VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

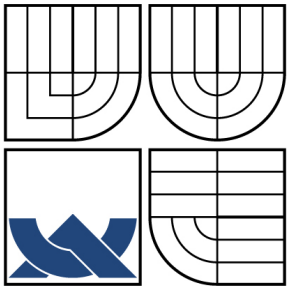
FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

MOTION CONTROL OF TWO-WHEEL ROBOT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

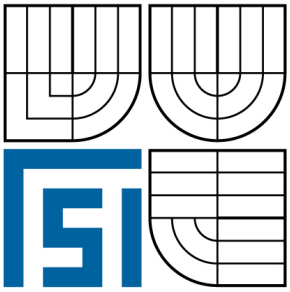
AUTOR PRÁCE JOSEF VEJLUPEK
AUTHOR

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

FACULTY OF MECHANICAL ENGINEERING

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

MOTION CONTROL OF TWO-WHEEL ROBOT

ŘÍZENÍ POHYBU DVOUKOLOVÉHO ROBOTU

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — JOSEF VEJLUPEK

AUTHOR

VEDOUCÍ PRÁCE — Ing. PAVEL HOUŠKA, Ph.D.

SUPERVISOR

BRNO 2008

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav mechaniky těles, mechatroniky a biomechaniky
Akademický rok: 2007/08

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Vejlupek Josef

který/která studuje v **bakalářském studijním programu**

obor: **Mechatronika (3906R001)**

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 bakalářské práce:

Řízení pohybu dvoukolového robotu

v anglickém jazyce:

Motion control of two-wheel robot

Stručná charakteristika problematiky úkolu:

Cílem práce je navrhnout a simulačně ověřit možnosti řízení pohybu dvoukolového balancujícího robotu.

Cíle bakalářské práce:

1. Prostudujte problematiku konstrukce a řízení dvoukolových robotů,
2. Realizujte zjednodušený dynamický model realizovaného dvoukolového robotu pro potřeby řízení,
3. Na základě dynamického modelu robotu navrhněte řízení pohybu robotu,
4. Navržené řízení simulačně ověřte.

Seznam odborné literatury:

[1] http://www.tlb.org/scooter.html
[2] http://www.barello.net/Robots/gyrobot/index.htm
[3] http://www.systron.com/
[4] http://www-robotics.cs.umass.edu/Robots/UBot

Vedoucí bakalářské práce: Ing. Pavel Houška, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2007/08.

V Brně, dne 21.11.2007

L.S.

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA STROJNÍHO INŽENÝRSTVÍ DĚKANÁT

prof. Ing. Jindřich Petruška, CSc.
Ředitel ústavu

doc. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

Abstrakt

Tato práce se zabývá návrhem a simulačním ověřením možností pro řízení pohybu dvoukolého balancujícího robotu. Obsahem práce je rovněž rešeršní studie zaměřená na již existující projekty.

Abstract

The goal of this work is to design motion control of two wheeled mobile robot. Part of this work is a literature research oriented on balancing robots design in both commercial and non-commercial sector.

Thanks

I would like to thank to my supervisor Ing. Pavel Houška, Ph.D. for his patience and guidance.
To my family for support and encouragement during my studies.

Declaration on word of honour

I statutory declare, that I wrote this paper by myself with usage of stated literature and under supervision of my instructor.
Josef VEJLUPEK, Brno, 2008

Contents

1 Introduction

Balancing two-wheeled robot is basically what is known as an `Inverted pendulum` problem. Inverted pendulum is inherently unstable system because its center of gravity (`CG`) is located above pendulum fulcrum. It may remain static in labile position if the `CG` is above the fulcrum. In 3D space it has 3 degrees of freedom (`DOF`), in 2D space 2-`DOF`. In addition to this, it is also an under actuated[1] system. The main goal is to keep the robot in `mechanical equilibrium`[2].

Motivation of this thesis is to prepare a model for further study of a nonlinear system, examine possible approaches for modeling and testing of nonlinear systems and their control development. Also we will briefly study character of used motion sensors in the literature search part. In future we should be able to use acquired experiences from this work to implement motion control on the real platform which we briefly introduce in the begining of chapter 3.

One of the most known application of an Inverted pendulum is Segway^{TM}human transporter[3]. And as it is very common tool for dynamic systems control, there is a lot of small "laboratory" prototypes starting with cart and a pole not ending with small two-wheeled robots. Some representatives will be presented in following chapter 2.

[1] Has fewer actuators than `DOF`s.

[2] A rigid body is in a mechanical equilibrium when the sum of all forces on all particles of the system is zero, and also the sum of all torques on all particles of the system is zero. A rigid body in mechanical equilibrium is undergoing neither linear nor rotational acceleration; however it could be translating or rotating at a constant velocity.

2 Literature search

2.1 Larry Barello's gyrobot

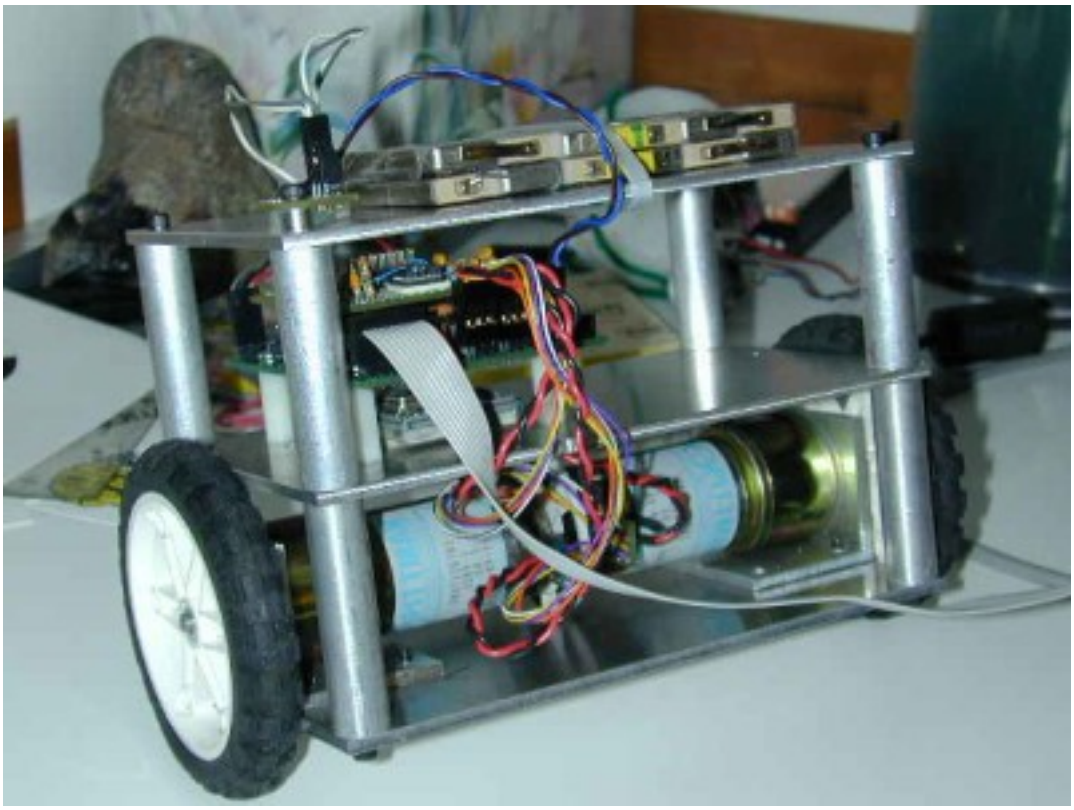


Figure 2.1: Larry Barello's balancing robotic platform[11]

This robot is quite simple, the sensoric array has some issues such as the gyro drift meaning that it can't run very long without reset of the zero-tilt angle. Control algorithm is basically simple `PID` regulator with fixed gains on all factors. Constants for the `PID` have been determined by trial and error method. Gyrobot is using gyroscope to get tilt rate, and ADXL2002 accelerometer to measure absolute tilt: Tilt angle can be measured from gyroscope by integrating its output, but the robot has no information referencing to any fixed reference coordinates frame (i.e.: ground). The concept is following: the robot starts in predefined position and tries to balance itself into a steady position, where the accelerometer indicates only g acceleration

in z direction. Last type of sensor is the odometry on motors done by quadrature encoders, which measure relative movement of the wheels and the body[1]. Mechanical base is made of aluminum sheets and tubes, simple but well done. Wheels have plastic hubs and rubber tires. As a microcontroller, there is an Atmel at90s8535 which is part of a whole kit developed for high school MiniSumo project.

There are some "future"[2] plans, that should lead to a better performance and make the platform more usable: Making the robot taller[3], integrating the tilt sensor output to get the angular displacement with respect to the ground, and replacing the velocity control with absolute motion control and position profiling.

[1] To get the distance traveled we have to include the tilt of the robot.

[2] The robot has been created in 2002, and there are no news on the webpage, so it can be assumed that the project has been abandoned.

[3] To be more specific: shifting the Center of gravity to make it more stable - we will show in section 3.2.2 equation (3.24) how it together with inertia affects behaviour of the system.

2.2 Trevor Blackwell's scooters

Figure 2.2: Trevor Blackwell's scooters[4]

These two platforms are a little specific, because they are man driven. Author states that the first version took him about a week to build and another week to tune it for performance. First version was build from "off-the-shelf" parts, which are mostly cheap and easy to get.

Version 1 (Figure 2.2 on left), built in 2002, was quite simple in construction, but was obviously an inspiration and evolutionary step for next model. Electronic and software parts are working fine, but they miss any redundancy or safety features. Mechanical construction is based on aluminum profiles and sheets, but the housing doesn't provide much shielding for electronics. Drives from powered wheelchairs were used to power the scooter. Motors are driven by power controller developed for Battlebots from RoboteQ[17], (both motors can drain up to 5kW). All computations are done by Atmel ATMEGA32 microcontroller. As for sensors, ceramic rate Gyro and 2-axis accelerometer set from Rotomotion[18] is used.

Control algorithm is quite simple, it is basically a `PID` controller which has only three inputs: angle, angle rate and steering knob.

Version 2 (Figure 2.2 on right), built in 2005, has better mechanical construction, also safety features had improved. There are several changes in sensorical array compared to first version. Mechanical construction is still based on aluminum profiles and sheets, but the housing is now closed up and therefore platform is suited for outdoor use. Whole platform is lighter, faster and also the range has increased. As for electronics there are changes also. The motor controller in first version communicated over serial line (7-bit), which caused delay in feedback loop. It was changed with two OSMC controllers which takes 9-bit PWM signal directly from microcontroller. Also the gyro was changed to CRS03-02 from Silicon Sensing Systems[14], and finally the ADXL102 accelerometer was changed for ADXL105, which has higher saturation threshold (5g, instead of 2g). These changes lead to a faster response of whole system resulting in tighter control of balance, and better overall performance. Safety features of second version has been also improved by several ways: First more powerful batteries has been used, enabling higher power drain in case of some peak, and battery monitoring has been added. The software speed limit has been also programmed in.

Both versions are nicely done pieces of complex engineering. Author showed great amount of invention and skill in creating something that would take a team of engineers to do. It is simpler parallel of Segway^{TM}, but the cost is lower and performance is better, however the Segway^{TM}has redudancy backups for safety reasons.

2.3 David P.Anderson's nBot

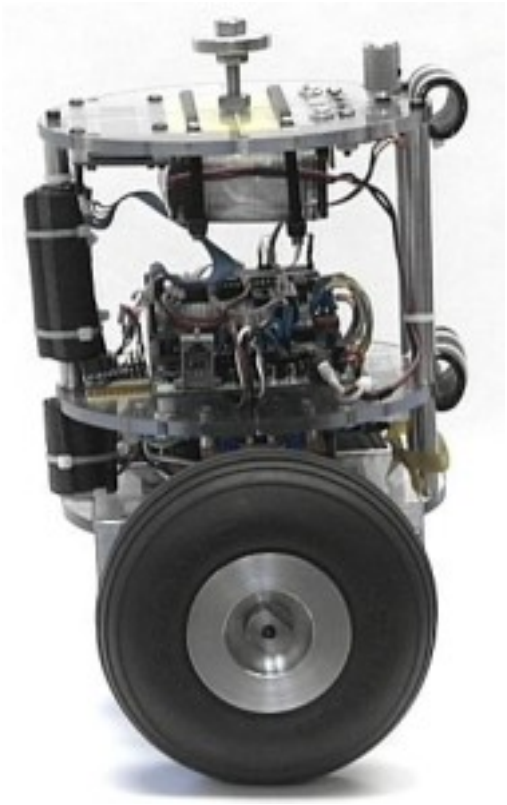


Figure 2.3: David P.Anderson's nBot[6], Revision 4

Evolutionary steps:

- Rev 1: Three wheeled robot with inverted pendulum on top.
  This was a platform designed to learn how to control an inverted pendulum. On top of a three wheeled robot was attached a pole topped with a ball. To the pivot of the pole was attached a low-friction potentiometer used for measuring the tilt angle.
- Rev 2: Two wheeled robot with feeler used to measure the tilt angle (still in contact with floor).
  On the bottom of two-wheeled robot was attached short aluminum feeler on a ball-bearing pivot, so robot can sense it's angle to the floor.
- Rev 3: Changes in construction.
  In this version, only some mechanical adjustments have been done, to shift `CG`

higher above the wheel axis, allowing robot to generate more torque without having to tilt over so much.
Mechanical feeler were first replaced by a piezo-electric gyroscope and iMEMS®[15] ADXL202 accelerometer, later by a commercial inclinometer from MicroStrain, the FAS-G[16].

- Rev 4: New mechanical construction of base.
  This version brought new motors: Pittman GM8712 DC geared motors with home made shaft encoders, and new batteries. This gave to nBot more torque at lower revolutions, which lead to greater stability and faster response.

This robot went through several evolutionary steps. Mechanical construction and drive control have been very precisely designed & manufactured. Control algorithm for balancing is quite simple: It is using tilt angle, speed, and derivate of these two variables, scheme could be found on a homepage of a project[6]. Changing speed is done by adding some offset to tilt angle, and steering is done by adjusting voltages on motors (voltage added on one, subtracted on other in a way it does not affect balance).

Mechanical parts done by author are carefully designed and custom made. Gear frame is made of aluminum as well as wheel hubs. Body frame is using acrylic sheets and aluminum distance posts. Robot has big rubber tires which allow him to do outdoor trips.

This robot was featured as NASA's Cool Robot of the Week for 19 May 2003. Thereafter Scientific American's online website, SCI/Tech Web Awards, honored the NASA page as one of the top 10 engineering and technical web sites for 2003, referencing nBot in its text. nBot is also featured in a new O'Reilly book spun off from Make Magazine in 2006, called The Makers.

2.4 Dan Piponi's Equibot

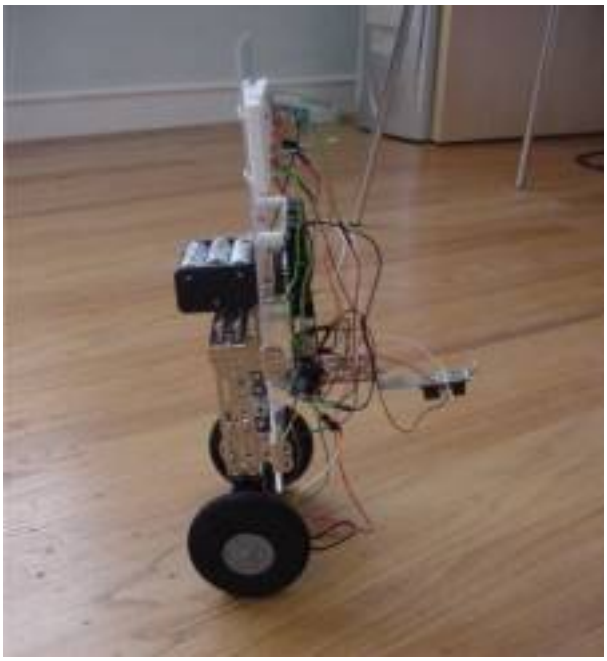


Figure 2.4: Dan Piponi's Equibot[9]

Equibot differs a little in the approach to the problem. It is not using any reference to gravity or acceleration. Instead it is using Sharp GP2D120 range sensor to detect tilt angle assuming that it is running on a flat horizontal floor. Any bigger obstacles or an inclined floor may cause problems with stability. This method could be helpful for some corrections when using accelerometers and gyro as main tilt sensors, ie. to correct the gyro sensor drift in long term.

- Simple mechanical construction: using prefabricated metallic profiles and acrylic sheet.
- Simple but purposive sensoric array: GP2D120 Infrared ranger from Sharp.
- Modified Hitec HS-311 servomotors.
- Atmel ATmega32 as microcontroller.
- 6 NiMh batteries, 5V regulator to power microcontroller and sensors.

There is PI controller programed on the ATmega, whole C code is available on web pages of this project[9].

2.5 Dean Kamen's iBOT and Segway

Figure 2.5: Dean Kamen's iBOT and Segway[1]

Dean Kamen is known in Czech Republic mostly for his Segway™, but more interesting and useful is his previous project called iBOT [1]:

> *The iBOT is a variety of powered wheelchair, developed by Dean Kamen in a partnership between DEKA and Johnson and Johnson's Independence Technology division. It is a medical technology, made to help people with severe mobility problems.*
> Wikipedia - iBOT [1]

As the iBOT and Segway™are both commercial projects, it is difficult to find any valuable informations for purposes of this work, but still it is worth mentioning as these two products initiated wider research in this course of study. I also believe that both products are important tools that can be really used in every-day life: iBOT is great aid for the disabled people. On the other hand, Segway™could replace cars in short distance trips, ie.: for shopping. As our cities are getting more and more crowded with cars, which burn lots of gas and pollute air, such vehicles do offer an alternative. While being much more environmentally friendly and efficient (power / space) they do sustain liberty of individual movement.

iBot main features are: Function to balance on two wheels, rising sitting person to eye level of other peoples, also enabling to reach higher objects. Also iBOT can climb 20cm high stairs without any external assistance, also it can go through rough terrains because of its massive four wheels. Another feature is a remote control. Maximum range is about 20km, top speed 11 km/h. iBOT is using patented iBALANCE®Technology which consists of multiple controllers and gyroscopes, whole system is built redundantly to assure the safety.

In 2000, Segway™ introduced their Segway Human Transporter (Segway™ `HT`). It is controlled by shifting the weight of the rider to go forwards or backwards and a knob to steer. Segway™ `HT` is using `PID` controller.

Primary sensor system is formed by a set of five gyroscopes, measuring pitch and pitch change rate of the platform. It needs only three of them, those two extra are for redundancy, to make the platform more reliable. In addition, Segway™has two electrolytic tilt sensors, these are similar to function of human inner ear: position is determined by tilt of a fluid (electrolit) surface.

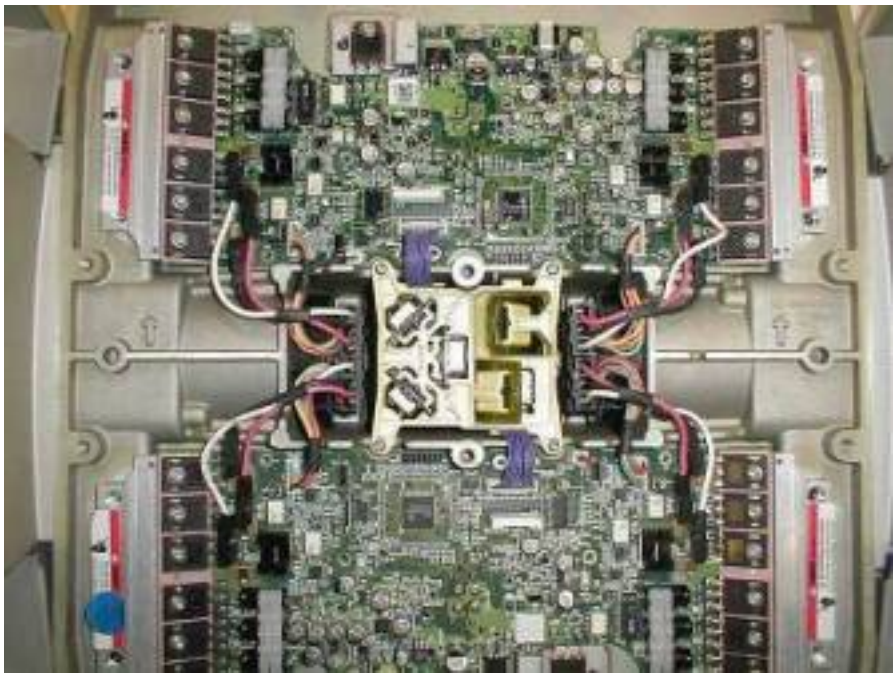


Figure 2.6: Detailed look on inside of Segway™[19]

2.6 Felix Grasser's Joe le pendulum

Figure 2.7: Joe le pendulum[10]

This robot has been developed by team from the Industrial Electronics Laboratory at the Swiss Federal Institute of Technology, Lausanne, Switzerland. First idea is from year 1996, which is actually 4 years before the first release of Segway™, and the first working model was made in 1999. It is very well done from the first step. Authors made a dynamic model and verified it with simulations before creating the working model.

Latest version has all controlls on board, except for the remote joystick. It is using only odometry and gyroscope. The robot weights about 12kg and can go up to 1,5m/s.

It should be noted that in their dynamic model for purpose of linearization they assume the deviations of body tilt angle to be small, also they assume Body inertia around axe perpendicular to the groud is constant.

2.7 Dirk Uffmann's Artist

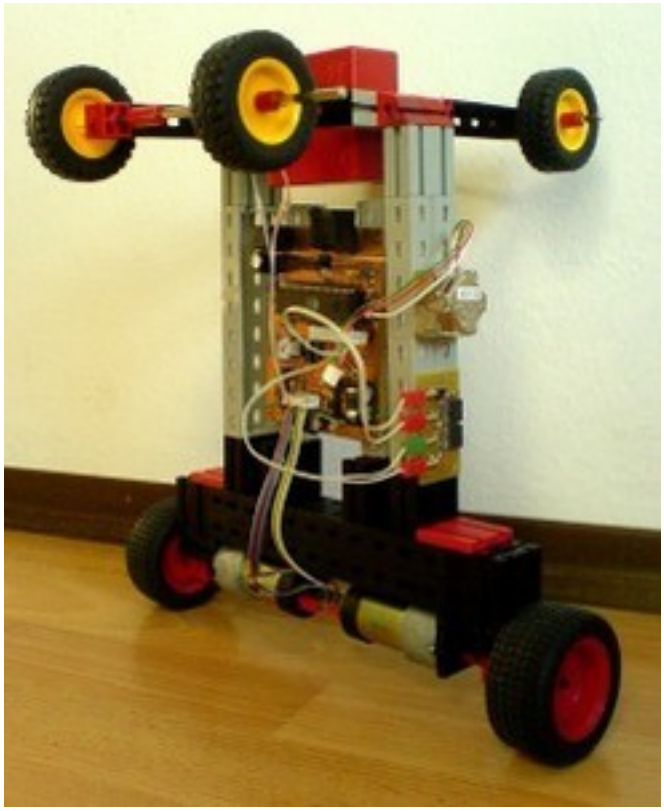


Figure 2.8: Dirk Uffmann's Artist[7]

This platform intrigued us for its mechanical construction, using fischertechnik™[13], which is an exceptional tool for such experimental projects[4].

Control is done by Atmel AVR ATMEGA16 and L293D motor controller. Sensoric array is measuring tilt angle using Freescale™accelerometers MMA2260D and MMA1260D. To measure angular velocity CRS03-02 (by Silicon Sensing / BAE Systems[14]) is used. Remote control has been implemented via infrared remote used for TV. Balance regulation in last version is done by a `PID` controller. In a previous step a PD-controller was used. According to the author, `PID` is more stable than PD. For `PID` control the tilt angle have to be integrated for use in the con-

[4]Author had a chance to work with it before.

trol equation. There is no odometry implemented, however it is planned in further development[5].

[5]Last update on the project web was on March 23rd, 2008; Original version was published on March 5, 2007

2.8 Matt Cross's Fire Marshal Bill

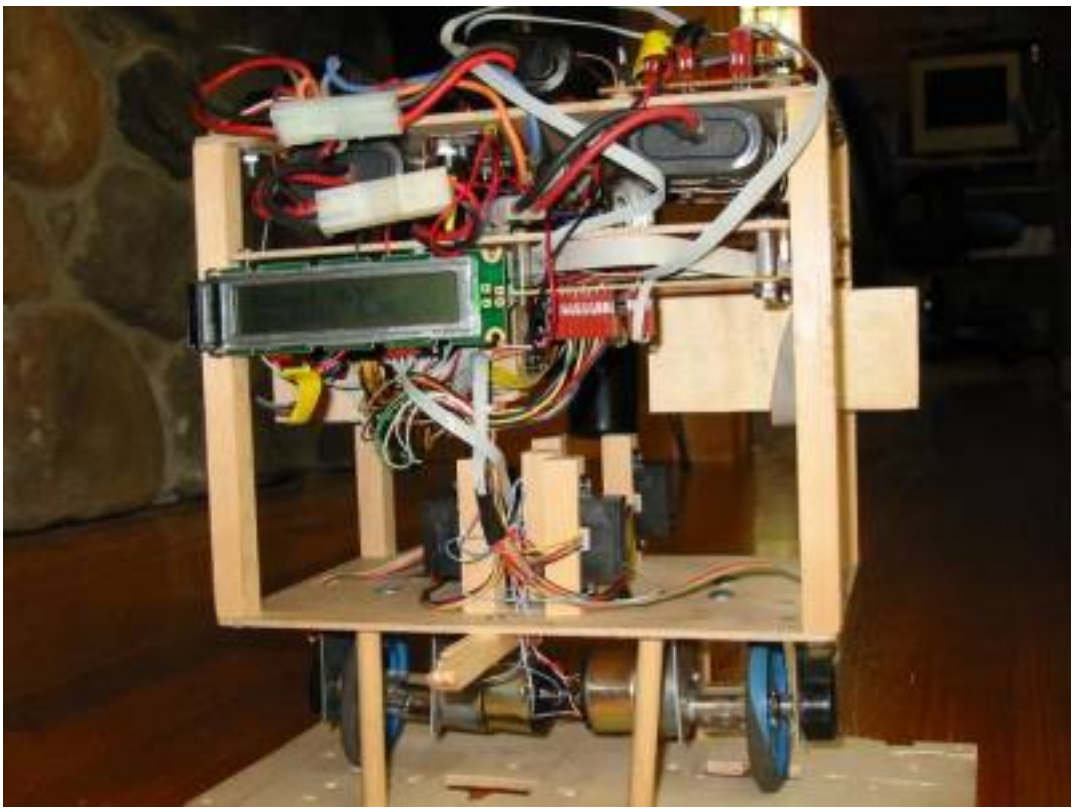


Figure 2.9: Matt Cross's Fire Marshal Bill[12]

This robot is a little unique, as it was built for Fire-fighting competition. The idea is based on David Andersons nBot and Larry Barellos GyroBot, but it has some new ideas on its own. The goal in the competition is to snuff out a candle flame by blowing it with fan. This actually brings an interesting problem into stability of the two-wheeled robot, as by turning the fan on and off, the inertia and dynamics of the robot changes slightly (draft done by fan). But if we would attach for example some manipulator arm, and started to actuate it, it could have significant influence on the control system.

The robot itself is kept functional simple in construction: wooden based body, two geared DC motors with optical encoders, ADXL202 accelerometer, IR sensors, distance sensors, and piezoelectric gyroscope. What makes this robot interesting is

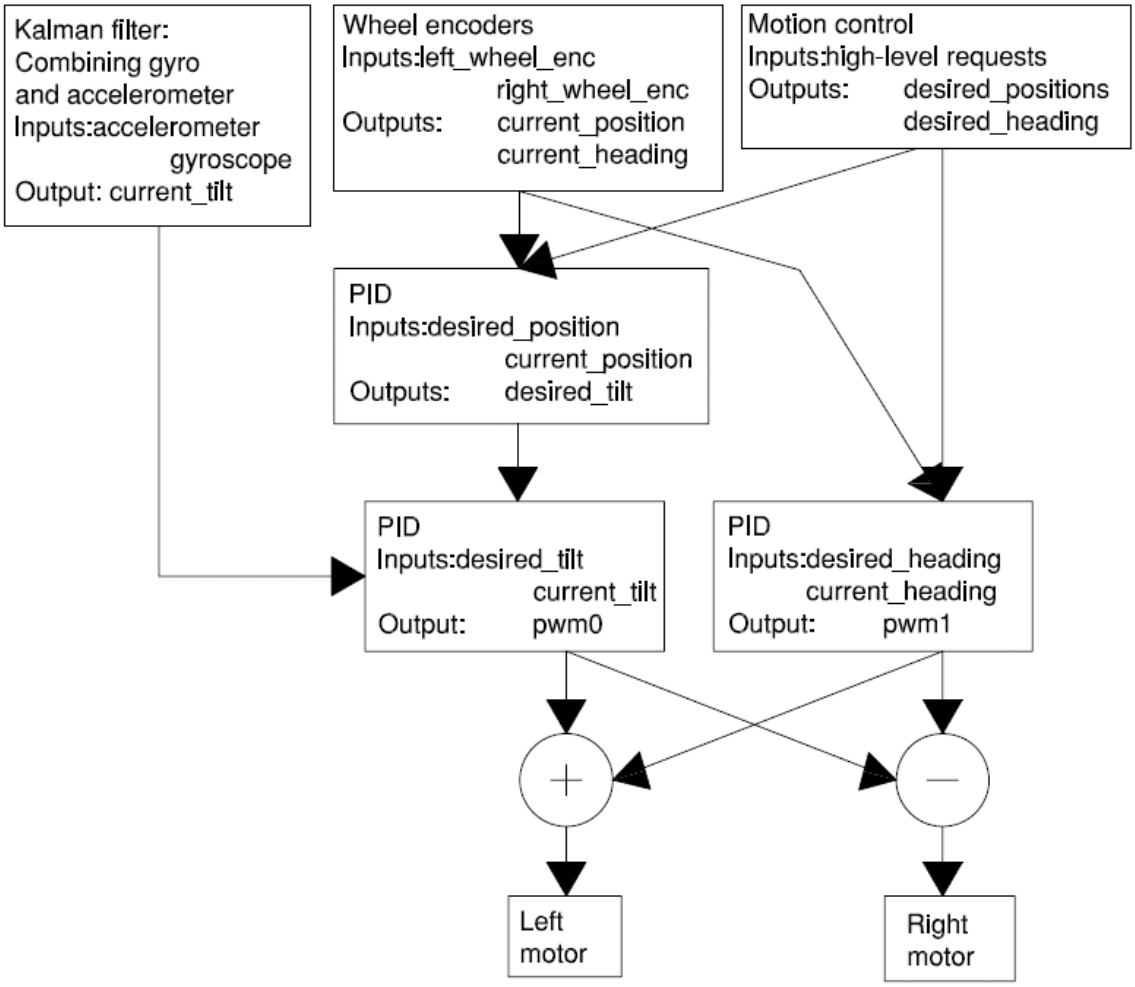


Figure 2.10: Fire Marshal Bill control scheme[12]

use of Kalman filter[6], which combines and refines data readings from gyroscope and accelerometer. Complete control scheme is shown on Fig: 2.10. This scheme also shows an interesting way to control the robot position, when first `PID` determines desired tilt from current and desired position, second `PID` takes current tilt from Kalman filter and desired tilt and sets PWM signal for motors. The PWM is then modified by third `PID` which is for heading control. As this robot is quite simple in the mechanical construction, it is very well compiled in the control section. All the source codes are available for download from the project homepage.

[6]The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown.[23]

2.9 Conclusion

There are many more similar projects in the field of small inverted pendulum based robots, and they are still being developed and build, as it is an interesting school project topic offering hands-on experience with various problems. To end-up with a clear and understandable conclusion, we have to look on each part of problem separately: construction, electronics + sensoric and control algorithms.

On the first glance it doesn't seem that the frame of the robot is something we should spend to much time on. However after having closer look we will realize that it has an important effect on dynamics of the body: position of `CG` and inertias, which will determine the stability and response of the whole robot. The higher is `CG`, the greater is inertia with respect to wheel axis, which in the end makes it more stable. On the other side, it requires more torque to get in upright position from position when lying on the ground.

Another chapter are sensors: Odometry - there is nothing much to deal with here, most motors can be purchased with some kind of encoder. Tilt angle and rate - can be determined by gyroscope and accelerometers, which is most common combination. Use of Kalmann filter provides accurate and for control purposes usable results. So only problem that needs to be resolved is the speed / sample rate of these two sensors, noise resistance and in long term the readings drift.

Because neither the accelerometers nor the rate gyro alone are capable of providing us with tilt angle and rate with sufficient stability and accuracy, we need to use them together. The accelerometers give a long term stable information of the direction to the earth middle, nevertheless they give a noisy signal and are not accurate enough for this kind of control. The rate gyro is very accurate but has a drift of its bias. To obtain the tilt angle, the angular change rate of the gyro must be integrated. When integrating the signal with a bias error, this error adds up considerably and leads to an instable behavior of the robot.

As for the microcontroller: basically any DSP can be used, it has to be able to communicate with all sensors (analog or digital port), control of motors (ideally capable of PWM) and perhaps some communication with operator. In following, we will try to resolve the dynamics and control design of small sized two wheeled inverted pendulum robot.

3 Mobile robot Pierot

3.1 Pierot design

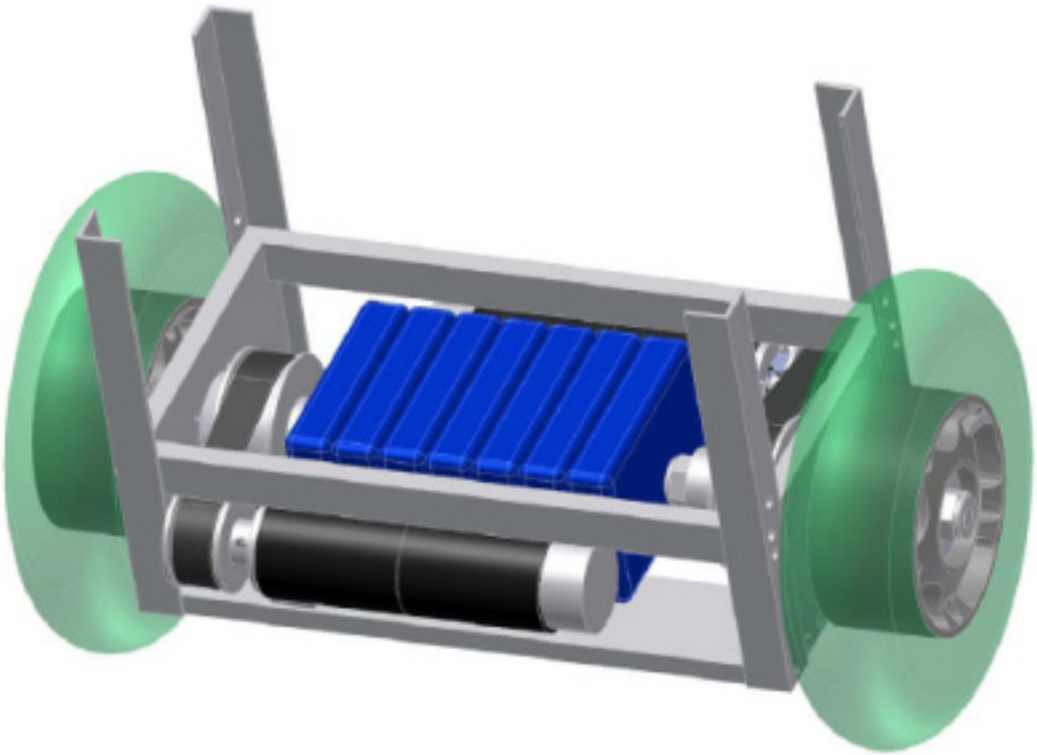


Figure 3.1: Experimental platform Pierot

For the purposes of testing and development of the control system, a real platform has been designed by my supervisor, it is a mobile inverted pendulum. The robot is called "Pierot" and it is shown on figure 3.1.

- Dimensions: about 200x100x70 mm
- Weight of body with batteries: approx. 1.5 kg.
- Wheels: Inline scooter wheels ø102 mm.
- Motors: 2x Maxon EC-max 22, with gearbox GP 22 C and encoder

3.2 Pierot dynamic model

3.2.1 Pierot dynamics

Dynamic model is important for understanding behavior and controller development for a robot. For compilation of a dynamic model, Lagrange's method was used. Problem has been simplified to a 2D 2-DOF problem, as shown in figure 3.2, used variables are stated in table 3.1. Other model have been created in MATLAB - Simulink using SimMechanics toolbox, this will be further described in section 3.2.3. Simplification into 2D problem is done by "removing" the second wheel, in consequence we loose ability to change direction, we treat model as it has one wheel, but the mass and inertia are considered double as well as the maximal motor momentum, and damping constants.

Symbol	Description
φ_B	body tilt angle
m_B	body mass
H_B	distance between wheel axle and body CG
φ_W	wheel angle
m_W	wheel mass
R_W	wheel radius
I_W	wheel inertia
M_M	motor momentum
b_{roll}	damping in contact between wheel and ground
$b_{gearing}$	damping in gearing
q	generalized coordinates
g	gravitational acceleration
x	wheel base position

Table 3.1: 2D Model variables

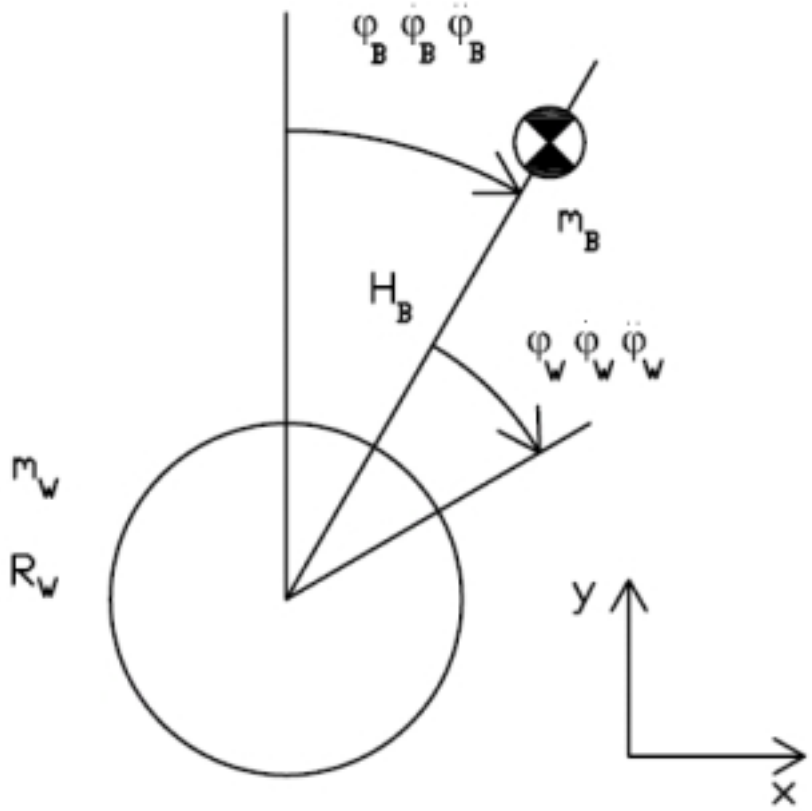


Figure 3.2: 2D model

3.2.2 Lagrangian dynamics equations in 2D

Kinetic Energy (generalized coordinates x, φ_B), and generalized coordinate derivate:

$$E_k = \frac{1}{2} m_W \cdot \dot{x}^2 + \frac{1}{2} I_W \cdot \frac{\dot{x}^2}{R_W^2} + \frac{1}{2} m_B \cdot \dot{x}^2 + \frac{1}{2} \left(I_B + m_B \cdot H_B^2 \right) \cdot \dot{\varphi}_B^2 \tag{3.1}$$

$$E_k = \frac{1}{2} \left(m_W + m_B + \frac{I_W}{R_W^2} \right) \dot{x}^2 + \frac{1}{2} \left(I_B + m_B \cdot H_B^2 \right) \cdot \dot{\varphi}_B^2 \tag{3.2}$$

$$\frac{\partial E_k}{\partial \dot{x}} = \left(m_W + m_B + \frac{I_W}{R_W^2} \right) \dot{x} \tag{3.3}$$

$$\frac{d}{dt} \left(\frac{\partial E_k}{\partial \dot{x}} \right) = \left(m_W + m_B + \frac{I_W}{R_W^2} \right) \ddot{x} \tag{3.4}$$

$$\frac{\partial E_k}{\partial x} = 0 \tag{3.5}$$

$$\frac{\partial E_k}{\partial \dot{\varphi}_B} = \left(I_B + m_B \cdot H_B^2\right) \cdot \dot{\varphi}_B \tag{3.6}$$

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \dot{\varphi}_B}\right) = \left(I_B + m_B \cdot H_B^2\right) \ddot{\varphi}_B \tag{3.7}$$

$$\frac{\partial E_k}{\partial \varphi_B} = 0 \tag{3.8}$$

Potential Energy and generalized coordinate derivate:

$$E_p = g \cdot m_B \cdot H_B \cdot \cos\varphi_B \tag{3.9}$$

$$\frac{\partial E_p}{\partial x} = 0 \tag{3.10}$$

$$\frac{\partial E_p}{\partial \varphi_B} = -g \cdot m_B \cdot H_B \cdot \sin\varphi_B \tag{3.11}$$

Loss force and generalized coordinate derivate:

$$\mathrm{Re} = \frac{1}{2} b_{roll} \cdot \dot{x}^2 + \frac{1}{2} b_{gearing} \cdot \left(\frac{\dot{x}}{R_W} - \dot{\varphi}_B\right)^2 \tag{3.12}$$

$$\frac{\partial \mathrm{Re}}{\partial \dot{x}} = b_{roll} \cdot \dot{x} + b_{gearing} \cdot \left(\frac{\dot{x}}{R_W^2} - \frac{\dot{\varphi}_B}{R_W}\right) \tag{3.13}$$

$$\frac{\partial \mathrm{Re}}{\partial \dot{\varphi}_B} = b_{gearing} \cdot \left(\dot{\varphi}_B - \frac{\dot{x}}{R_W}\right) \tag{3.14}$$

Work and generalized coordinate derivate:

$$W = M_M \left(\frac{x}{R_W} + \varphi\right) \tag{3.15}$$

$$\frac{\partial W}{\partial x} = M_M \cdot \frac{1}{R_W} \tag{3.16}$$

$$\frac{\partial W}{\partial \varphi} = M_M \tag{3.17}$$

Lagrange's dynamic equations:

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \dot{q}}\right) - \frac{\partial E_k}{\partial q} + \frac{\partial E_p}{\partial q} + \frac{\partial \mathrm{Re}}{\partial \dot{q}} = \frac{\partial W}{\partial q} \tag{3.18}$$

Assembled equations:
$q = x, \dot{q} = \dot{x}$

$$\left(m_W + m_B + \frac{I_W}{R_W^2}\right)\ddot{x} - 0 + 0 + b_{roll}\cdot\dot{x} + b_{gearing}\cdot\left(\frac{\dot{x}}{R_W^2} - \frac{\dot{\varphi}_B}{R_W}\right) = M_M \cdot \frac{1}{R_W} \quad (3.19)$$

$$\left(m_W + m_B + \frac{I_W}{R_W^2}\right)\ddot{x} + \left(b_{roll} + \frac{b_{gearing}}{R_W^2}\right)\dot{x} - \frac{b_{gearing}}{R_W}\dot{\varphi}_B = M_M \cdot \frac{1}{R_W} \quad (3.20)$$

$q = \varphi_B, \dot{q} = \dot{\varphi}_B$

$$\left(I_B + m_B \cdot H_B^2\right)\ddot{\varphi}_B - 0 - g\cdot m_B \cdot H_B \cdot \sin\varphi_B + b_{gearing}\cdot\left(\dot{\varphi}_B - \frac{\dot{x}}{R_W}\right) = -M_M \quad (3.21)$$

$$\left(I_B + m_B \cdot H_B^2\right)\ddot{\varphi}_B + b_{gearing}\cdot\dot{\varphi}_B - \frac{b_{gearing}}{R_W}\dot{x} - g\cdot m_B \cdot H_B \cdot \sin\varphi_B = -M_M \quad (3.22)$$

Final arrangement:

$$\left(m_W + m_B + \frac{I_W}{R_W^2}\right)\ddot{x} + \left(b_{roll} + \frac{b_{gearing}}{R_W^2}\right)\dot{x} = M_M \cdot \frac{1}{R_W} + \frac{b_{gearing}}{R_W}\dot{\varphi}_B \quad (3.23)$$

$$\left(I_B + m_B \cdot H_B^2\right)\ddot{\varphi}_B + b_{gearing}\cdot\dot{\varphi}_B - g\cdot m_B \cdot H_B \cdot \sin\varphi_B = -M_M + \frac{b_{gearing}}{R_W}\dot{x} \quad (3.24)$$

Arrangement for Matlab:

$$\ddot{x} = \frac{1}{\left(m_W + m_B + \frac{I_W}{R_W^2}\right)R_W}M_M + \frac{b_{gearing}}{\left(m_W + m_B + \frac{I_W}{R_W^2}\right)R_W}\dot{\varphi}_B - \frac{\left(b_{roll} + \frac{b_{gearing}}{R_W^2}\right)}{\left(m_W + m_B + \frac{I_W}{R_W^2}\right)}\dot{x} \quad (3.25)$$

$$\ddot{\varphi}_B = -\frac{M_M}{(I_B + m_B \cdot H_B^2)} + \frac{b_{gearing}}{R_W\,(I_B + m_B \cdot H_B^2)}\dot{x} - \\ -\frac{b_{gearing}}{(I_B + m_B \cdot H_B^2)}\dot{\varphi}_B + \frac{g\cdot m_B \cdot H_B}{(I_B + m_B \cdot H_B^2)}\sin\varphi_B \quad (3.26)$$

Figure 3.3 shows Simulink model based on set of equations (3.25) and (3.26), as it is prepared for linearization with linmod() command. Nonlinearity is caused by the $\sin\varphi_B$ in equation (3.26). All the constants are defined in m-file. Another model representation have been created with SimMechanics toolbox, it will be introduced in section 3.2.3. Model has been used for behavior testing and verification of the equations. For comparison of Lagrange based and SimMechanics model we have run the simulation with same initial conditions. Plotted results are shown on figure 3.4, both models behave very alike. Please note that for the testing, we have set

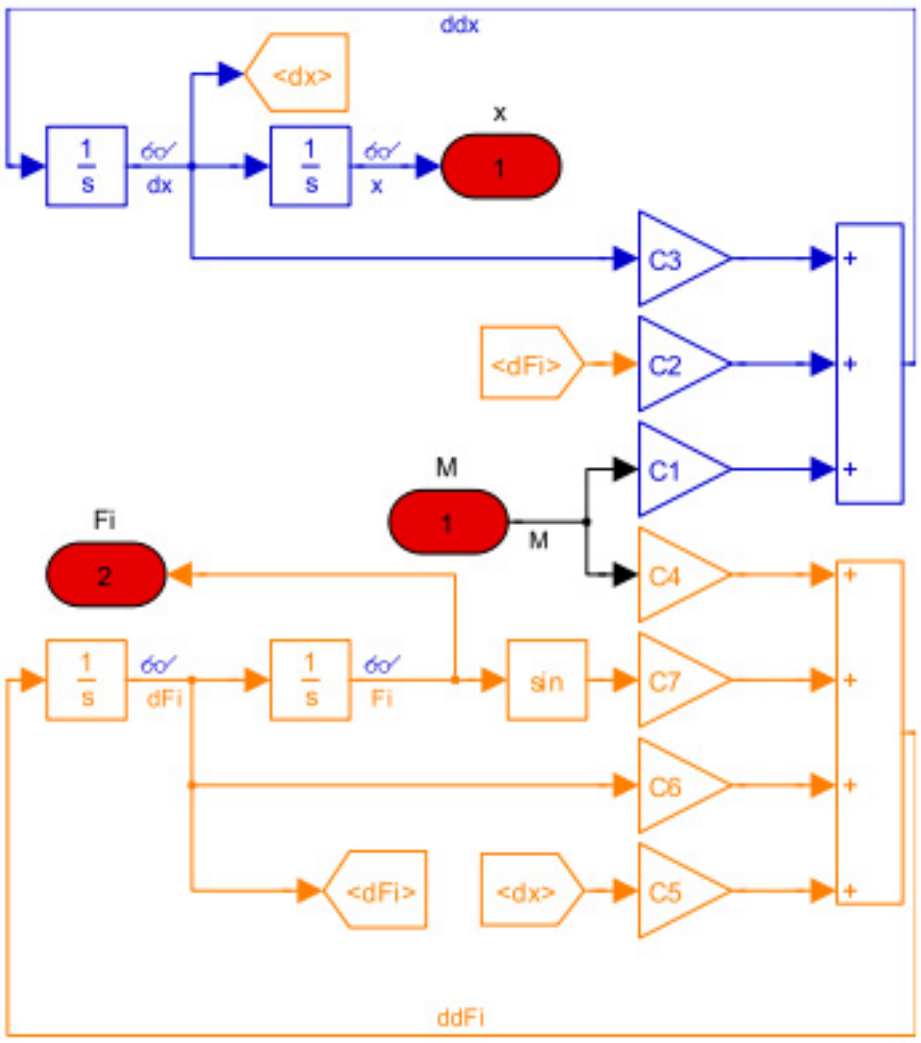


Figure 3.3: Simulink model based on Lagrange equations set for linmod()

$broll = 1$ to simulate stopped motor[1]. Because there are no constrains such as a floor in the simulation, the body angle φ stabilize at 3.14 (π), which corresponds with 180°. There are some other possible approaches to get the motion equations, one of them is Kane's method[2] which has been used by Yeonhoom Kim[21] and his

[1]This is not really accurate: If we would want to simulate non powered motor, we have to consider its inertia, acceleration etc., so this would lead to another problematic, which is not a subject of this work.

[2]Kane's method has been presented by Thomas Kane and David Levinson in [25].

colleagues. In their paper they are working on dynamic analysis of a nonholonomic[3] two-wheeled inverted pendulum robot. Model based on their equations is much closer to the real, as they are done in 3D - with all 3 DOFs.

[3]Nonholomic constrain is constrain which can't be expressed just by coordinates or time, in general it is described by differential equation, which can't be integrated to holonomic form.

3.2.3 MATLAB - SimMechanics Model

Wheel reduction

It is to complicated[4] to model wheel as a real wheel in a SimMechanics, and if we don't need to consider slipping, whole problem can be simplified by reducing wheel to a sliding box. Modeling the wheel with this method is fully equivalent to a non-slipping wheel.

$$E_k = \frac{1}{2} m_{red} \cdot \dot{x}^2 = \frac{1}{2} m_W \cdot \dot{x}^2 + \frac{1}{2} I_W \cdot \dot{\varphi}_W^2 = \frac{1}{2} \left[m_W + \frac{I_W}{R_W^2} \right] \dot{x}^2 \tag{3.27}$$

$$\begin{aligned} P &= F_{red} \cdot \dot{x} = M \cdot \dot{\varphi}_W - b_{roll} \cdot \dot{x}^2 - b_{gearing} \cdot \dot{\varphi}_W \left(\frac{\dot{x}}{R_W} - \dot{\varphi}_B \right) = \\ &= M \cdot \frac{\dot{x}}{R_W} - b_{roll} \cdot \dot{x}^2 - b_{gearing} \cdot \frac{\dot{x}}{R_W} \left(\frac{\dot{x}}{R_W} - \dot{\varphi}_B \right) = \\ &= M \cdot \frac{\dot{x}}{R_W} - b_{roll} \cdot \dot{x}^2 - b_{gearing} \frac{\dot{x}^2}{R_W^2} + b_{gearing} \frac{\dot{\varphi}_B}{R_W} \dot{x} = \\ &= \left(\frac{M}{R_W} - b_{roll} \cdot \dot{x} - b_{gearing} \frac{\dot{x}}{R_W^2} + b_{gearing} \frac{\dot{\varphi}_B}{R_W} \right) \dot{x} \end{aligned} \tag{3.28}$$

From equations (3.27) and (3.28), we can express m_{red} and F_{red}, and them put them into our model.

$$m_{red} = m_W + \frac{I_W}{R_W^2} \tag{3.29}$$

$$F_{red} = \frac{M}{R_W} - b_{roll} \cdot \dot{x} - b_{gearing} \frac{\dot{x}}{R_W^2} + b_{gearing} \frac{\dot{\varphi}_B}{R_W} \tag{3.30}$$

SimMechanics Model

For the purposes of simulation and control design, model in MATLAB - SimMechanics has been created. Model shown on figure 3.5 is as it is prepared for the linmod() MATLAB command, which obtains linear state-space model from system. Before using the linmod, we also need to use the trim() command, which finds steady state parameters for a system described in Simulink[22]. To make trim work properly, we need to have the model in certain form: There has to be input, output and integrator

[4] By modeling the wheel as an rotational and prismatic constrain, we would add an extra `DOF`, which would make the model inaccurate, unless we are considering slipping between the ground and wheels.

blocks which defines inputs, outputs and states of the system, also initial conditions should be set. After linearization and creating the linear state space model, we need to ensure, that the linearized model actually behaves like the non-linearized one, and also to check under which conditions[5] it fits enough to be usable. So it can be determined if one model is good for whole working range, or we need to linearize in more than one set of conditions.

[5]This mostly considers the tilt angle, which is the biggest source of non-linearities.

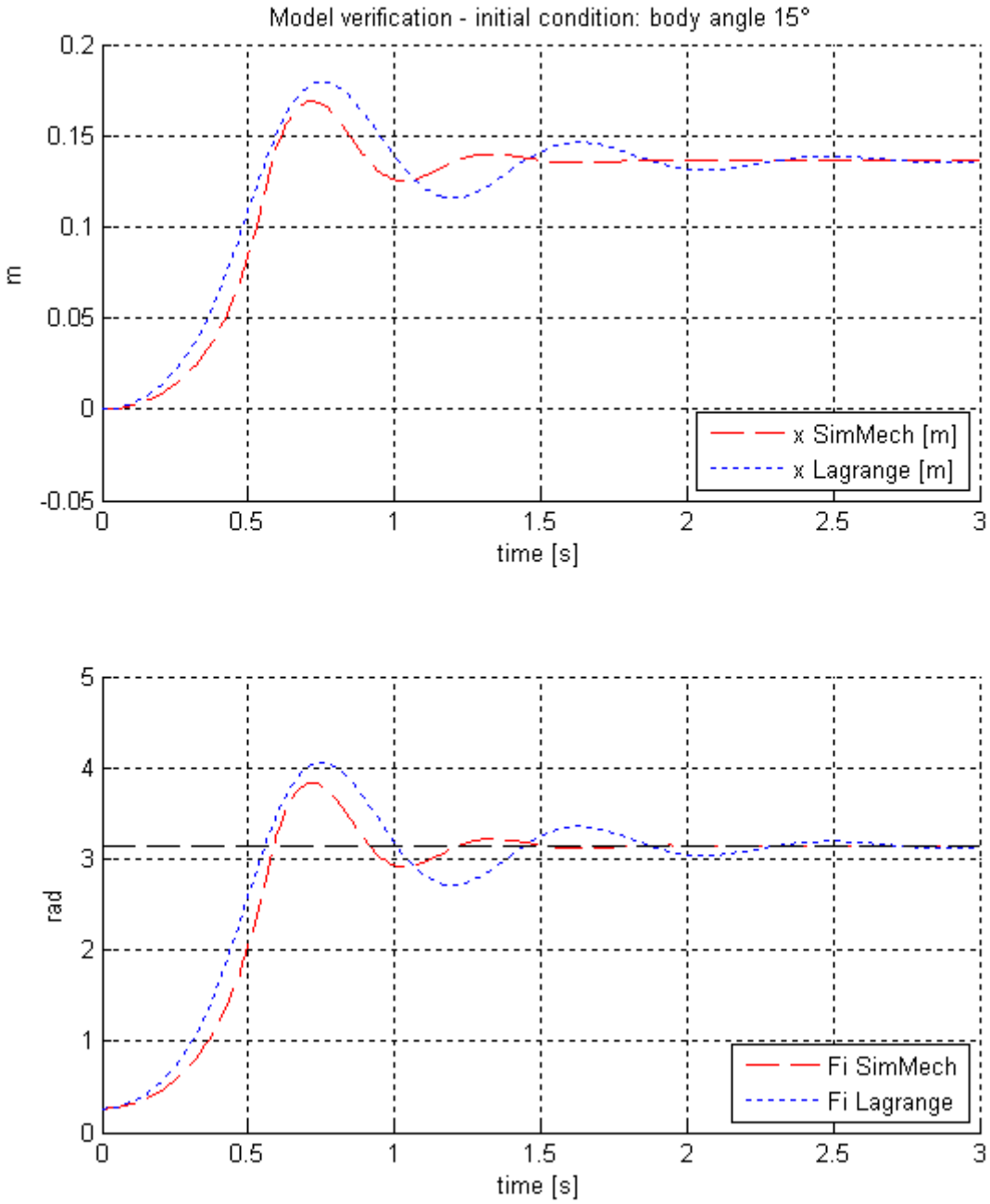


Figure 3.4: Model comparison: Lagrange and SimMechanics model response on initial deflection

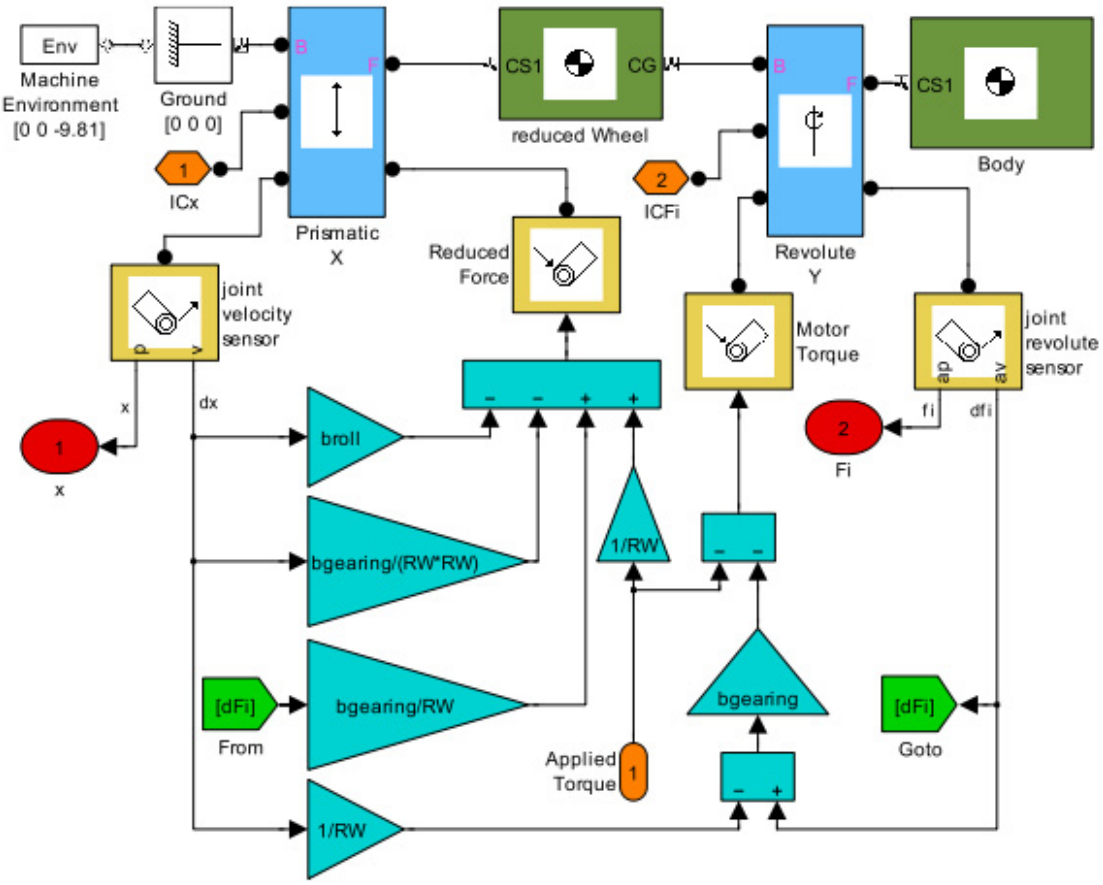


Figure 3.5: Plane model of wheeled inverted pendulum with 2 `DOF`, prepared for linmod()

4 Control design

Main goal of the control system is to keep the robot body under desired angle (mostly in upright position,) any other control (heading, position) is useless if the robot falls down. This problem is very similar to the inverted pendulum control problem, but not the same. What is known as inverted pendulum is basically a stick on a cart, where there is no actuator between the stick and the cart, and the cart is stable. In the case of two-wheeled balancing robot, there is a motor on the body (stick) which actuates both the wheels and the body at the same time. When we want robot (initially in upright position) to move in one way, we have to tilt him first by going the other way, the tilt angle will depend on desired speed.

4.1 Control scheme

To define motion and position of a two-wheeled inverted pendulum, we need to determine following variables - states:

- Tilt angle
- Tilt rate (angle first derivate)
- Platform position
- Platform velocity (position first derivate)

These all are directly or indirectly measurable states, so the system should be controllable from this point of view. Simple general control scheme could be as shown in figure 4.1, where regulator could represent ie.: `PID` controller, Fuzzy logic controller...

As a system input we use a motor momentum, which is not exactly accurate, since we are using DC motors, the input of the system should be the current. Because the time constant of DC motors even with current control loop is small compared to the time constant of the inverted pendulum system, we can neglect motors dynamics, therefore we can use the motor torque as input.

Inverted pendulum control design is a complex control problem as it is unstable and nonlinear system. There are many ways to design a control, but all the methods

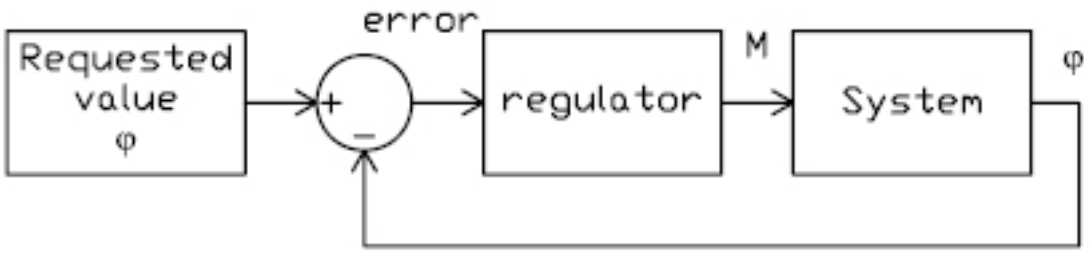


Figure 4.1: Simple control scheme

are limited in use on linear[1] systems, so we have to linearize our system. That can be done analytical by using Taylor series on the Lagrange equations, or we can use linmod() tool in MATLAB. There are options with that as well, we can run the linmod() command on the Lagrange equations representation shown in figure 3.3, or we can use a SimMechanics toolbox and then use linmod() on this model. In further work, we will use both models to compare their results. SimMechanics model is introduced in section 3.2.3.

Before we start designing any controller, according to Březina[24] we should study `Dynamic system behavior`. Usually this would mean testing response of the system for the:

- Unit step (in time realm)[2]
- Unit impulse (in time realm)[3]
- Harmonic signal (in frequency realm)[4]

To show the results of linearization, and evaluation we have compared step response of nonlinear and linearized step response. These are shown in figures 4.2 and 4.3. We can se from the figures, that the position x is affected far less then the angle[5]

[1] *Linear* system mathematical representation is system of linear differential or difference equations. *Nonlinear* system mathematical representation is system of nonlinear differential or difference equations.

[2] in MATLAB: step(sys)

[3] in MATLAB: impulse(sys)

[4] in MATLAB: bode(sys) - amplitude and phase characteristics, nyquist(sys) - Nyquist diagram: assessment of the stability

[5] Please note that the units of the second output (φ) on4.3 are radians.

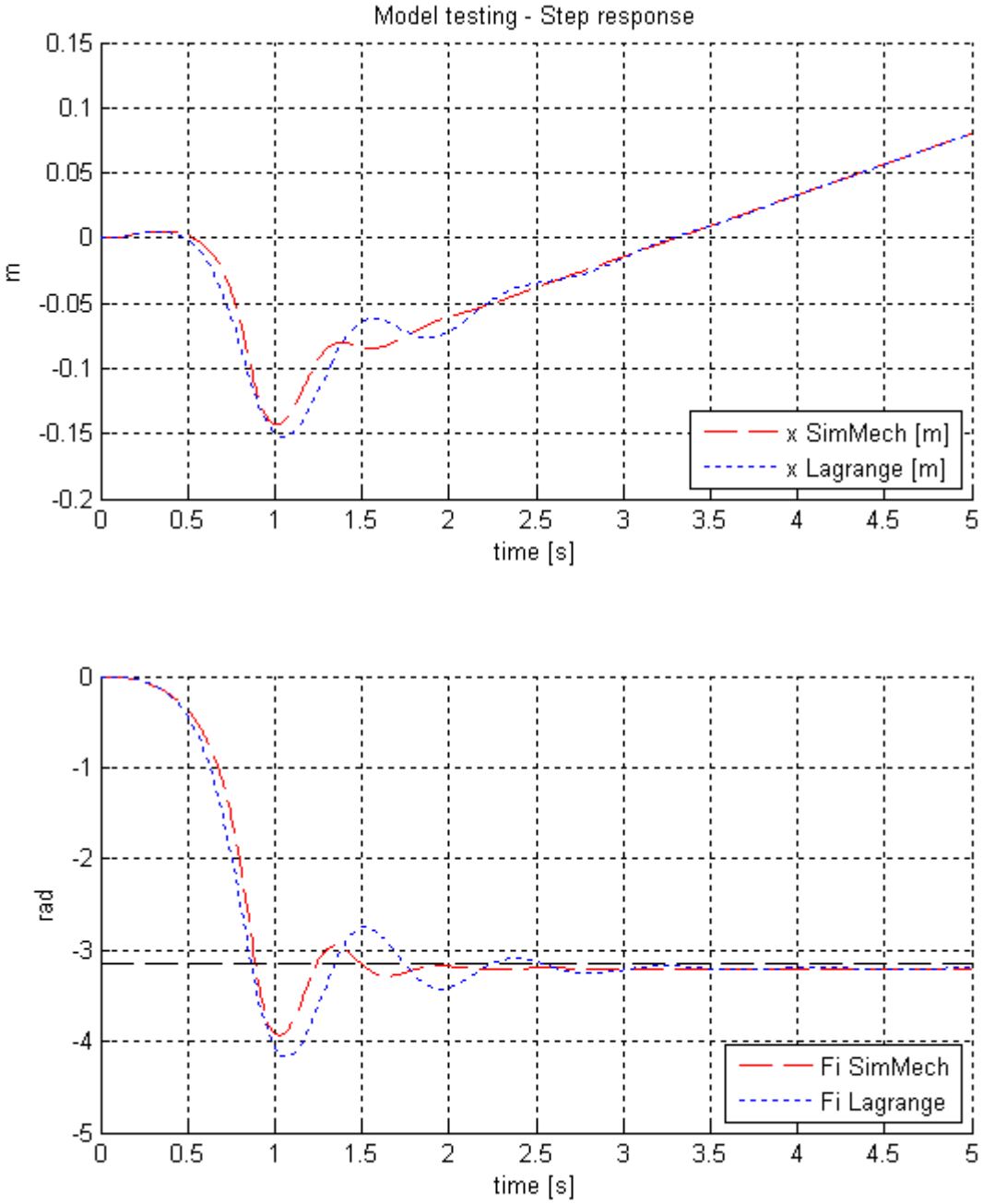


Figure 4.2: Unit step Response - $M_M = 1Nm$

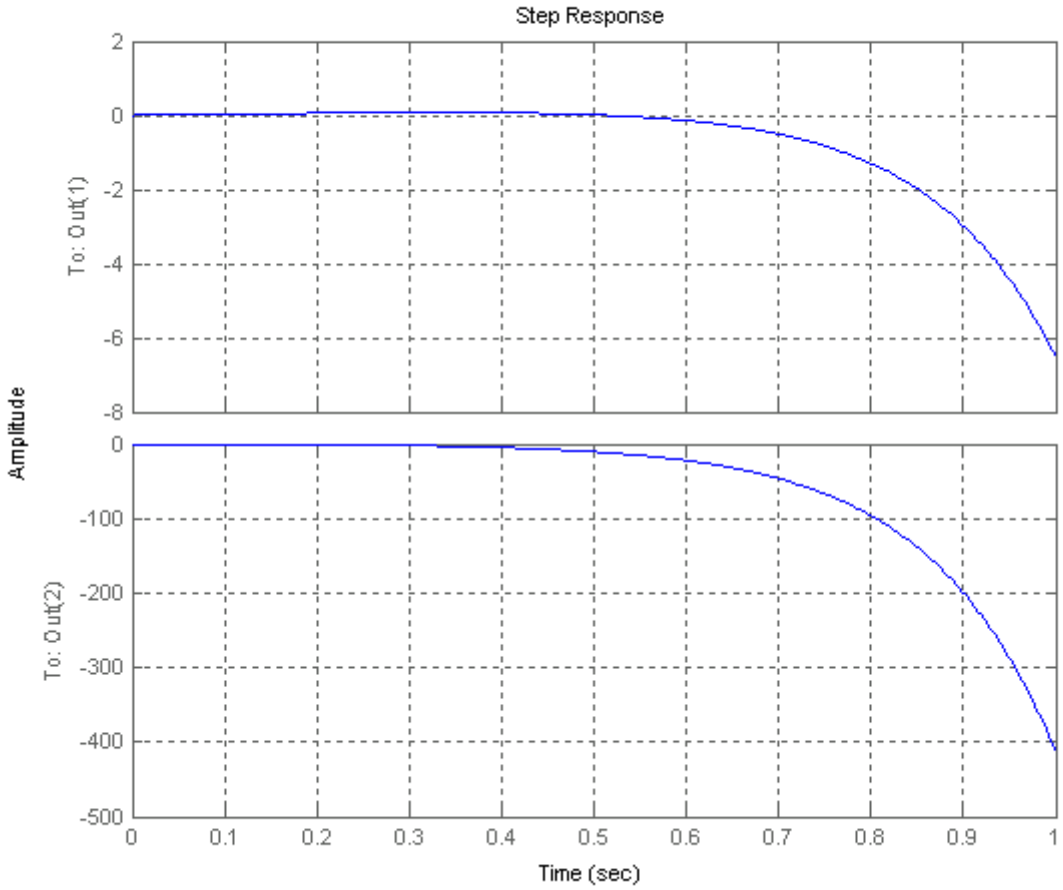


Figure 4.3: Unit step Response of the linearized model - $M_M = 1Nm$

4.2 Linearization of the Lagrange model

As we have stated before in section 3.2.2, nonlinearity is caused by the $\sin\varphi_B$ in equation (3.26), if we assume that the operating range would be in vicinity of zero tilt angle $\varphi_B = \pm 10°$, we can approximate $\sin\varphi_B \cong \varphi_B$. Then we can use state-space

representation presented in (4.1), (4.2), (4.3), and (4.4):

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\left(b_{roll}+\frac{b_{gearing}}{R_W^2}\right)}{\left(m_W+m_B+\frac{I_W}{R_W^2}\right)} & 0 & \frac{b_{gearing}}{\left(m_W+m_B+\frac{I_W}{R_W^2}\right)R_W} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{b_{gearing}}{R_W\left(I_B+m_B\cdot H_B^2\right)} & \frac{g\cdot m_B\cdot H_B}{\left(I_B+m_B\cdot H_B^2\right)} & -\frac{b_{gearing}}{\left(I_B+m_B\cdot H_B^2\right)} \end{bmatrix} \tag{4.1}$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{\left(m_W+m_B+\frac{I_W}{R_W^2}\right)R_W} \\ 0 \\ -\frac{1}{\left(I_B+m_B\cdot H_B^2\right)} \end{bmatrix} \tag{4.2}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.3}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.4}$$

4.3 Linearization in MATLAB

As we have brought up before, we can use MATLAB `linmod()` command to obtain a linearized model from a system of ordinary differential equations `ODE`s. To use it we need to have the system of `ODE`s described in the Simulink block diagram. In the diagram, inputs are represented by Input blocks, outputs are represented by Output blocks and states are represented by Integrators. `Linmod()` command returns matrixes A, B, C, D; where A is a State matrix, B is Input (excitation) matrix, C is Output matrix, and D is matrix of Direct transfer. To find out the sequence of states in matrixes, we need to execute following: `[sizes,x0,xstring] = Pierot_Simulink_LagrangeBased_R02`, where `xstring` will represent names of each state, as they are named in model:

```
xstring =
    'Pierot_Simulink_LagrangeBased_R02/Ix'
    'Pierot_Simulink_LagrangeBased_R02/IFi'
    'Pierot_Simulink_LagrangeBased_R02/Idx'
    'Pierot_Simulink_LagrangeBased_R02/IdFi'
```

This is important, because we could expect States to be in different order (x, $\dot{x}$, φ, and $\dot{\varphi}$), which could lead to some troubles in further work with the model. To obtain the sequence of States from model created in SimMechanics, we have to use a different command:

```
[vector_mgr, mech_states] = mech_get_states(stateVector(end,:),
 'SM_Pierot_v8_linearizationReady/Machine Environment [0 0 -9.81]')
```

Where we get the States from structure `vector_mgr.StateNames`

```
vector_mgr.StateNames =
    'SM_Pierot_v8_linearizationReady/RevoluteY:R1:Position'
    'SM_Pierot_v8_linearizationReady/PrismaticX:P1:Position'
    'SM_Pierot_v8_linearizationReady/RevoluteY:R1:Velocity'
    'SM_Pierot_v8_linearizationReady/PrismaticX:P1:Velocity'
```

We can see, that this command gives us States in different sequence in comparison with the other model, but the regularity is in sequence of order of derivate, first always goes the Position, and then its derivate: Velocity.

4.4 Controller Design

In following section, we will present some options for two-wheeled inverted pendulum control. Main goal of this section will be to design controller, which will keep the robot in upright position. We will describe several types of controllers and methods to design them.

4.4.1 PID Tuning

`PID`[6] controller is very common type of control loop feedback mechanism. It is designed to correct an error between measured and requested state of a system by setting an action variable. Basic `PID` controller scheme is on figure: 4.4. There are several methods used for `PID` controller tuning:

- Manual tuning: No math required. Online method. Requires experienced personnel.
- Zieger - Nichols oscillation method: Proven method. Online method. Only valid for open loop stable plants.
- Zieger - Nichols reaction curve method: Proven method. Online method.
- Cohen - Coon reaction curve method: Good process models. Some math. Offline method. Only good for first-order processes.
- Software tools: Consistent tuning. Online or offline method. May include valve and sensor analysis. Allow simulation before downloading. Some cost and training involved.

We will start with manual tuning, which is based on some regularities, these are described in table 4.1[7]. For `PID` tuning, we have used a Lagrange model shown in figure 4.5 as a subsystem for our controller scheme. In this case, model have been controlled on zero $Fi(\varphi_B)$ angle with initial 15° deviation. We have started with manual tuning and we have reached satisfactory performance. To tune the `PID` gains further, we have used `Output Constraint` block in simulink, which allows us to set boundaries of its input signal and then run Optimization process on selected parameters: In our case, as input is angle $Fi(\varphi_B)$, and as tuned parameters are Kp, Ki, and Kd. We didnt run the optimization process on the SimMechanics model,

[6] Proportional Integral Derivative

[7] `NDT` - No definite trend / minor change.

Response	Rise Time	Overshoot	Settling Time	S-S Error
K_P	Decrease	Increase	NDT	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_D	NDT	Decrease	Decrease	NDT

Table 4.1: Effects of P I D gains

because the simulation of the SimMechanics model is slower, and it should lead to similar result. However we have used gains obtained by optimalization for both models to compare the behavior. These are shown in figure 4.6.

4.4.2 Linearized model PID control

We have used the linmod() command to obtain linear state space models from models described in 3.2.2, and 3.2.3 and we have tested them with our `PID` controller scheme described in 4.4.1 together with the State Space model which we extrapolated in 4.2. Results are shown in figure 4.7. All models, except the SimMechanics model are showing same behavior, we can see that the linear model is very accurate compared to the nonlinear Lagrange model, which is shown in same graph. We did not include the linearized SimMechanics model, as it behaved strangely. Unfortunately, we did not find the reason of the differences between SimMechanics and Lagrange models.

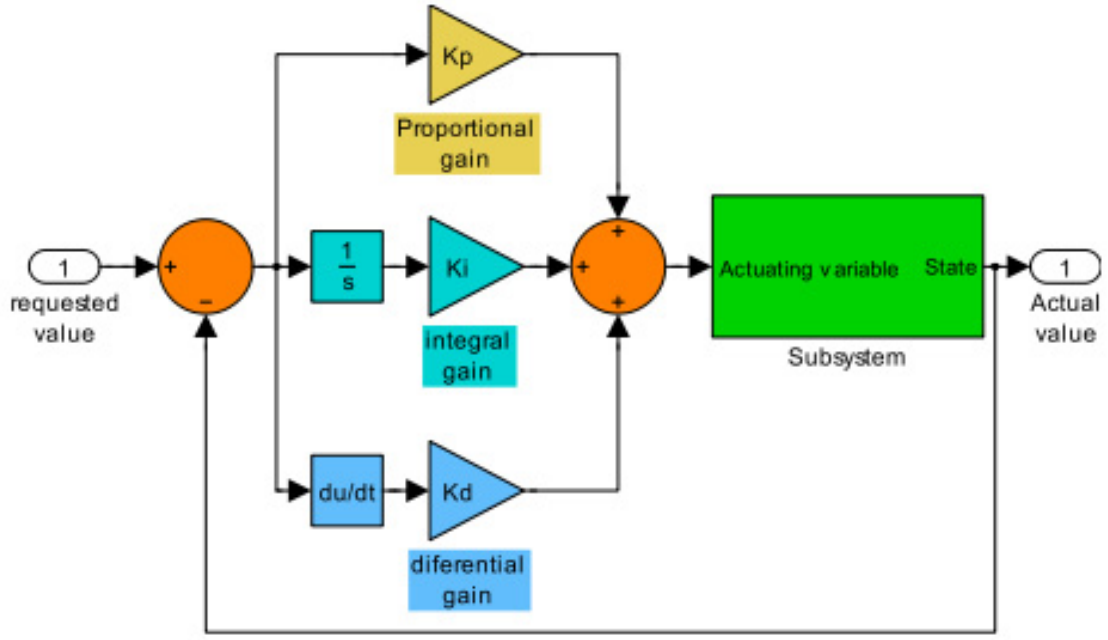


Figure 4.4: PID controller scheme

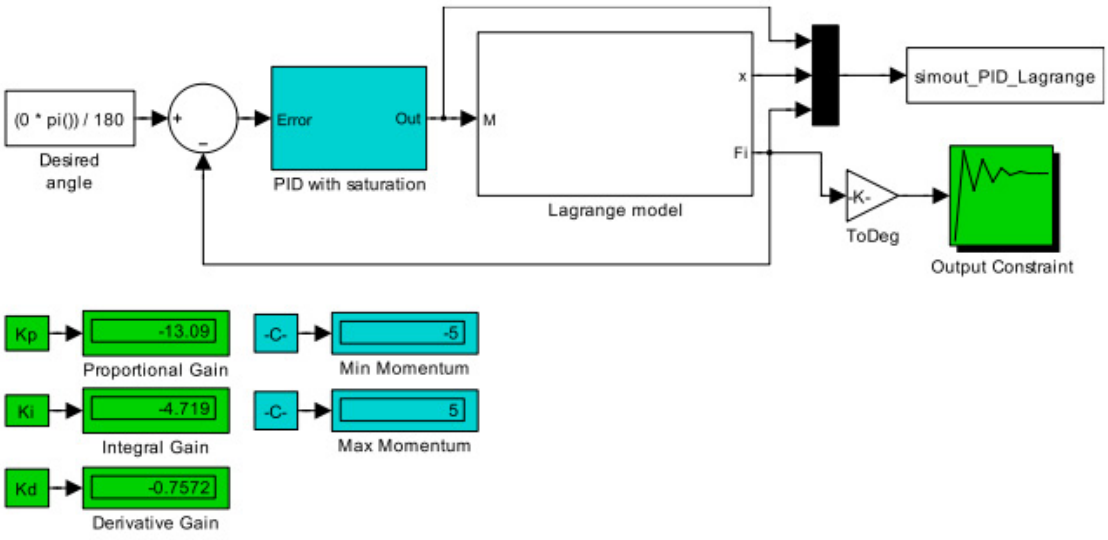


Figure 4.5: PID controller tuning with the Lagrange nonlinear

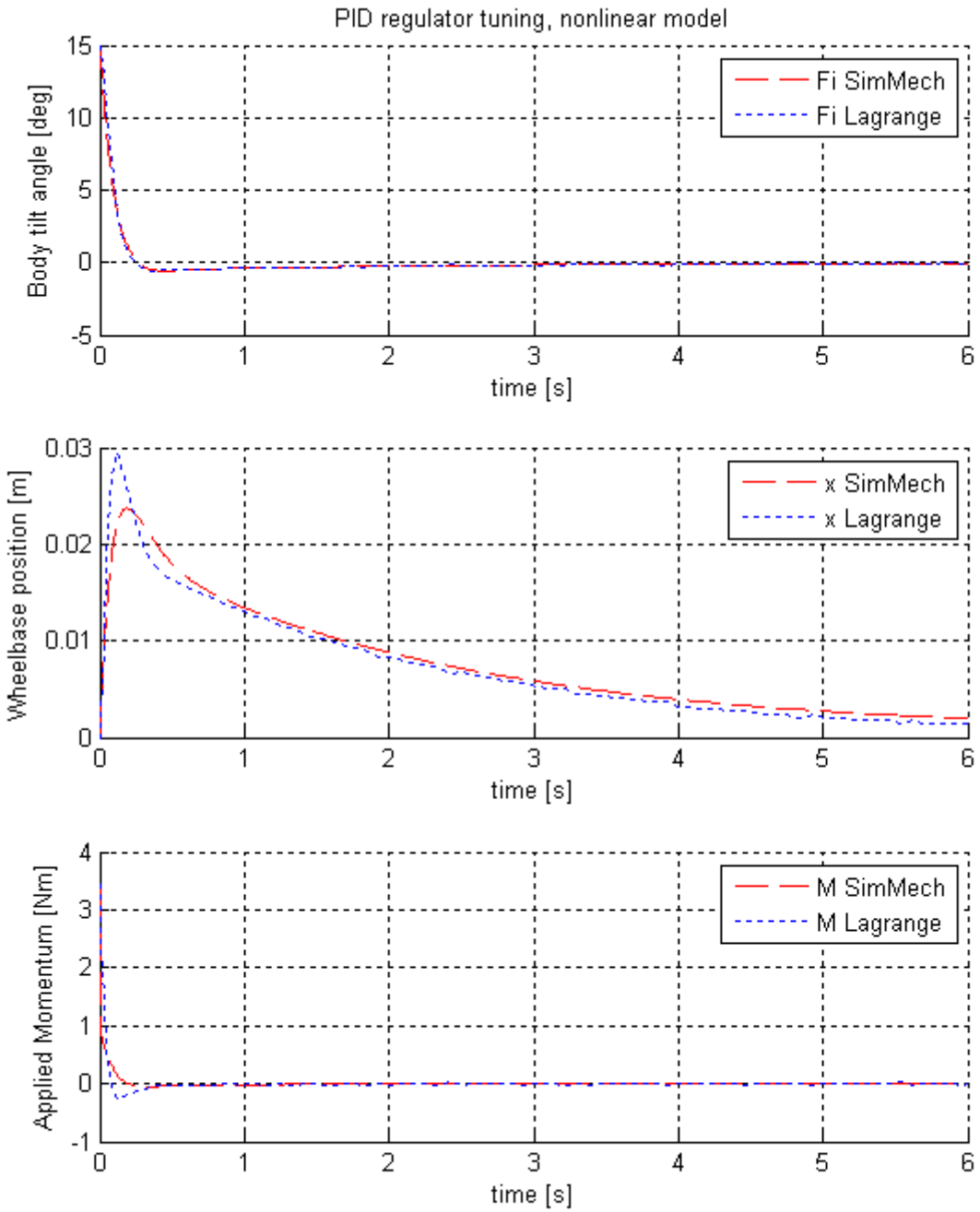


Figure 4.6: System with Tuned PID response

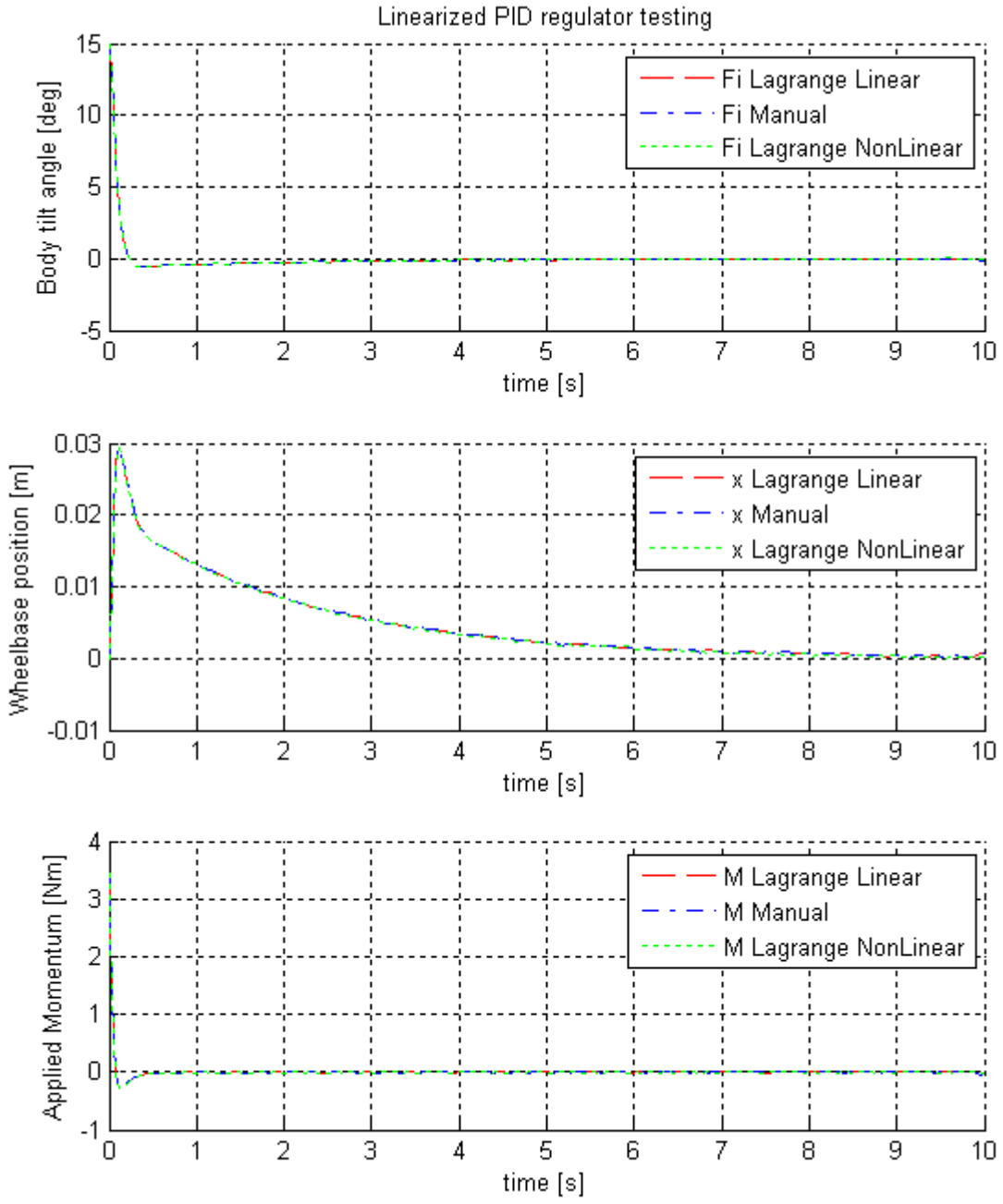


Figure 4.7: Linearized model PID control

5 Conclusion

The goal of this work was to design a controller for mobile two-wheeled inverted pendulum. We started with literature research on this topic, and come across several solutions and interesting ideas. We have compiled dynamics equations of simplified model and verified them in MATLAB-Simulink, also we have used SimMechanics toolbox to create corresponding model. Than we have designed simple `PID` controller to control the body tilt angle and used some tools to tune the `PID` gains. We have created a linear model from compiled equations and also in MATLAB with linmod() command, and used them in the `PID` control scheme.

We did not experienced many problems or complications during the work, development of the dynamics model is pretty straight. When we put the model based on Lagrange method into Matlab simulink, it is simple and well working, only special attention is needed for the linearization part, when we have to check on the order of the states in the returned state space model. As for SimMechanics toolbox, there we can expect a bit more compliactions, as we have to use the wheel reduction, which should lead to same results, but for some uncertain reasons we have experienced several divergences in the system behaviour. The conclusion of this is that the Lagrangian approach is fully satisfactory for system, where we can estimate behaviour in advance. SimMechanics on the other hand helps us with the visualization toll, that shows how the system behave in time.

As a part of this work, we have created a summary of different types of sensors, that can be used to measure states of the mobile two-wheeled pendulum. They are listed in the attachments.

6 References

[1] iBOT, From Wikipedia, the free encyclopedia:, http://en.wikipedia.org/wiki/IBOT, 22.03.2008

[2] iBOT - Dean L. Kamen's wheelchair:, http://www.ibotnow.com/, 22.03.2008

[3] Segway^{TM}- Dean L. Kamen's scooter:, http://www.segway.com/, 22.03.2008

[4] Trevor Blackwell's 1. scooter:, http://www.tlb.org/scooter.html, 22.03.2008

[5] Trevor Blackwell's 2. scooter:, http://www.tlb.org/scooter2.html, 22.03.2008

[6] David P.Anderson's nBot:, http://www.geology.smu.edu/~dpa-www/robo/nbot/, 22.03.2008

[7] Dirk Uffmann's Fischertechnik balancing platform:, http://home.arcor.de/uffmann/ARTIST.htm, 22.03.2008

[8] Dirk Uffmann's Fischertechnik balancing platform - Sensor Data Processing:, http://home.arcor.de/uffmann/ARTIST3.htm, 22.03.2008

[9] Dan Piponi's Equibot:, http://homepage.mac.com/sigfpe/Robotics/equibot.html, 22.03.2008

[10] Felix Grasser's Joe le pendulum:, http://leiwww.epfl.ch/joe/, 22.03.2008

[11] Larry Barello's gyrobot:, http://www.barello.net/Robots/gyrobot/index.htm, 22.03.2008

[12] Matt Cross's FireMarshalBill:, http://www.joustinghill.org/matt/robots/firemarshalbill/, 30.04.2008

[13] fischertechnik^{TM}- Building blocks for life:, http://www.fischertechnik.de/en/, 22.03.2008

[14] Silicon Sensing Systems Ltd:, http://www.siliconsensing.com/, 22.03.2008

[15] Analog Devices: MEMS Sensors:, http://www.analog.com/en/cat/0,2878,764,00.html, 12.04.2008

[16] MicroStrain: Gyro enhanced inclinometer:, http://www.microstrain.com/fas-g.aspx, 12.04.2008

[17] RoboteQ: Products for robotics:, http://www.roboteq.com/, 22.04.2008

[18] Rotomotion: Automotion for UAV and UAS:, http://www.rotomotion.com/, 23.04.2008

[19] ExtremeTech - The technology behind the Segway^{TM}:, http://www.extremetech.com/article2/0%2C1697%2C26752%2C00.asp, 24.04.2008

[20] Grasser F.,D'Arrigo A., Colombi S., Rufer A.: *JOE: A Mobile, Inverted Pendulum*, Laboratory of Industrial Electronics, Swiss Federal Institute of Technology Lausanne, EPFL, 2002

[21] Yeonhoom Kim, Soo Hyun Kim, Yoon Keun Kwak: *Dynamic Analysis of a Nonholonomic Two-Wheeled Inverted Pendulum Robot*, Department of Mechanical Engineering, KAIST, 373-1 Guseong-dong, Yuseong-gu, Daejeon, 305-701, South Korea, 2005

[22] Mathworks: *Mathworks - Matlab®help, The language of Technical Computing*, Mathworks, Inc, 1984-2007

[23] Greg Welch, Gary Bishop: *An Introduction to the Kalman Filter*, Department of Computer Science, University of North Carolina at Chapel Hill, July 2006

[24] doc. RNDr. Ing. Tomáš Březina, CSc.: *State feedback control design in Matlab through pole placement method*, Faculty of Mechanical Engineering Brno University of Technology, 2006

[25] Kane, Thomas R.; Levinson, David A.: *Dynamics: theory and applications*, McGraw-Hill, New York, 1985

[26] Švejda, Pavel; Grepl, Robert: *Diplomová práce: Modelování dynamiky a řízení dvoukolového nestabilního prostředku*, Fakulta strojnho inenrstv VUT, Brno, 2007

7 Used shortcuts

CG	Center of Gravity
DC	Direct Current
DOF	Degrees Of Freedom
DSP	Digital Signal Processor
HT	Human Transporter
MEMS	MicroElectroMechanical Systems
NDT	No definite trend / minor change
ODE	ordinary differential equation
PID	Proportional Integral Derivative
PWM	Pulse-Width Modulation

8 Annexes

8.1 Matlab files

- `01 Model testing.zip`:
  Model testing - Lagrange and SimMechanics

- `02 PID testing.zip`:
  PID controll of nonlinear model - Lagrange and SimMechanics

- `03 Linmod.zip` PID:
  controll of linearized model - Lagrange and SimMechanics

8.2 Other files

- `compare table - sensoric - Accelerometers.pdf`:
  Sensors overview - accelerometers

- `compare table - sensoric - Gyroscopes.pdf`:
  Sensors overview - gyroscopes

- `compare table - sensoric - Tilt angle.pdf`:
  Sensors overview - tilt angle