are not present
  - with applied constraints estimated mass of the driver becomes accurate, however the position of the centre of gravity remains inaccurate
- **Control Law**
  - the Ackermann's formula disregards the real system's physical limitations ,i.e., maximal output torque of the motor, therefore the controller output can not be performed and the system becomes uncontrollable

– the parameter space can not be traversed well enough to be certain that the on-line calculated controller will stabilize the vehicle and will not jeopardize user safety

Based on the above-mentioned observations the adaptive algorithm was modified.

– the estimates are taken after the angular acceleration of the leaning angle is between a certain upper and lower bound.

– the position of the centre of gravity can be approximated based on the estimated mass of the driver as these parameters are related

– the control law is realized similarly to a lookup table, to each interval precomputed gains are assigned

After implementation of these changes the following satisfactory results were obtained:

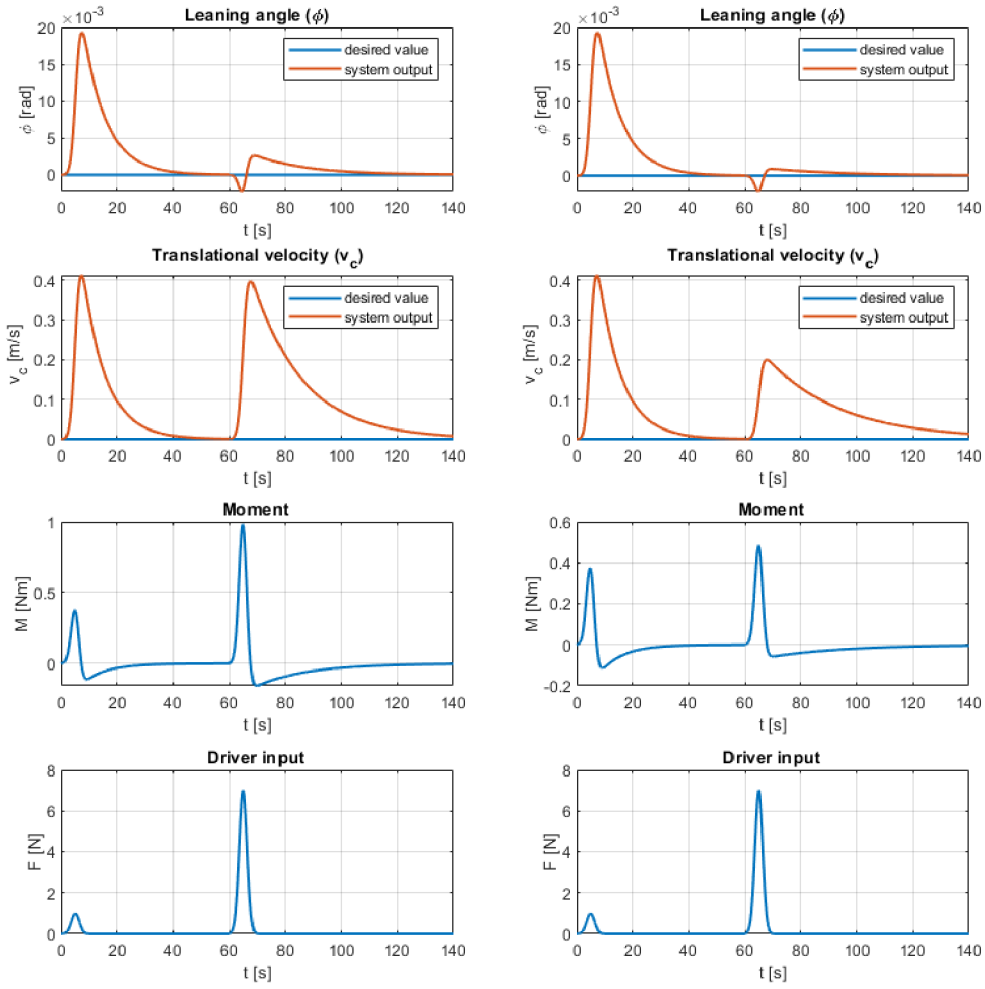


Figure 5.6: Comparison of the classical pole placement(left) and the adaptive controller (right)

5.4 The Plan of Modifications

Based on the tests from the preceding sections the final control system's structure was created. This section briefly presents the plan of the eventual solution that would be implemented on the vehicle Hummer, more in debt analysis and description of the problematic will be given in the upcoming chapters.

The selected adaptive pole placement controller algorithm can be divided into three main parts, which are the following:

- **regulator**
- **parameter estimator**
- **gain scheduler**

The working of the regulator, parameter estimator and the gain scheduler was explained thoroughly therefore no additional explanation is required, however, there is one more important thing that has been said little about.

The regulator that stabilizes the vehicle and is responsible also for controlling its heading direction was designed to output the desired amount of moment that has to be induced by the DC motors to control the vehicle. To ensure that the moment produced by the motors matches the desired value an additional regulator for moment had to be implemented. That means that stabilization and control algorithm was realized as a cascaded structure of regulators, consisting of a high speed moment loop and slower regulator for stabilization and heading direction control.

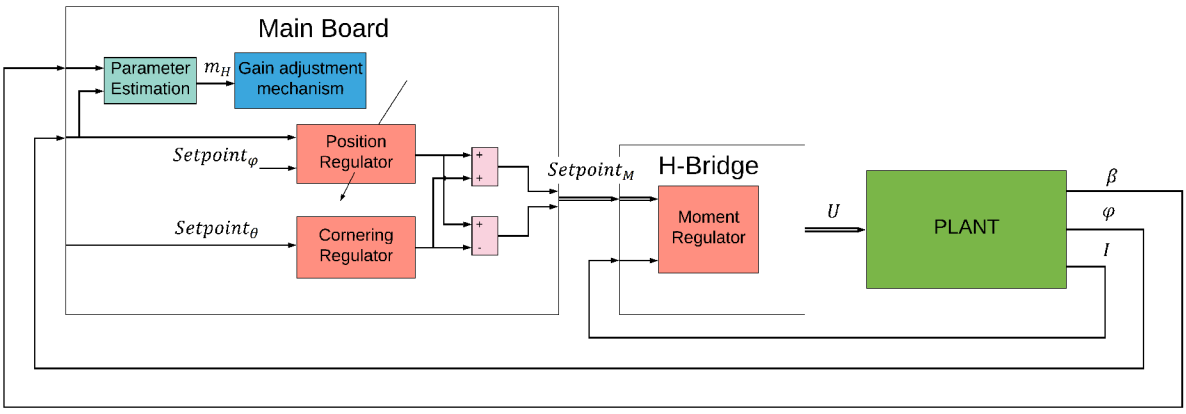


Figure 5.7: Block diagram of the implemented algorithm

The Figure 5.7 introduces the basic structure of the new controller algorithm. The regulator, parameter estimator and the gain scheduler algorithms together with fault detection and advanced safety features were implemented on the electronic control units of the Main Board. As it was mentioned above the redundant structure of the ECU's was kept. Therefore both ECU's would have identical tasks and in case of failure the CPLD would switch between the ECU units.

On the other hand the moment regulator will be implemented separately on the H-Bridge modules. This solution had the following advantages:

- lessening the computational burden on the ECUs
- lower requirements on the communication between the main board and the PCB with the H-Bridge
- modular structure makes prospective maintenance and upgrades easier

6 Electronics

With the plan of the modifications in mind, a list of requirements for the vehicle's hardware was created. Based on this list the state of the electronics was examined to find out whether it meets the requirements. Emphasis was on reusing most of the original electronics and only replacing what was necessary.

This chapter begins by presenting the functional units of the "old" electronics that were reused. Next the requirements on the new parts of the hardware are presented, based on the knowledge about the requirements set by the new control and parameter estimation algorithms. In the last part of this chapter the proposed solution is presented.

6.1 Preserved parts from the old electronics

During the previous works on the Hummer a dual ECU structure was implemented, which mimicked the structure of the control system from the original Segway.

The two ECUs were responsible for handling the position regulation of the vehicle and interfaced the sensors, while a third unit with CPLD was used to switch between them in case one of the ECUs failed. This structure throughout the years proved to be efficient and improved the robustness of the whole system. Therefore it was a unambiguous decision to preserve this organization of control systems. More detailed information about it is available in [6], [5].

During the early development stages ECU units were tested and it was found that their maximal utilization with the software from the previous master's thesis was just below 30%. Hence it was concluded that the microcomputers of the ECU board are powerful enough to handle the possible increase in required amount of computation as a result of the new software.

Regarding the hardware, the following functional units were kept as their replacement was unnecessary.

- two PCBs with the dsPIC33EP512MU810 micro-controller, which served as the electronic control unit for the whole vehicle
- the PCB with the Xilinx XC94144 CPLD module, which was used as safety feature to switch between the two ECUs in case one of them stopped working
- the board that served as the power supply for the control electronics

6.2 Requirements for the new modules

Among the units that needed to be redesigned were the PCBs with H-Bridge and the Main Board which served as a motherboard, while it connected all the sensors with the individual microcomputers.

In the following the specification on the new parts of the Hummer are presented.

1. **H-Bridge Board**
   (a) H-Bridge circuit that can supply 36 V and around 5 to 6 A continuous current
   (b) overcurrent and overtemperature protection
   (c) high-quality current measurement
   (d) built in micro-controller to handle the current regulation
   (e) communication between the main and the H-Bridge Board
      i. desired control quantity as pulse width modulated signal and individual pins for setting the direction, enabling the device and board status
      ii. serial peripheral interface (SPI)
   (f) similar physical dimensions as the "old" H-Bridge Board

2. **Main Board**
   (a) data transfer via USB between the Main Board and PC/display
   (b) data logging
   (c) interfacing the following sensors
      i. 2x IMU (one for each ECU)
      ii. 2x encoder
      iii. 2x H-Bridge
      iv. 2x potentiometers(one for each ECU)
   (d) ensure data transfer between the two ECUs and the CPLD Board
   (e) supply 3.3 V for the control electronics and 5 V for the encoders

Besides the required upgrade of the electronics, changes were needed also in the sensory part of the vehicle. The algorithm for parameter estimation requires the use of two different sensors. Except of the IMU sensor which was already used for tilt angle measurement a quadrature encoder had to be added to measure the speed of rotation of the wheels. The following requirements had to be addressed while choosing the encoder:

- differential output
- maximal resolution should be around 250 to 300 counts per revolution
- constraints on the physical dimensions of the encoder
- dust, vibration and electromagnetic field resistance
- maximal supply voltage 5 V
- mounting on the motor

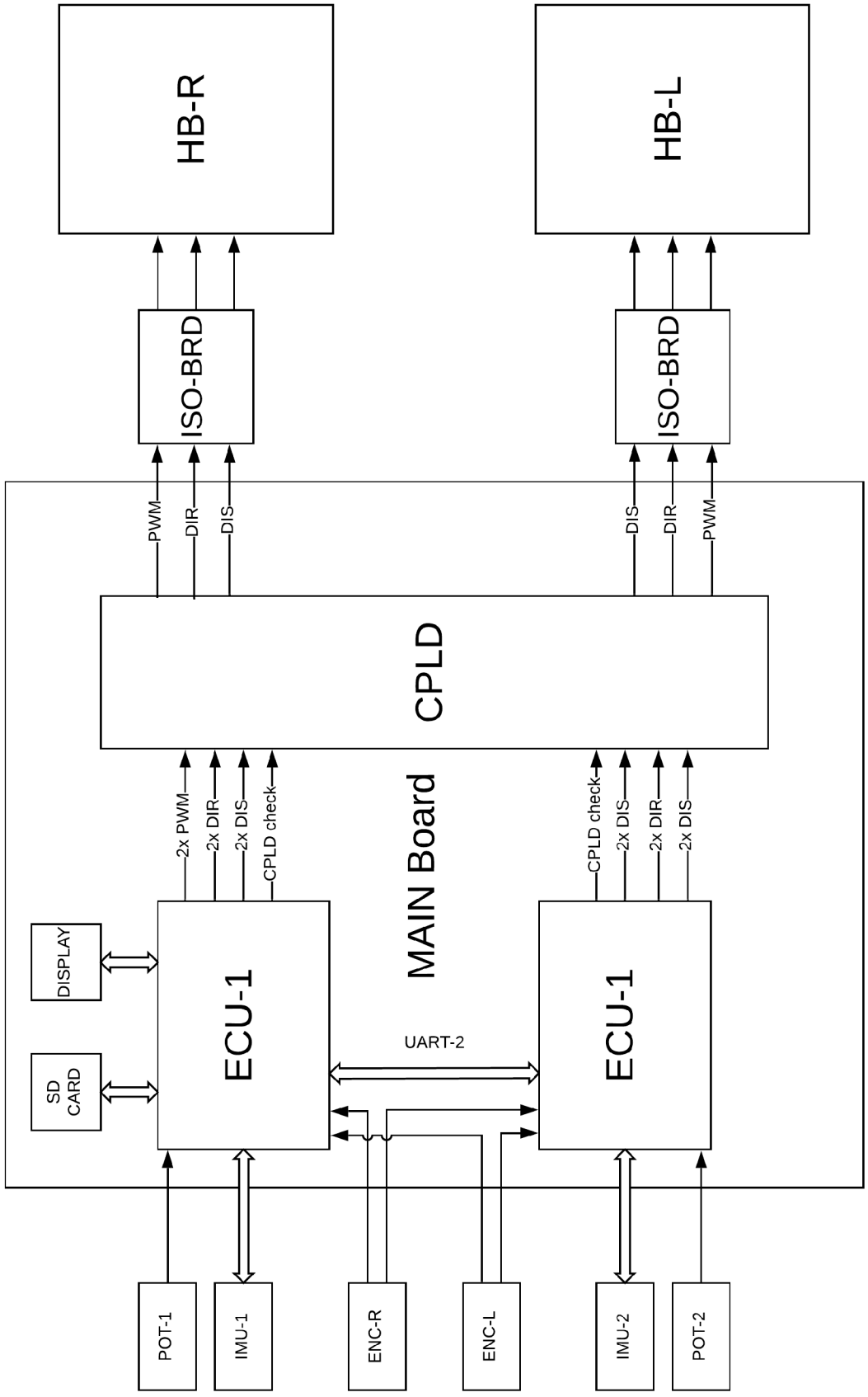


Figure 6.1: Block diagram of the electronics with the new parts. Adapted from [5]

6.3 The new hardware

Based on the requirements from the previous section the new modules were designed. In the following parts a step-by-step approach will be taken to introduce each piece of the new hardware with all the design consideration and problems that occurred during their implementation.

6.3.1 H-Bridge Board

The inspiration for the new design of the H-Bridge Board was drawn from PCB designed by Ing. Jan Chalupa in Mechatronics Laboratory, which implemented an integrated motor driver circuit called *DRV-8432* [28].

The *DRV-8432* is a highly efficient (up to 97%) dual H-Bridge module from Texas Instruments, with integrated driver and protection circuitry. Among its desirable features are supply voltage up to 52 V and 7 A continuous current, overheat and overcurrent protection and simple implementation. Expect of the previously mentioned features, the fact that its functionality was tested and verified made it a premier choice for this application.

The on board MCU was chosen based on previous experience with Microchip's 16-bit micro-controllers. The chosen *dsPIC33EP32MC204* microcomputer dispose among others, numerous re-mappable pins, a built-in high-speed PWM, input capture, and two SPI modules for communication with other devices; e.g. analog-to-digital converter.

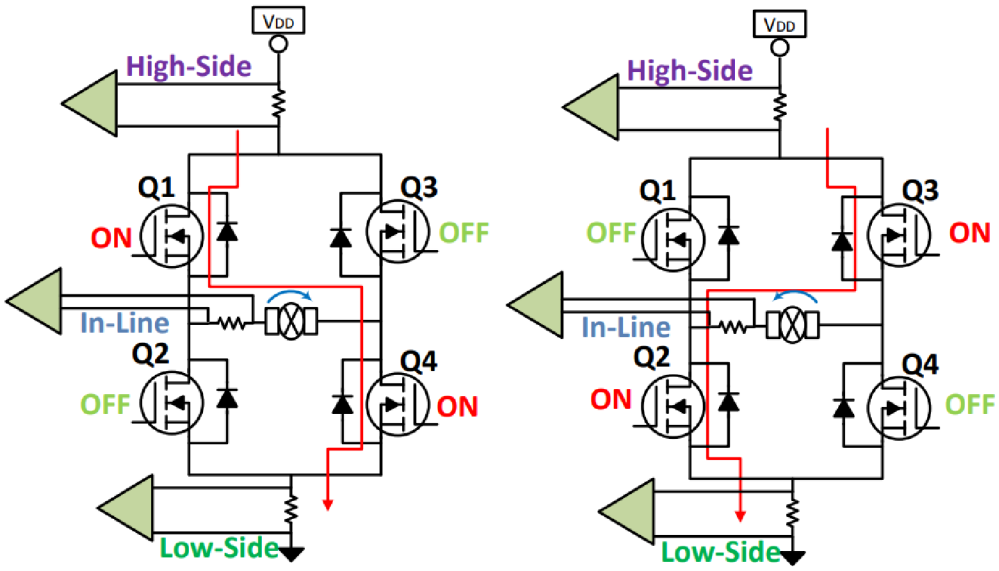


Figure 6.2: Current Sensing, placement of the current sensor in H-Bridge circuit [29]

The requirements on high quality current measurement were solved by the *ACS711* Hall-effect linear current sensor from Allegro Micro-systems. This transducer offers sensing range up to ± 12.5 A with 110 mA/V sensitivity and produces galvanically isolated single-ended analog voltage signal, in case of 3.3 V supply voltage the output voltage ranges from 0.3 to 3 V. The sensor was placed in-line with the H-bridge and the DC mo-

tor Figure 6.2. This placement of the transducer is often used in control applications as it offers direct measurement of the load current and also information about its direction.

The outputted voltage signal is fed through a voltage follower and is filtered before it is digitized. The conversion of the analog voltage is done by a *ADS8866* 16-bit ADC. The digitized data is transferred to the processing unit via serial interface.

For connecting the H-Bridge Board and the Main Board 9 pin Cannon D-sub connector was used. Table 6.1 presents the PIN mapping.

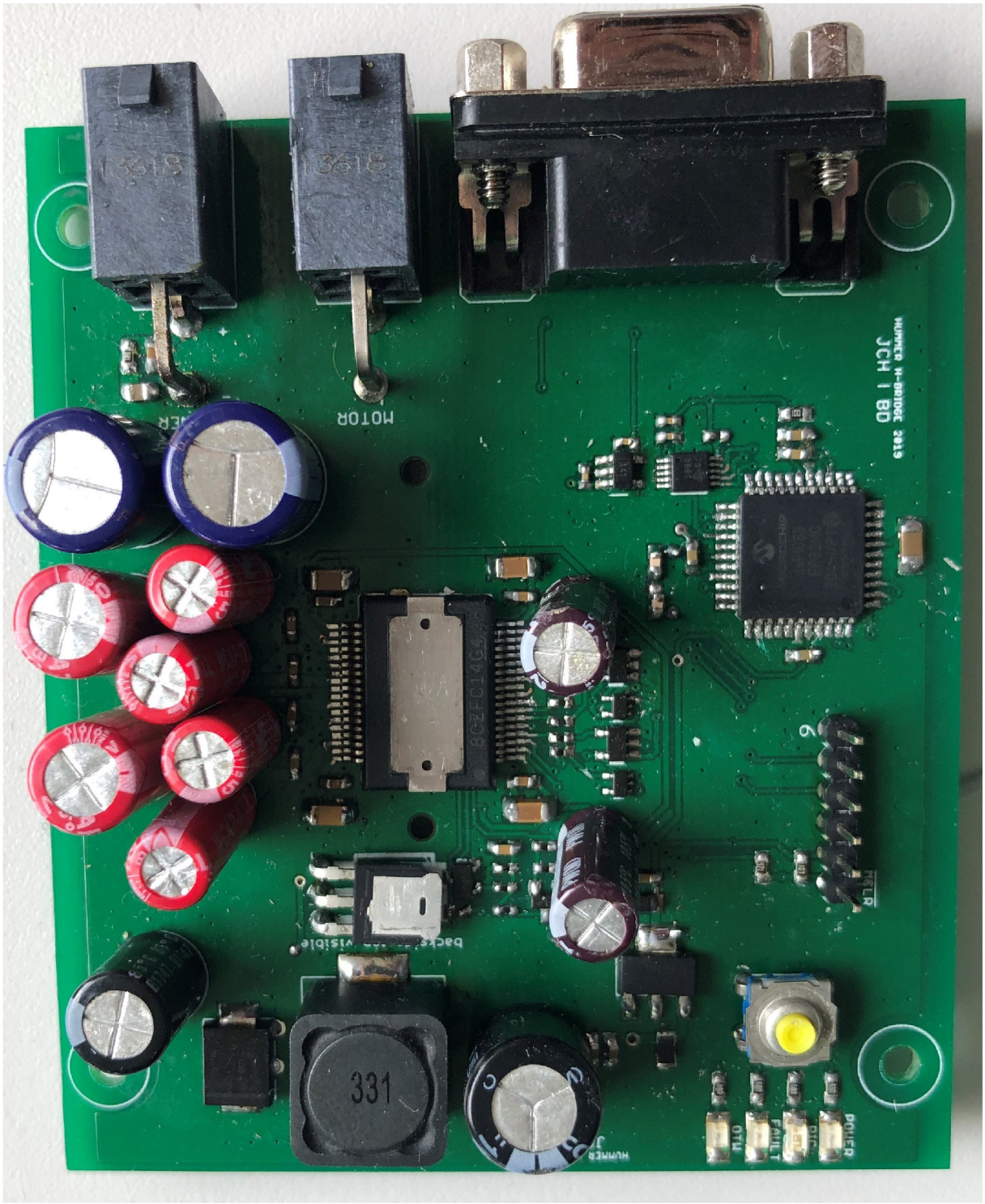


Figure 6.3: The new H-Bridge Board

PIN No.	Signal	Description
1	GND	ground
2	PICST_MB	H-Bridge status pin
3	SDI_MB	serial data input
4	DIR_MB	motor's spinning direction
5	VCC	supply voltage $3.3V$
6	PWM_MB	PWM input
7	SCK_MB	serial clock input
8	SDO_MB	serial data output
9	DIS_MB	disable H-Bridge

Table 6.1: H-Bridge Board PIN mapping table

ISO Board

During the tests of the H-Bridge Board on the vehicle, some unexpected problems were encountered, which resulted in oscillating rotation of the wheels and insufficient reaction to changes in duty cycle of the PWM and in direction signal. This kind of malfunctioning was not encountered during the tests on the test bench on the same motor.

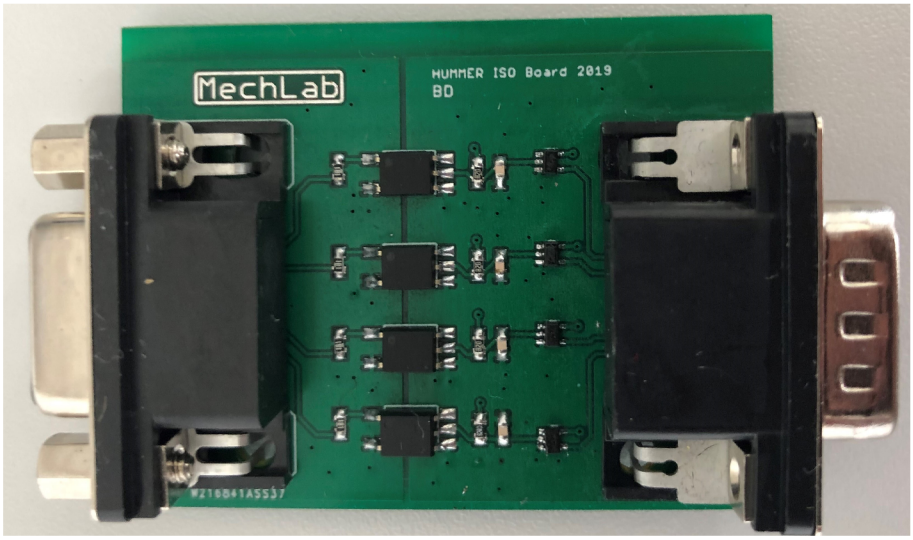


Figure 6.4: The Isolation Board for galvanic separation

The problems were first of all caused by noise, which was induced on the motors and through return paths of the ground got transferred to the sensitive control signals.

Two different approaches were tested to solve this problem one of them included the replacement of normal cables, which were used for transmission of the control signals from the Main Board to the H-Bridge Board with shielded cables. The second approach used

galvanic isolators to separate the two printed circuit boards. Opto-isolators were chosen as they passed the following criterion:

1. the maximal frequency of the control signal will not be higher than $1kHz$
2. only one side requires external voltage supply

As a result of experiments on the vehicle the solution with opto-isolators came out on top. It proved to be more robust and reliable than the shielded cables.

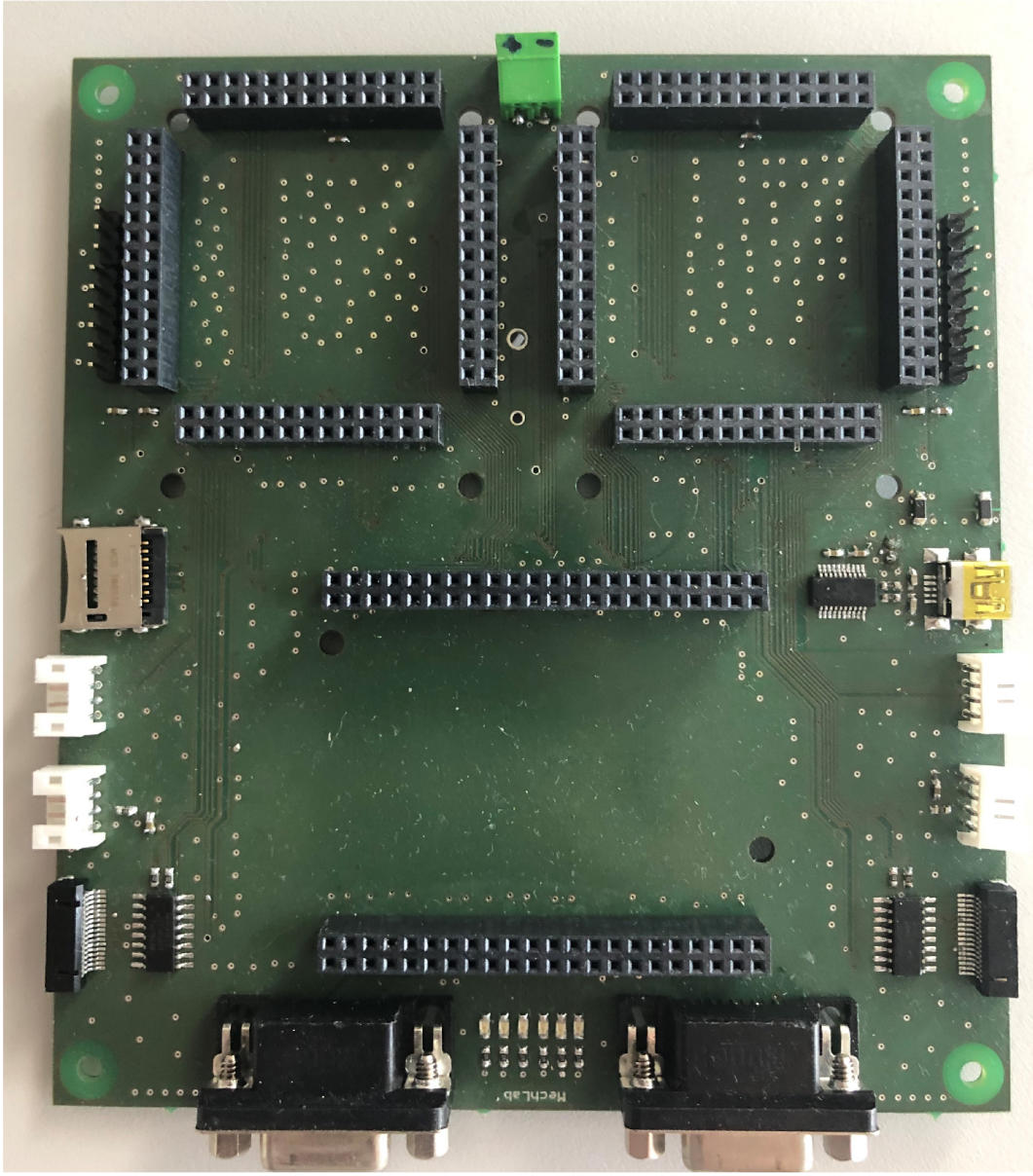


Figure 6.5: The Main Board with the new features

6.3.2 Main Board

The Main Board serves as the junction point of the whole hardware. Among its main functions are supplying individual parts of the electronics, which among others include the sensors and MCUs and also data transmission between the different units.

Compared to the previous version from [5] the new PCB was extended by an *FTDI-231x* device, which is a serial UART to USB interface. Thanks to this upgrade the available information about the state of the vehicle can be transferred to computer or other devices with USB interface and it can be utilized for monitoring purposes or just for improving user experience.

An SD memory card reader was also added to the board for data logging. Both of theses new features can be accessed from the MCU unit marked *ECU-1*.

The power grid on the board was replaced as a whole. Instead of three separated zones, each equipped by its own 3.3 V voltage regulator supplying one individual processing unit (previous version), only one voltage regulator was used to supply the whole board and the related sensors. This change was made possible by discarding the galvanic isolation between the processing units, as its application was superfluous. Except of the 3.3 V voltage network an another 5 V grid is used to supply the encoders.

Two *JAE FI-W17S* connectors are used to interface the *CUI Inc. AMT112* encoders. Signals from the encoders are available for both of the MCU units for processing.

The 15 pin Cannon D-sub connector from the previous Main Board was replaced by 9 connector of the same type, the PIN mapping is available at Table 6.1.

6.4 Sensor Upgrade

In the previous section changes in the PCB were presented. This part is devoted to the upgrades in the sensory part of the vehicle.

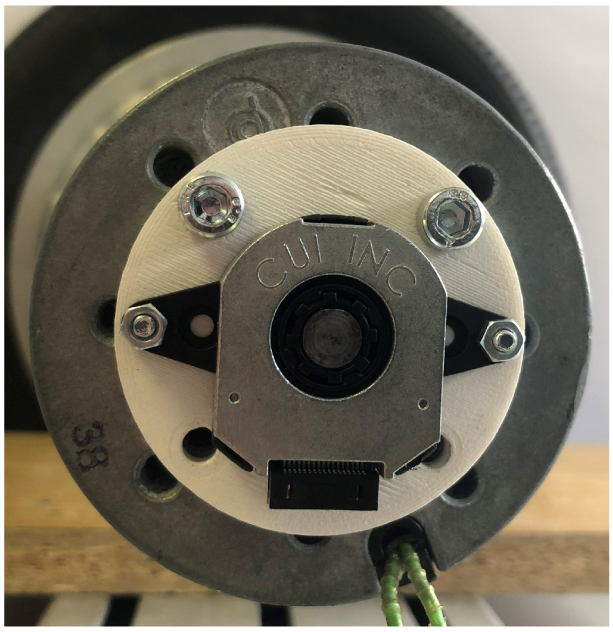


Figure 6.6: Capacitive encoders after mounting on the motor shaft

In case of the encoder the most demanding requirement regarded its physical dimensions as the space on the chassis between the two motors was very tight. The maximal allowable height of the encoder was 20 mm.

On the other hand the robustness condition discarded the optical type as vibration, dust would affect its precision and durability, therefore it was necessary to look for a

more suitable replacement. Magnetic and capacitive encoders were examined as they offer better properties in case of dust resistance and durability. Moreover these technologies are also resistant against disturbance from electro-magnetic fields. As both of the presented technologies offered the same features from the technical viewpoint, the decision was made based on their price.

Capacitive encoder from *AMT112* with *256 CPR* from *CUI Inc.* was chosen. For this application the 256 counts per revolution was more than enough regarding the fact that the motor has a gear box with $9 : 1$ gear ratio and quadrature processing mode is also available. As result of that the available resolution was more than *9000 CPR*.

The mounting of the encoder on the motor was challenging as the motor's mounting holes did not match the holes on encoder's base. This was solved by inserting a custom made 3D printed mounting pad between the motor and the encoder.

7 Software

This section gives an overview about the implementation of the proposed software solution presented and tested in the Section 5.3 and 5.4. According to the plan the solution that was implemented is an adaptive pole placement controller, which consisted of an adaptive law realized by the Newton-Raphson method and a control law carried out by a gain scheduling-like approach.

In Section 5.4 the plan regarding the organization of the individual algorithms was presented. According to the plan the software will be divided between the electronic control units on the Main Board and the H-Bridge Board in the following way:

- **H-Bridge Board**
  - current(moment) control loop
- **Electronic Control Unit (ECU)**
  - stabilization and cornering control
  - parameter estimator and gain adjustment mechanism
  - safety features (system failure detection)
  - signal processing (sensor fusion, filtering)

Next in this chapter the deployment of the individual parts will be described. In the first section the realization of the current (moment) regulator is described. The second section will be devoted to introduce the software implemented on the ECUs.

7.1 Current Control Loop

This section discusses the development of the current control loop for the micro-controller on the H-Bridge Board.

To simplify and accelerate the implementation, the controller was first developed and tested in software simulations. For this purpose Simulink model of the motor was created based on the previously derived mathematical model Equation 4.1 and 4.2 and the estimated parameters from Table 4.2. To control the current loop a *PI regulator* was chosen, for the following reasons:

- DC motor with the non-conservative forces is a slightly non-linear system
- ease and simplicity of implementation
- no readjustment of the regulator during operation is required
- good performance

After the controller was tuned and the tests produced the desired output it was deployed on the micro-controller situated on the H-Bridge Board. But first the continuous time regulator was converted to discrete time with sampling time of 100 μs. The program on the MCU was written in *C language*.

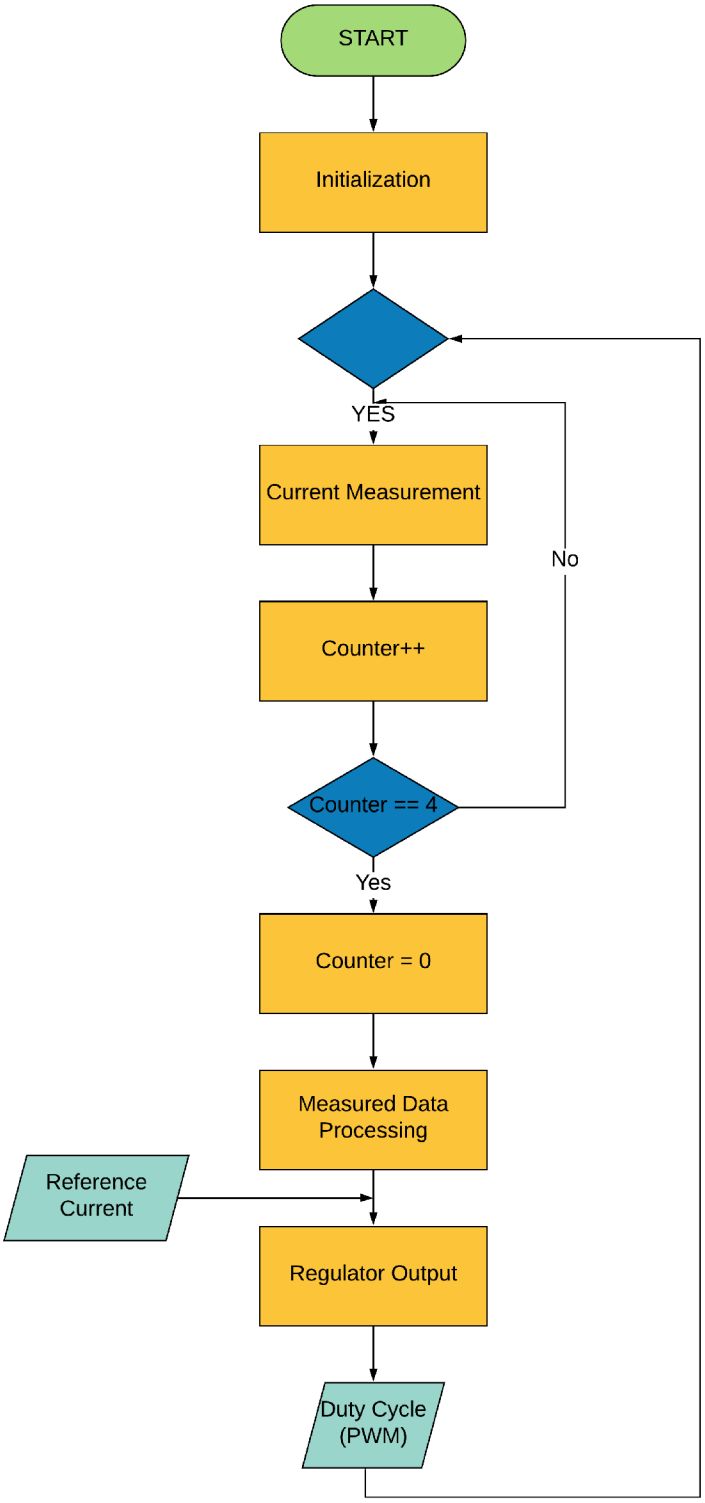


Figure 7.1: Flowchart of the program on the H-Bridge

7.1.1 Implementation of the Controller

In Figure 7.1 the flowchart of the implemented current control loop is displayed. In the following, parts of the code that require more attention will be described in detail.

Data Transfer on the PCB

To realize the current control loop, first the data transfer between the ECUs and the H-Bridge Board and between the individual integrated circuits (ADC, motor driver, micro-controller) on the board had to be established.

- **Communication between the ECU and the micro-controller on the PCB**
  - three data lines:
    - Enable
    - Reference current value encoded as PWM signal
    - Polarity of the reference current
- **Communication between the micro-controller and the ADC**
  - 3-Wire SPI
    - One way data transfer from the ADC to the MCU (*MISO*)
    - Clock (*CLK*)
    - Chip Select (*CS*)
- **Communication between the micro-controller and the motor driver IC**
  - three data lines:
    - Enable
    - Duty Cycle
    - Direction

Current Measurement

Another requirement on a high-class current controller is to ensure the quality of the measured current, which mostly depends on the physical realization of the hardware, but it can also be improved by the way how the signal is processed. In the previous chapter in Section 6.3.1 the hardware part has already been discussed, in the following the software solution will be presented.

In case of current loop it was vital to ensure that the measured current was the average value of the current on the motor. Therefore the first step in configuring the current measurement was to select the right type of PWM mode. Centre aligned PWM mode was chosen due the symmetrical nature of the modulation signal, which is advantageous in motor control applications. Thanks to the symmetry the peak value of this signal occurs at the same time as the current on the motor reaches its average value, this phenomenon is also depicted in Figure 7.2. As a result no further processing was required to obtain the sought mean value.

As the peak value of the modulation signal was reached an interrupt routine was called, that triggered the current measurement by setting the *CS* signal of the ADC to logical high level. The signal was kept high until the digitization of the sample was not finished, then the data was transferred to the MCU, where its processing took place. More information on the centre aligned mode and on the operation of the ADC can be found in [31] and [32].

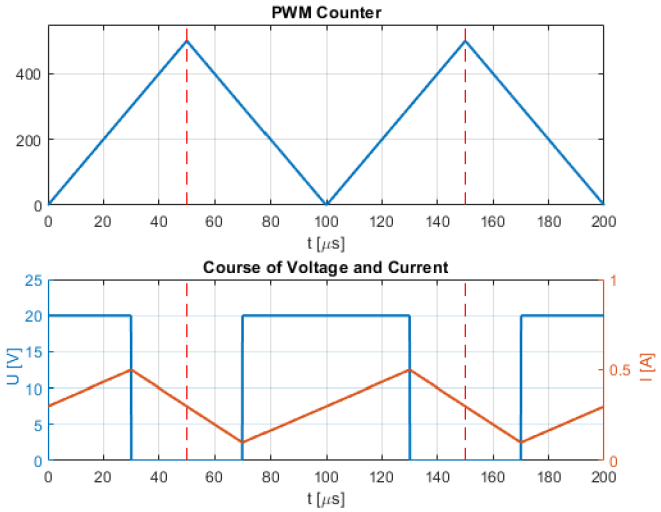
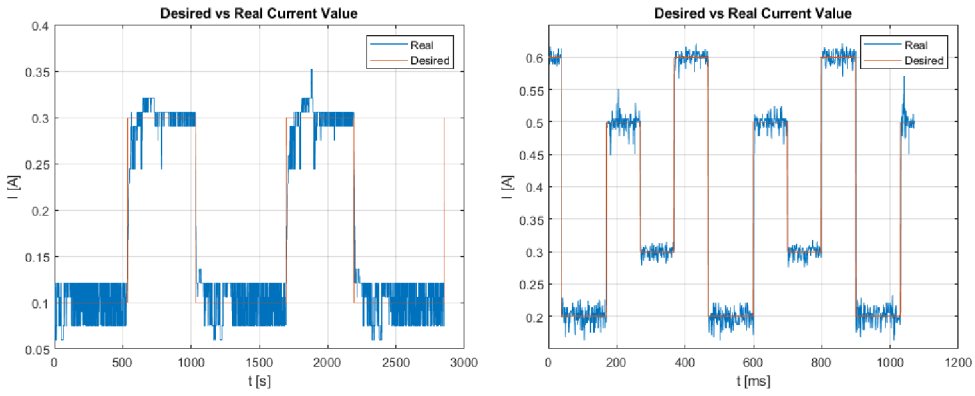


Figure 7.2: Illustrative course of voltage and current in centre aligned PWM mode, the vertical dashed lines mark the sampling instants

During the first experiments the measurements were taken at the same rate as the calculation of the current control loop took place, which was at the rate of 10 kHz, however, the measured data was of bad quality as it was highly distorted by measurement noise. To solve this problem the frequency of the measurements was increased fourfold to 40 kHz. Then these four new samples that resulted from the increase of the speed of sampling were replaced by their mean value, as a result of that the signal was smoother and the noise was decreased and also the desired update rate of 10 kHz of the current regulation loop was kept.



(a) (b)

Figure 7.3: The measured current signal (*a*) before and (*b*) after averaging

PI Controller

As controller the previously developed PI regulator was used. This controller structure was further extended by *anti-windup* to reset the integrator in case it started saturating.

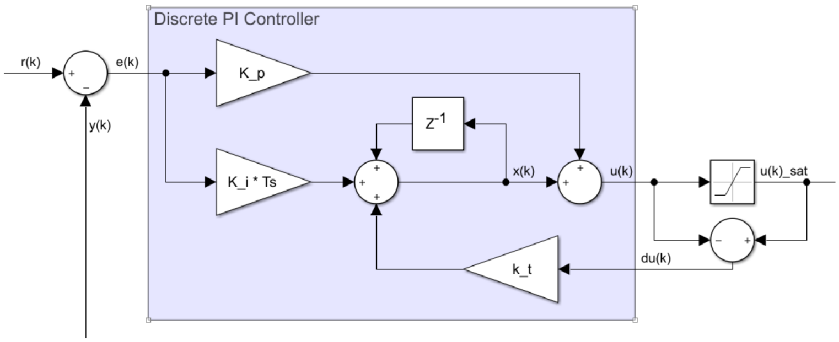


Figure 7.4: PI regulator with anti-windup schematics. Adapted from [33]

To avoid undesired behaviour of the anti-windup, which can happen if the anti-windup gain is too large, it was set according to the formula presented by[8]:

$$k_t = \frac{1}{K_i} \tag{7.1}$$

where K_i = integral gain
k_t = anti-windup gain

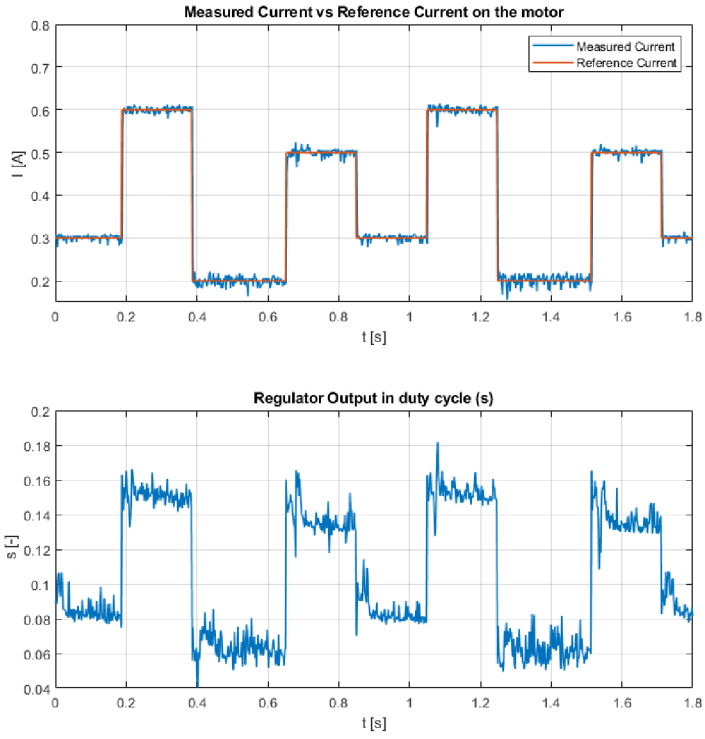


Figure 7.5: Response of the DC motor controlled by PI regulator with anti-windup.

After applying the regulator which was developed in simulation an additional fine tuning of the coefficients was yet required. However, this fine-tuning was carried out on the real hardware with the motor, which was placed on a test bench for this occasion. During the tuning the motor was braked and its angular velocity was kept at zero, therefore it behaved like a resistive load and the set-up of the gains was made simpler and more accurate. The Figure 7.5 presents the course of the current on the motor for sequence of step inputs.

7.2 Software Implementation on the ECU

This section builds on the various sections of the previous chapters, namely 5.2.2 and 5.3 that already discussed the selected solution.

This section discusses the implementation of the new elements of the control algorithm on the electronic control units. Before any adjustments were made first the original firmware was examined with the sole purpose of identifying parts of the code that did not require any modification and were compatible with the new algorithm. The parts that were preserved from the previous version included:

- safety features (system failure detection)
- signal processing (sensor fusion, filtering)

Further description of the preserved parts is out of the scope of this thesis, more detailed information can be found in [6].

Next in this section the implementation of the following algorithms will be discussed:

- stabilization and cornering control
- parameter estimator and gain adjustment mechanism

7.2.1 Description of the new Algorithms

The flowchart in Figure 7.6 depicts the whole algorithm that was implemented on the ECUs including the parts that were part of the previous theses. The code was created in the Simulink environment by utilizing a special library called MPLAB Device Blocks for Simulink to interface the peripherals and other functionality of the MCU.

Control Algorithm

In the first step regulator with constant feed-back gain matrix was implemented for the following two reasons

- to see whether the gains designed by the pole placement method produced satisfactory response
- to test the precision of the estimation algorithm

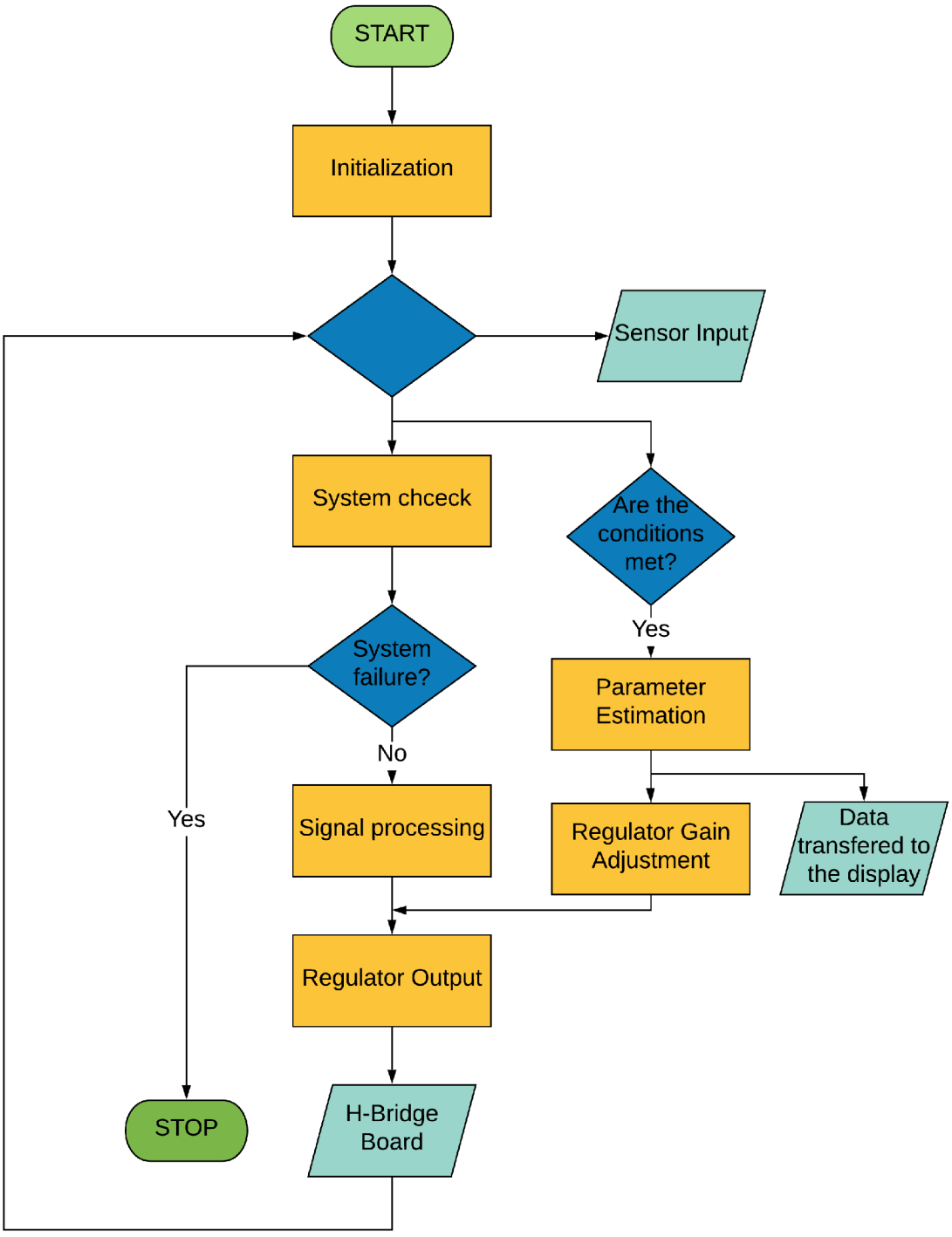


Figure 7.6: Flowchart of the program on the ECU

The poles obtained during SIL simulations proved to be too aggressive, while it evoked oscillations as a result of that the vehicle could not keep its stability. Therefore adjustments had to be made, which resulted in changing the position of the poles. While making adjustments to them it was imperative not to choose complex poles, but poles solely with real part, while the addition of imaginary parts would have amplified the oscillatory behaviour.

By moving one of the poles closer to the imaginary axis, which resulted in decreasing

its response speed, the problem with the undesirable oscillations was solved. In general it was also important to select such poles that can stabilize the vehicle with and without the driver.

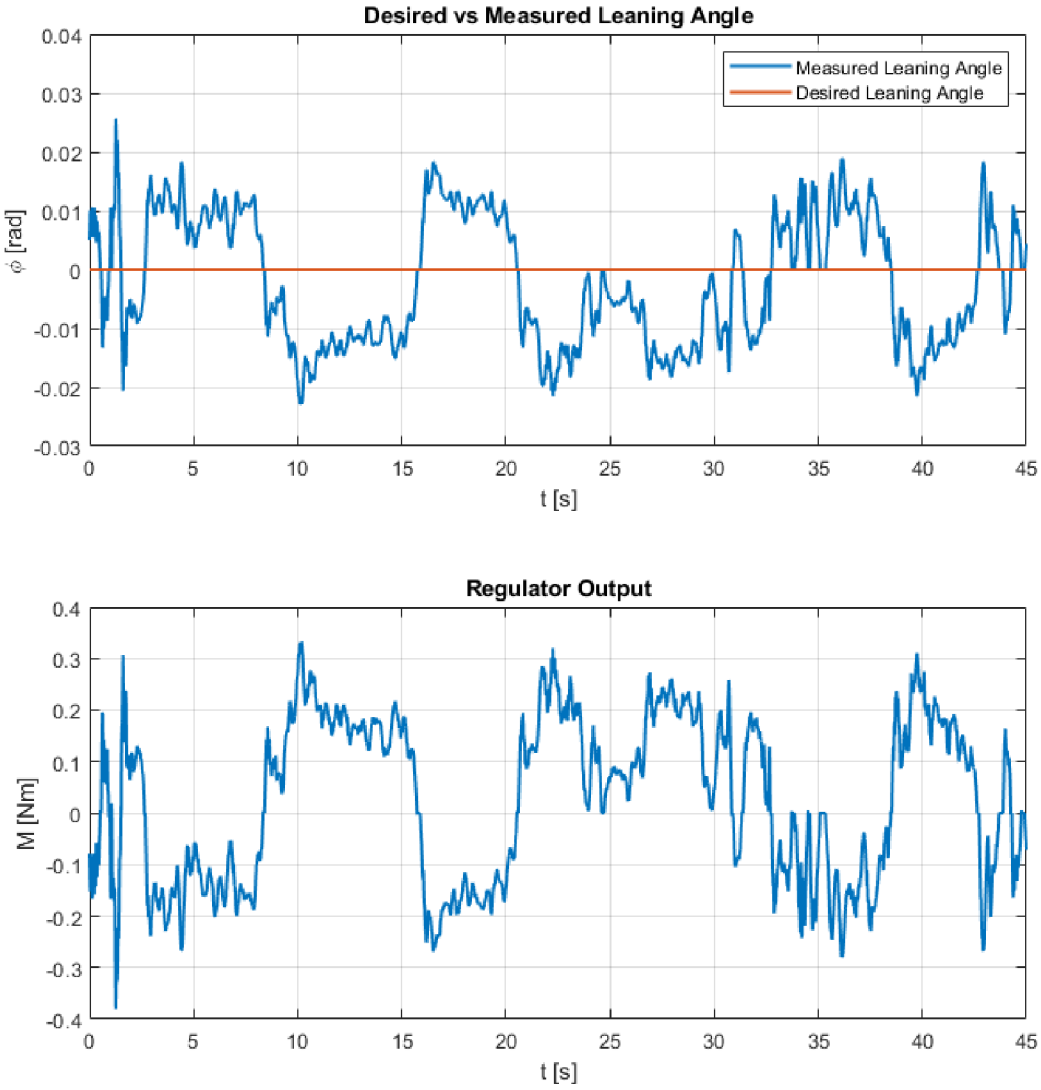


Figure 7.7: Response of the of the system

After the controller was fine-tuned driving data was acquired first of all to see whether the regulator is working as desired, secondly to use it for configuring the parameter estimation algorithm. The controller's response during operation is displayed in Figure7.7.

Estimator algorithm

Based on the measured data the parameter estimation algorithm was fine-tuned. The fine-tuning procedure included the selection of the right cut-off frequency of the filters that were used to compute numerical derivatives of the leaning angle and the encoder's angular displacement, which were then used as input to the parameter estimator. The

chosen cut-off frequency was the same as the frequency of calculating the estimates which is 2 Hz.

As it was mentioned in the Section 5.3 due to unmodeled effects, e.g. the force exerted by the driver to evoke forward/backward movement, the parameters can not be estimated accurately. To overcome this problem the estimation took place only at times when the influence of neglected forces on the model was small and the estimated parameters were accurate. That means, that the estimation was carried out when the value of angular acceleration was between a predefined upper and lower bound. However, on the real system the limits had to be tighter than the results of the SIL simulation it previously suggested. Therefore, readjustments in the bound were necessary.

After the bounds were set the desired accuracy of ±10 % of the driver's mass was reached, however the estimated height of centre of gravity remained inaccurate, just like the results of previous experiments predicted it. Therefore, the regulator's feed-back gain adjustments were based exclusively on the estimated mass.

Implementation of the Gain Adaptation Mechanism

As it was discussed in Section 5.3 the adaptation mechanism was limited to a gain scheduler-like solution. The following three intervals were created based on the driver's body mass:

- **from 0 to 50 kg**
- **from 50 to 75 kg**
- **from 75 to 120 kg**

To improve user safety the feed-back gains in each interval were tested on the real vehicle.

Communication with the Display

One of the new features of the main board design was the addition of USB interface to implement data transfer between the ECU and an external display.

To interface the USB an FTDI IC was added which communicates with the ECU via UART. Next, the created communication protocol will be briefly described.

Start Byte	DATA 1	DATA 2	DATA 3	DATA 4	Stop byte

Figure 7.8: Communication protocol between the ECU and display

where **Start** = 13
Data 1 = translational velocity
Data 2 = leaning angle
Data 3 = mass of the driver
Data 4 = height of the driver
Stop = 17

Initially to approximate driver's height the estimated position of his or her centre of mass was intended to be used, however due to its inaccuracy this idea was abandoned. The proposed solution builds on the fact that the humans' height and their body mass are related. The formula describing that relationship was derived based on the data presented at the website [34].

$$H = 0.935 \cdot m_h + 107 \tag{7.2}$$

where H = is the driver's approximated height.

8 Conclusion

The goal of this thesis was to implement an adaptive controller to control and stabilize the two-wheeled self-balancing inverted pendulum-like vehicle, nicknamed Hummer and also to verify the idea of indirect measurement of the driver's parameters.

The applied solution can be divided into three independent subtasks namely the estimation of unknown parameters, the adaptation mechanism and the regulator that stabilizes the vehicle in upright position and ensures control over cornering.

The parameter obtainment was achieved by implementing the Newton-Raphson method which is a gradient based optimization algorithm. Due to some unmodeled effects the estimation algorithm operated correctly only under specific circumstances. This malfunctioning was first noticed during the SIL tests of this algorithm where also a solution was proposed, which was based on estimating the parameters only if the value of the angular acceleration of the leaning angle was between certain upper and lower bounds. This solution proved to be effective even on the real vehicle where it estimated the body mass of the driver with ± 10 % accuracy. Even though the inclusion of these constraints the estimated value of the position of centre of mass remained inaccurate.

The adaptation mechanism was realized by a gain scheduler-like structure, which distinguished three different operating states based on the driver's body mass with each state having its own specific gain value. The feed-back gains' of the regulator responsible for stabilizing the vehicle were readjusted after the new estimate of the driver's unknown parameters was obtained. This new solution compared to the original controller brought improvements into the quality of regulation by making it more stable and the personalization which was made possible by adaptation mechanism made the driving of this vehicle more comfortable.

To be able to implement such a complex algorithm upgrade in parts of the hardware was also required. The application of the estimation algorithm required the measurement of quantities related to the motion of the wheels, therefore the vehicle was equipped with a capacitive encoder.

During the works a new PCB with an integrated H-Bridge module and a high quality current sensor to control the DC motors was designed. This PCB was also equipped with a micro-controller unit, which served as an on-board current controller. During the overhaul the main board was also redesigned. Its new features include improved data transfer and logging capabilities represented by the new USB port and SD card reader the improvements of the sensory part and the complete overhaul in its power supply.

8.1 Suggestions for Future Development

The future development might concern the following parts of the vehicle :

- **Hardware**
  - replacement of the current *dsPIC33EP512MU810* MCU with higher perfor-

mance 32-bit micro-controller
  - addition of state of charge monitoring of the batteries
  - upgrade of the power electronics, which supplies the main board
  - addition of display for monitoring purposes
- **Software**
  - improvement of the estimation algorithm
    * increase of the number of samples required to calculate the parameter estimate
    * implementation of non-linear observer algorithm, e.g., Extended Kalman filter to estimate the currently unmodeled force exerted by the driver and use this force estimate as another input parameter of the model used for the unknown parameter estimation

List of Abbreviations

ADC	Analog-to-digital Converter
APPC	Adaptive Pole Placement Control
ECU	Electronic Control Unit
IC	Integrated Circuit
IMU	Inertial Measurement Unit
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
MCU	Micro Controller Unit
MIT	Massachusetts Institute of Technology
MRAC	Model Reference Adaptive Control
NLS	Non-linear Least Squares
PCB	Printed Circuit Board
PID	Proportional-Integral-Derivative
SIL	Software-in-the-Loop
SPR	Strictly Positive Real
UART	Universal Asynchronous Transmitter-receiver
USB	Universal Serial Bus

Bibliography

[1] NITCH SMITH, Matthew. The number of cars worldwide is set to double by 2040 [online]. 2016 [cit. 2019-04-06]. Dostupné z: https://www.weforum.org/agenda/2016/04/the-number-of-cars-worldwide-is-set-to-double-by-2040

[2] GORMAN, Steve. Elon Musk unveils his first Los Angeles-area tunnel. Reuters [online]. 2018, 19.12.2018 [cit. 2019-04-06]. Dostupné z: https://www.reuters.com/article/us-musk-tunnel/elon-musk-unveils-his-first-los-angeles-area-tunnel-idUSKBN1OI03W

[3] ŠTĚPÁNEK, Jan. Identifikace systému, sensorika a implementace řídicího algoritmu pro nestabilní balancující vozidlo. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2011.

[4] ZOUHAR, František. Návrh konstrukce, řízení a elektroniky pro nestabilní balancující vozidlo. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2011.

[5] BASTL, Michal. Návrh elektroniky pro řízení dvoukolového nestabilního vozidla. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2015.

[6] MATĚJÁSKO, Michal. Návrh bezpečného řídicího systému pro dvoukolové balancující vozidlo. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2015.

[7] PID Theory Explained. National Instruments [online]. 19.3.2019 [cit. 2019-04-08]. Dostupné z: www.ni.com/cs-cz/innovations/white-papers/06/pid-theory-explained.html

[8] MURRAY, Richard M. PID Control [online]. s. 22 [cit. 2019-04-08]. Dostupné z: www.cds.caltech.edu/∼ murray/courses/cds101/fa04/caltech/am04 _ch8-3nov04.pdf

[9] ASTRÖM, Karl Johan a Richard M. MURRAY. Feedback Systems. Woodstock, Oxfordshire: Princeton University Press, 2008. ISBN 978-0-691-13576-2.

[10] SANKALP, Paliwal. Stabilization of Mobile Inverted Pendulum System Using Conventional and Fuzzy PID Controllers [online]. 2018 [cit. 2019-04-08]. Dostupné z: file:///C:/Users/Admin/Downloads/finalest.pdf. Phd thesis. Birla Institute of Technology and Science Pilani.

[11] HORÁK, Petr. Řízení laboratorního modelu nestabilního balancujícího vozidla. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2011.

[12] The DIY Segway. Massachusetts Institute of Technology [online]. 7.17.2008 [cit. 2019-04-08]. Dostupné z: http://web.mit.edu/first/segway/

[13] OGATA, Katsuhiko. Modern Control Engineering. 5th Edition. Pearson Education, 2009. ISBN 9780137133376.

[14] Introduction: State-Space Methods for Controller Design. Control Tutorial for Matlab and Simulink [online]. [cit. 2019-04-12]. Dostupné z: www.ctms.engin.umich.edu/CTMS/index.php?example=Introduction§ion=ControlStateSpace

[15] SKALICKÝ, Jiří. Teorie řízení. Druhé. Vysoké učení technické v Brně Fakulta elektrotechniky a komunikačních technológií Ústav výkonové elektrotechniky a elektroniky, 2002. ISBN 8021421126.

[16] GREPL, Robert. Balancing Wheeled Robot: Effective Modelling Sensory Processing and Simplified Control. Engineering MECHANICS [online]. 2009, 16(2), 141-154 [cit. 2019-05-09]. Dostupné z: http://www.engineeringmechanics.cz/pdf/16_2_141.pdf

[17] IOANNOU, Petros a Jing SUN. Robust Adaptive Control. Dover Publications, 2013. ISBN 9780486498171.

[18] Caldwell, W.I., Control System with Automatic Response Adjustment. American patent, 2,517,081. Filed 25, April 1947, 1950.

[19] MCRUER, Duane T., Irving Louis ASHKENAS a Dunstan GRAHAM. Aircraft Dynamics and Automatic Control. Princeton University Press, 2014. DOI: 10.2307/j.ctt7ztqhj. ISBN 0691080836.

[20] KANEHARA, Akira a K.Z. LIU. A Gain-Scheduling Control Method for Automotive Vehicle Stabilization. Transactions of JSME. 2007, 73, 1285-1290. DOI: 10.1299/kikaic.73.1285.

[21] PRIYANK, Jain a M.J. NIGAM. Design of a Model Reference Adaptive Controller Using Modified MIT Rule for a Second Order System. Advance in Electronic and Electric Engineering. 3(4), 477-484. ISSN 2231-1297.

[22] NORTH, Gerald R., J. C. PYLE a Fuqing ZHANG, ed. Encyclopedia of Atmospheric Sciences. 2. Elsevier Science, 2014. ISBN 9780123822253.

[23] Parameter Estimation: Estimate model parameters. Mathworks [online]. [cit. 2019-04-16]. Dostupné z: https://www.mathworks.com/discovery/parameter-estimation.html

[24] NELLES, Oliver. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. New York: Springer, c2001. ISBN 35-406-7369-5.

[25] Finding Roots. Caltech: The Division of Physics, Mathematics and Astronomy [online]. [cit. 2019-05-22]. Dostupné z: www.pmaweb.caltech.edu/ physlab/lab_22_current/Ph22_0_Finding_Roots.pdf

[26] Constrained Nonlinear Optimization Algorithms. MathWorks [online]. [cit. 2019-04-20]. Dostupné z: www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html

[27] GR 80x80 —. Dunker Motoren [online]. [cit. 2019-04-26]. Dostupné z: https://www.dunkermotoren.com/uploads/tx_products/downloads/MKS/gr-80x80-8844501502.pdf

[28] DRV8432: 12A Dual Brushed DC or Single Bipolar Stepper Motor Driver. Texas Instruments [online]. [cit. 2019-04-23]. Dostupné z: www.ti.com/product/DRV8432#

[29] PRAKASH, Arjun. Current Sensing in an H-Bridge: Current Sensing Products [online]. [cit. 2019-04-23]. Dostupné z: www.ti.com/lit/an/sboa174a/sboa174a.pdf

[30] Holonomic and Nonholonomic Constraints [online]. In: . University of Pennsylvania [cit. 2019-05-07]. Dostupné z: https://alliance.seas.upenn.edu/ meam535/cgi-bin/pmwiki/uploads/Main/Constraints10.pdf

[31] 16-Bit Microcontrollers and Digital Signal Controllers with High-Speed PWM, Op Amps and Advanced Analog [online]. In: . s. 530 [cit. 2019-05-16]. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/70000657H.pdf

[32] ADS8866 16-bit, 100-kSPS, serial interface, micropower, miniature, single-ended input, SAR analog-to-digital converter [online]. In: . [cit. 2019-05-16]. Dostupné z: http://www.ti.com/lit/ds/symlink/ads8866.pdf

[33] Discrete PI Controller: Discrete-time PI controller with external anti-windup input. Mathworks [online]. [cit. 2019-05-19]. Dostupné z: www.mathworks.com/help/physmod/sps/ref/discretepicontroller.html

[34] Adult Male and Female Height to Weight Ratio Chart. Disabled World [online]. [cit. 2019-05-16]. Dostupné z: www.disabled-world.com/calculators-charts/height-weight.php

Appendix

The thesis contains the following electronic appendixes:

- PCB layouts:
  - Main Board
  - H-Bridge Board
  - Isolation Board
- Algorithms used during the SIL tests:
  - Parameter Estimation
  - Controller Synthesis
  - Adaptive Controller