ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Autonomous Systems Lab
Prof. Dr. Roland Siegwart

Bachelor Thesis

# Modeling and Control of a Ballbot

Spring Term 2010

**Supervisors**
Stefan Leutenegger
Dr. Francis Colas

**Authors**
Péter Fankhauser
Corsin Gwerder

## Abstract

A ballbot is an extremely agile mobile system due to its natural instability. In order to perform complex motions at high speeds, a sophisticated controller is needed. This thesis first approximates the dynamics of a ballbot with three two-dimensional decoupled systems. It analyzes the dynamic internal forces occurring when operating a ballbot. Conversions for torques, moments of inertia and the odometry are calculated for the implementation of a controller in the real system. The planar decomposition is the state of the art approach for ballbot control. However, it mainly suffers from unphysical modeling of actuators requiring artificial calculations, and neglecting the strong coupling effects that are important for dynamic motion. To overcome these limitations, a full three-dimensional model is proposed. This new model is verified and the advantages are shown by comparing it to the planar model. The performance of a linear quadratic controller is simulated in real-time with MATLAB and displayed in a three-dimensional visualization. As a consequence of the strong nonlinearities, a controller based on gain scheduling shows to improve the performance significantly. With the resulting controller, the prototype is able to follow highly dynamic, complex trajectories, which is shown in various experiments. Compared to previous solutions, this fundamentally new approach to control a ballbot allows for a considerably higher performance regarding acceleration, speed, motion precision and robustness.

## Acknowledgements

We would like to thank a number of people who have encouraged and helped us in writing this thesis. We are very proud of the achieved results and we appreciate the support we received from all sides.

We are much obliged to Prof. Dr. Roland Y. Siegwart for his support and confidence in our team and for providing the opportunity to be part of such a far reaching project.

Special thanks to our supervisor and PhD student Stefan Leutenegger. His broad knowledge in system modeling and control and his very professional and friendly way of support had a wide influence on the results and also on our education. He persistently helped us with many complex questions and motivated us to extend the scope of the project into the right directions.

Many thanks to Dr. Francis Colas for the supervision of this thesis and also for coaching the whole focus project. Not only was he able to keep the very important overview, with a lot of effort and knowledge about each thesis, but also helped defining and managing all interconnections in order to bring all parts together.

We would like to thank Dr. Marcel Buttliger and Céline Martin for their extensive help with the structure and proofreading of this work.

We are very thankful to all members of Team Ballbot and are very happy to have spent a year working together in this project. We have formed a very capable team and helped each other to realize our ideas.

Most importantly, we are very grateful to all our friends and family members who supported us during the whole project and helped us to achieve our aims.

Zürich in June 2010,
Péter Fankhauser and Corsin Gwerder

# Contents

## Abbreviations

| | |
|---|---|
| 3D | Three Dimensional |
| ARM | Advanced RISC Machine |
| ASL | Autonomous Systems Lab |
| CAD | Computer Aided Design |
| CMU | Carnegie Mellon University |
| DC | Direct Current |
| DoF | Degrees of Freedom |
| EPOS | Positioning controller |
| ETH | Swiss Federal Institute of Technology |
| FIR | Finite Impulse Response |
| GUI | Graphical User Interface |
| IMU | Inertial Measurement Unit |
| LQR | Linear Quadratic Regulator |
| PD | Proportional–Derivative |
| PI | Proportional–Integral |
| PID | Proportional–Integral–Derivative |
| RISC | Reduced Instruction Set Computer |
| RS-232 | Recommended Standard 232 |
| TGU | Tohoku Gakuin University |
| UA | University of Adelaide |
| ZHAW | Zurich University of Applied Sciences |
| ZHdK | Zurich University of the Arts |

# 1 Introduction

## 1.1 Motivation

*Figure 1.1:* General concept of a ballbot

A ballbot uses a single ball as core of its propulsion system (fig. 1.1). Tilting spontaneously in any direction and rotating about its own axis, a ballbot performs its motion in a very unique way. The concept is very similar to an inverted pendulum. Especially designed as an unstable system, the fundamental part of a ballbot is a reliable control.
Other ballbots have been developed (see 1.5), but none of them is able to utilize the full potential of agility, which is a core feature of a ballbot. The prototype *Rezero* has been built with the goal of demonstrating an unprecedented level of agility and the capability of such a propulsion system.

## 1.2 Context

The presented thesis is closely linked to the ETH *Focus Project Ballbot 2009/10* [11]. In this project, eight undergraduate mechanical engineers from ETH Zurich, two electrical engineers studying at ZHAW[1] and three industrial design students from ZHdK[2] have developed a ballbot named *Rezero*[3] within nine months. The goal of this project was to develop a fully operational prototype of a ballbot which is unique in its motion, dynamics

[1] Zurich University of Applied Sciences
[2] Zurich University of the Arts
[3] The name *Rezero* refers to the important relationship between the real part of the poles $\pi$ of the transfer function and zero: a system with $\mathrm{Re}(\pi) > 0$ is unstable.

and agility compared to existing ballbots – a goal which can only be achieved with a strong focus on modeling and control.

## 1.3 Goals

This Bachelor thesis aims to analyze the system of the ballbot Rezero and to develop and implement an appropriate controller for it. The controller has to be optimized to ensure a highly dynamic and maneuverable motion of the ballbot. Additionally, a flexible realtime simulation of the system is required for early analysis and for the purpose of laying the foundation for other theses on the same prototype. The controller should comply with the conceptual requirements set forth by the Focus Project Ballbot as precisely as possible:

- different input modes (position, velocity, acceleration set points),
- speed up to 2 m/s,
- tilt angles up to 20°,
- high motion precision,
- high dynamic robustness.

Details are described in the project report „Rezero – dynamisch stabil auf einer Kugel“ [11].

## 1.4 System Overview

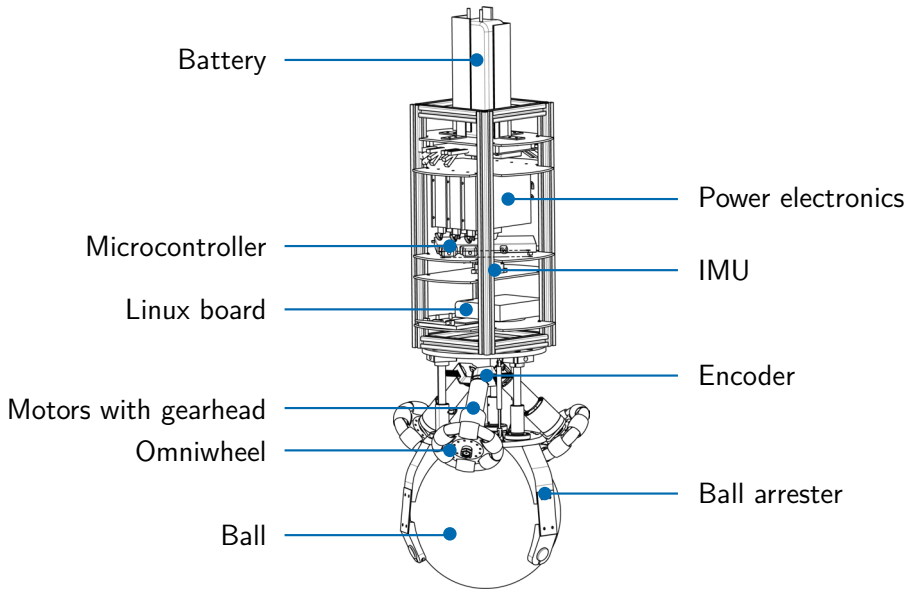



*Figure 1.2:* System description of Rezero

The system ballbot shown in figure 1.2 is a dynamically stable robot balancing and driving on a single ball. It is composed of three main parts: the ball, the propulsion

system and the body, the latter carrying all the electronics. The propulsive subsystem consists of three omniwheels[4] driven by 200 W DC brushless motors with gear heads with a maximum torque of 5 Nm. Each motor has an encoder attached, measuring the position and speed of the motor axis. The ball consists of a 2 mm thick aluminum hollow sphere, coated with a 4 mm high stiction synthetic. The ball arrester pushes the ball towards the omniwheels in order to increase the contact force.
A very important sensor is the inertial measurement unit (IMU) which measures all three Euler angles and the angular rates of the body. On the microcontroller, an ARM[5] 7, the controller is implemented running at 157.8 Hz. The set points are sent to the embedded microcontroller by a Linux board over RS-232.
A list of parameters is given in table 2.1 and details on all components are described in [11].

## 1.5 State of the Art

The first ballbot was developed in 2006 at Carnegie Mellon University (CMU) in the USA ([10],[12]). Another ballbot was developed at Tohoku Gakuin University (TGU) in Japan [7] and a third one as a student project 2009 at the University of Adelaide (UA) in Australia [5]. The specifications in table 1.1 show the differences.

| University | CMU | TGU | UA |
|---|---|---|---|
| Year | 2006–2010 | 2006–2008 | 2009 |
| Weight | 45 kg | 11 kg | 34.5 kg |
| Height | 150 cm | 50 cm | 160 cm |
| Maximum tilt angle | $<1^\circ$ | $<5^\circ$ | $<1^\circ$ |
| Maximum speed | $<0.1$ m/s | $<0.3$ m/s | $<0.1$ m/s |
| Rotation around its own axis | (yes) | yes | no |
| Used controller | LQR/PID | PD | LQRI |

*Table 1.1:* Specifications of already existing ballbots

The first version of the CMU ballbot was stabilized with a LQR (linear quadratic regulator) and an inner PI (proportional–integral) controller, the second version with a cascaded PID (proportional–integral–derivative) controller. The TGU ballbot uses a PD (proportional–derivative) controller. The UA ballbot is controlled with an LQR controller that uses an additional integrating part. It is important to mention that all these projects developed a controller based on a planar model as is described in section 2.1.

[4] A holonomic wheel that can assign forces into to the tangential direction and allows low-friction motion into the lateral direction.
[5] Advanced RISC Machine

## 1.6 Project Schedule and Procedure

The project described in this report is structured into chronological phases which end with the following milestones:

**17.03.2010:** Simulation environment is available to others

**29.03.2010:** Ballbot is balancing

**12.04.2010:** Ballbot is moving, state of the art

**26.04.2010:** Global system test 1

**10.05.2010:** Global system test 2

A detailed schedule is shown in the appendix A.
At the beginning of the project, the focus is set on the simulation, because other work within the focus project is heavily relying on it. The first system model is based on a planar simplification of the system. Based on the result of the simulation, the next step is to implement the controller and get the ballbot to keeping stability. With first tests, the controller is improved to reach the state of the art.
A new approach is taken by modeling the full three-dimensional system, allowing the design of a much better performing controller. With a test driven optimization, the controller is constantly improved and the results are verified in the global system tests 1 and 2.

## 1.7 Structure of the Report

The report is subdivided in seven major chapters. A first chapter describes the *Planar System Modeling and Control* (2), followed by the *Simulation and Verification of the Planar System* (3). The same structure is followed with the *3D System Modeling and Control* (4) and the *Simulation and Verification of the 3D System* (5), representing the chronological order of the development of the ballbot.
The results are discussed in *Performance Analysis* (6) and the details of the implementation process described in *Implementation* (7).

# 2 Planar System Modeling and Control

## 2.1 Model

### 2.1.1 Model Description

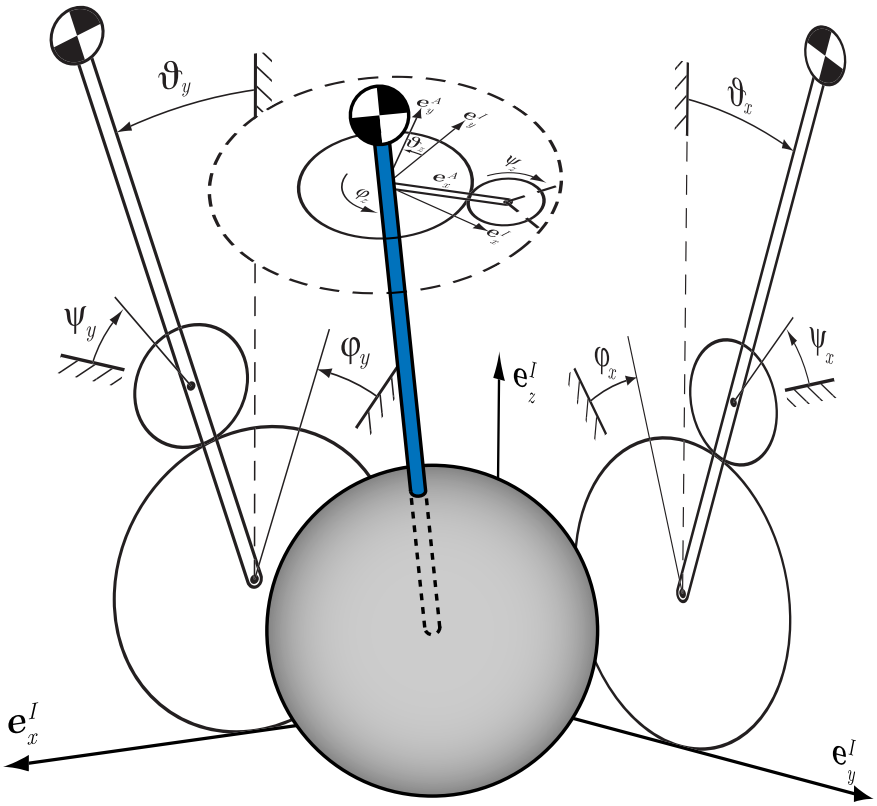



*Figure 2.1:* The three planar models, each model consists of the ball, the body and the actuating wheel

The three-dimensional system is divided into three independent planar models (fig: 2.1), where each can be described with the following two Degrees of Freedom (DoF):

- 1 DoF for the rotation/translation of the ball
- 1 DoF for the rotation of the body

The propulsion system, which is composed of three omniwheels, is modeled as one virtual actuating wheel in each plane which drives the ball. That particular wheel does of course not have the same position and speed as the omniwheels in the real system. Therefore, conversions are needed (see section 2.3).

The states of the system are the orientation angles and angle rates of the body and the position and speed of the ball. Since all these states are measured by the IMU and the motor encoders, the system outputs are equal to the states. The inputs are the torques generated by the motors.

### 2.1.2 Assumptions

In the planar system modeling, the system is treated as three independent planar models. Therefore no coupling effects between these three models are taken into account. Two of them (the ones in the $xz$- and $yz$-plane) are identical and very similar to an inverted pendulum, as shown on the left side of figure 2.2. The third one (in the $xy$-plane) describes the rotation around the $z$-axis in the body fixed reference frame, often described as the rotation around the ballbot's own axis (right side of figure 2.2).
The system is assumed to consist of three rigid bodies: the ball, the actuating wheel and the body. The three motors and omniwheels are therefore simplified as one actuating wheel that fits in a plane.
Additionally, the following assumptions are made:

**No slip** The contact points between the ball and the ground and between the wheels and the ball are assumed to be free of slippage. The system construction is optimized in order to minimize possible slipping. Thus, the torques applied have to be limited within the range that does not cause slippage in order to ensure the validity of the model.

**No friction** Even though friction occurs in the system, it is neglected except for the rotation of the ball on the ground around the $z$-axis, because this is the friction with the biggest impact. Non-continuous friction is hard to represent analytically and would generate equations much more complex than without friction. Also, the precise measuring of friction is difficult.

**No deformation** Because of its synthetic coating, deformation appears at the contact points of the ball. This has to be neglected in order to keep complexity at a manageable level.

**Fast motor dynamics** The power electronics control the dynamics of the motors much faster than the controller the dynamics of the stabilized system. Consequently, the motor torques are modeled to be the inputs of the system.

**Ball moves only horizontally** In order to keep the prerequisite for the no slip condition, vertical movement of the ball has to be neglected. Finally, this model is designed to move primarily on flat surfaces without steep inclination.

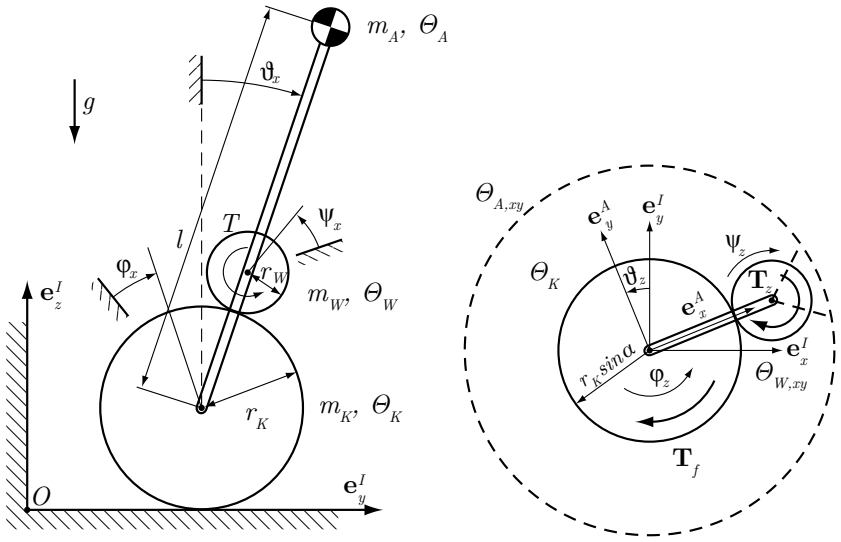



*Figure 2.2:* Sketch of the planar models showing the coordinates and parameters. On the left side the model for the two vertical planes and on the right side the $xy$-plane model is shown.

### 2.1.3 Coordinates

To describe the system, the coordinates shown in figure 2.2 are used. $\vartheta_{x,y,z}$ represent the orientation of the body, $\varphi_{x,y,z}$ represent the orientation of the ball and $\psi_{x,y,z}$ are the angles of the virtual actuating wheels. Using the following minimal coordinates, the system can be completely described.

$$\vec{q}_{xy} = \begin{bmatrix} \varphi_z \\ \vartheta_z \end{bmatrix} \quad \vec{q}_{yz} = \begin{bmatrix} \varphi_x \\ \vartheta_x \end{bmatrix} \quad \vec{q}_{xz} = \begin{bmatrix} \varphi_y \\ \vartheta_y \end{bmatrix} \tag{2.1}$$

### 2.1.4 Binding Equations

All coordinates can be expressed as functions of the minimal coordinates, which is necessary in order to derive the equations of motion. The positions of the ball, the wheel and the body can be written as follows, using the rolling constraint[1].

$$x_K = \varphi_x r_K \tag{2.2}$$

$$x_W = \varphi_x r_K + \sin\vartheta_x \cdot (r_K + r_W) \tag{2.3}$$

$$x_A = \varphi_x r_K + \sin\vartheta_x \cdot l \tag{2.4}$$

Analogous relations are valid for the $y$ coordinates.

[1] Meaning, that for example the balls position is determined by its slip free rolling on the ground.

The rotation of the actuating wheel expressed as a function of the minimal coordinates yields the following equations.

$$\dot{\psi}_x = \frac{r_K}{r_W}\left(\dot{\varphi}_x - \dot{\vartheta}_x\right) - \dot{\vartheta}_x \tag{2.5}$$

$$\dot{\psi}_y = \frac{r_K}{r_W}\left(\dot{\varphi}_y - \dot{\vartheta}_y\right) - \dot{\vartheta}_y \tag{2.6}$$

$$\dot{\psi}_z = \frac{r_K}{r_W} \cdot \sin\alpha \cdot \left(\dot{\varphi}_z - \dot{\vartheta}_z\right) \tag{2.7}$$

The derivation can be found in the appendix B.1.

### 2.1.5 Forces

In order to get an idea of the forces taking effect in the system, a Newton/Euler approach is used to derive analytical equation for the forces shown in figure 2.3. The detailed equations are listed in the appendix B.3.

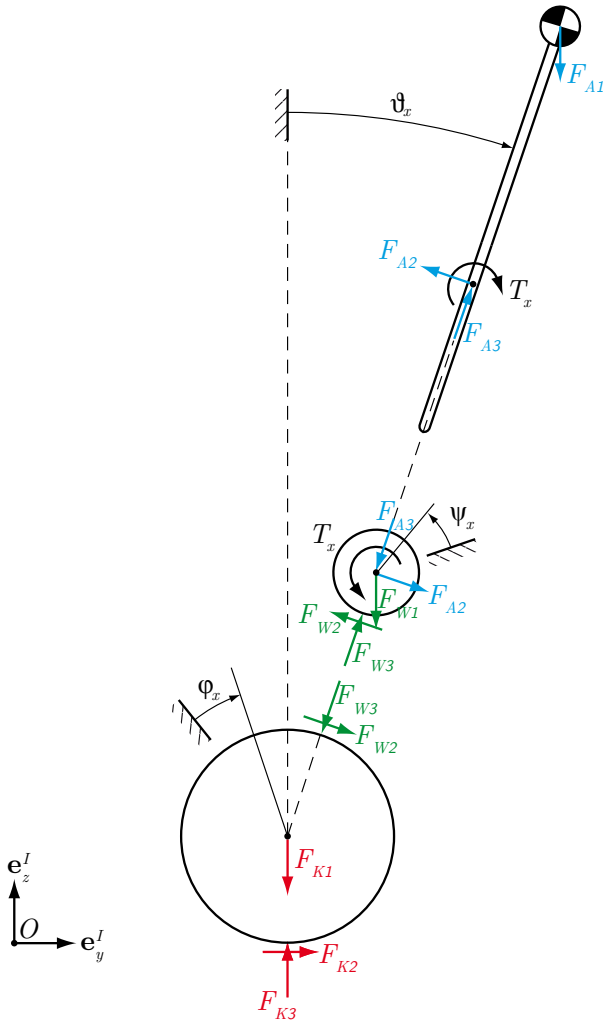



*Figure 2.3:* Free-body diagrams for the three parts of the model, showing all internal forces

### 2.1.6 Parameters

In table 2.1, the parameters describing the system are listed. Numeric values are derived by measuring or from the CAD model. Calculations of the moments of inertia are presented in section 2.3.2.

| Description | Variable | Value |
|---|---|---|
| Mass of the ball | $m_K$ | 2.29 kg |
| Mass of the virtual actuating wheel | $m_W$ | 3 kg |
| Mass of the body | $m_A$ | 9.2 kg |
| Radius of the ball | $r_K$ | 0.125 m |
| Radius of the omniwheels | $r_W$ | 0.06 m |
| Radius of the body (cylinder) | $r_A$ | 0.1 m |
| Height of the center of gravity | $l$ | 0.339 m |
| Inertia of the ball | $\Theta_K$ | 0.0239 kgm$^2$ |
| Inertia of the actuating wheel in the $yz$-/$xz$-plane | $\Theta_W$ | 0.00236 kgm$^2$ |
| Inertia of the actuating wheel in the $xy$-plane | $\Theta_{W,xy}$ | 0.00945 kgm$^2$ |
| Inertia of the body | $\Theta_A$ | 4.76 kgm$^2$ |
| Inertia of the body in the $xy$-plane | $\Theta_{A,xy}$ | 0.092 kgm$^2$ |
| Gravitational acceleration | $g$ | 9.81 m/s$^2$ |
| Gear ratio | $i_{Gear}$ | 26 |

*Table 2.1:* Parameters of the planar system

## 2.2 Dynamics

### 2.2.1 Approach

For the derivation of the equations of motion, the Lagrangian method was selected as the common approach for that kind of problems. The following steps are needed:

- express kinetic (T) and potential (V) energy of all rigid bodies as a function of the minimal coordinates,
- write non-potential forces as a function of the minimal coordinates,
- solve the Lagrange equation for the second derivative of the minimal coordinates.

The calculations are made for the $yz$-plane model, which is identical to the $xz$-plane model, and for the $xy$-plane model as well.

### 2.2.2 Energies

The kinetic energy of the ball in the $yz$-plane contains a translational and a rotational part.

$$T_{K,yz} = \underbrace{\frac{1}{2} \cdot m_K \cdot (r_K \cdot \dot{\varphi}_x)^2}_{Translation} + \underbrace{\frac{1}{2} \cdot \Theta_K \cdot \dot{\varphi}_x^2}_{Rotation} \tag{2.8}$$

The potential energy of the ball is zero, when choosing a coordinate frame in the center of the ball.

$$V_{K,yz} = 0 \tag{2.9}$$

The kinetic energy of the ball in the $xy$-plane consists of a rotational part only due to the fact that it describes only the rotation around the $z$-axis. For the same reason, the potential energy for the $xy$-plane does not need to be considered.

$$T_{K,xy} = \frac{1}{2} \cdot \Theta_K \cdot \dot{\varphi}_z^2 \tag{2.10}$$

The kinetic energy of the actuating wheel in the $yz$-plane is equally composed of a translational and a rotational part, but there is an additional coupling term. The complete derivation of the equations of motion can be found in the appendix B.1.

$$\begin{aligned} T_{W,yz} = \frac{1}{2} m_W \Bigg( & \underbrace{(r_K \dot{\varphi}_x)^2}_{Translation} + \underbrace{2 \cdot (r_K + r_W) \cos\vartheta_x \dot{\vartheta}_x (r_K \dot{\varphi}_x) + (r_K + r_W)^2 \dot{\vartheta}_x^2}_{Coupling} \Bigg) \\ & + \underbrace{\frac{1}{2} \Theta_W \left( \frac{r_K}{r_W} (\dot{\varphi}_x - \dot{\vartheta}_x) - \dot{\vartheta}_x \right)^2}_{Rotation} \end{aligned} \tag{2.11}$$

The potential energy of the actuating wheel is a simple function of the tilt angle of the body $\vartheta_x$.

$$V_{W,yz} = m_W \cdot g \cdot (r_K + r_W) \cos\vartheta_x \tag{2.12}$$

For the $xy$-plane, the kinetic energy of the motor and wheel is similar to that of the ball.

$$T_{W,xy} = \frac{1}{2} \cdot \Theta_{W,xy} \cdot \dot{\psi}_z^2 \tag{2.13}$$

The energies of the body are again very similar to the ones of the actuating wheel.

$$T_{A,yz} = \frac{1}{2} m_A \left( (r_K \dot{\varphi}_x)^2 + 2 \cdot l \cos \vartheta_x \dot{\vartheta}_x (r_K \dot{\varphi}_x) + l^2 \dot{\vartheta}_x^2 \right) + \frac{1}{2} \Theta_A \dot{\vartheta}_x^2 \tag{2.14}$$

$$V_{A,yz} = m_A \cdot g \cdot l \cdot \cos \vartheta_x \tag{2.15}$$

$$T_{A,xy} = \frac{1}{2} \cdot \Theta_{A,xy} \cdot \dot{\vartheta}_z^2 \tag{2.16}$$

### 2.2.3 Non-Potential Forces

Non-potential forces $f_{NP}$ represent the external forces acting on a system. In this case, the motor torque $T_x$ i.e. the system input is a non-potential force. The motor torque takes a direct effect on the coordinate $\psi_x$. Using equation (2.5), the effect of the motor torque on the minimal coordinates can be expressed:

$$\vec{f}_{NP,yz1} = \begin{bmatrix} \frac{r_K}{r_W} \cdot T_x \\ -\left(1 + \frac{r_K}{r_W}\right) T_x \end{bmatrix} \tag{2.17}$$

The counter torque acting on the coordinate $\vartheta_x$ of the body is expressed as:

$$\vec{f}_{NP,yz2} = \begin{bmatrix} 0 \\ T_x \end{bmatrix} \tag{2.18}$$

The total non-potential force is then the sum of $\vec{f}_{NP,yz1}$ and $\vec{f}_{NP,yz2}$. A similar procedure is applied in the $xy$-plane. In this model, an additional friction force $T_f$ is acting on $\varphi_z$ as the rotation of the ball:

$$\vec{f}_{NP,xy} = \begin{bmatrix} -T_f + \frac{r_K}{r_W} \cdot \sin \alpha \cdot T_z \\ -\frac{r_K}{r_W} \cdot \sin \alpha \cdot T_z + T_z \end{bmatrix} \tag{2.19}$$

### 2.2.4 Equations of Motion

Now, the equations of motion are derived by solving the Lagrange equation for $\ddot{\vec{q}}$.

$$\frac{\mathrm{d}}{\mathrm{d}t} \left( \frac{\partial T}{\partial \dot{\vec{q}}} \right)^T - \left( \frac{\partial T}{\partial \vec{q}} \right)^T + \left( \frac{\partial V}{\partial \vec{q}} \right)^T - \vec{f}_{NP} = 0 \tag{2.20}$$

$T$ and $V$ denote the total kinetic and potential energy respectively.

$$T = T_K + T_W + T_A$$
$$V = V_K + V_W + V_A$$

Finally, the equations of motion for the planar inverted pendulum in the $yz$-plane can be written in matrix form as follows.

$$M_x(\vec{q},\dot{\vec{q}})\ddot{\vec{q}} + C_x(\vec{q},\dot{\vec{q}}) + G_x(\vec{q}) = f_{NP} \tag{2.21}$$

The matrices containing masses and inertias $M_x$, Coriolis forces $C_x$ and gravitational forces $G_x$ are:

$$M_x = \begin{bmatrix} m_{tot}r_K^2 + \Theta_K + \left(\frac{r_K}{r_W}\right)^2 \Theta_W & -\frac{r_K}{r_W^2} r_{tot}\Theta_W + \gamma r_K \cos\vartheta_x \\ -\frac{r_K}{r_W^2} r_{tot}\Theta_W + \gamma r_K \cos\vartheta_x & \frac{r_{tot}^2}{r_W^2}\Theta_W + \Theta_A + m_A l^2 + m_W r_{tot}^2 \end{bmatrix} \tag{2.22}$$

$$C_x = \begin{bmatrix} -r_K \gamma \sin\vartheta_x \dot{\vartheta}_x^2 \\ 0 \end{bmatrix} \tag{2.23}$$

$$G_x = \begin{bmatrix} 0 \\ -g \sin\vartheta_x \gamma \end{bmatrix} \tag{2.24}$$

The following substitutions were used in the equations above.

$$\begin{aligned} m_{tot} &= m_K + m_A + m_W \\ r_{tot} &= r_K + r_W \\ \gamma &= l \cdot m_A + (r_K + r_W) m_W \end{aligned}$$

The Lagrange equation of the planar model in the $xy$-plane solved for $\ddot{\varphi}_z$ and $\ddot{\vartheta}_z$ yield the following equations of motion.

$$\ddot{\varphi}_z = -\frac{(r_W^2\Theta_{A,xy} + r_K^2\Theta_{W,xy}\sin^2\alpha)\cdot T_f + r_K r_W \Theta_{A,xy}\sin\alpha \cdot T_z}{r_W^2\Theta_{A,xy}\Theta_K + r_K^2(\Theta_{A,xy} + \Theta_K)\Theta_{W,xy}\sin^2\alpha} \tag{2.25}$$

$$\ddot{\vartheta}_z = -\frac{r_K \sin\alpha (r_K\Theta_{W,xy}\sin\alpha \cdot T_f + r_W\Theta_K \cdot T_z)}{r_W^2\Theta_{A,xy}\Theta_K + r_K^2(\Theta_{A,xy} + \Theta_K)\Theta_{W,xy}\sin^2\alpha} \tag{2.26}$$

To compute the friction force, it is assumed that the ball always sticks on the ground, thus the condition for stiction $\ddot{\varphi} = 0$ has to be fulfilled. With equation (2.25), $T_f$ can be written as follows:

$$T_f = \frac{r_K r_W \Theta_{A,xy}\sin\alpha \cdot T_z}{r_W^2\Theta_{A,xy} + r_K^2\Theta_{W,xy}\sin^2\alpha} \tag{2.27}$$

## 2.3 Conversions to the Real System

### 2.3.1 Torque Conversion

The planar model uses a virtual wheel to actuate the system. The real system has an actuating structure which differs strongly from the one assumed in the planar model.

Since a controller for the planar model is going to be implemented on the real system, conversions have to be calculated. In order to be able to control the real system, the torques on the virtual motors have to be converted into the torques for the real motors.

### Definitions

One element of the real system (omniwheel driven by a motor) is shown on the left side in figure 2.4. It actuates the ball by scrolling on a circle on the ball, characterized by the motor arrangement angle $\alpha$.
The right side of figure 2.4 shows a top view on the real actuating system where the torque of each omniwheel generates a tangential force on the surface of the ball.

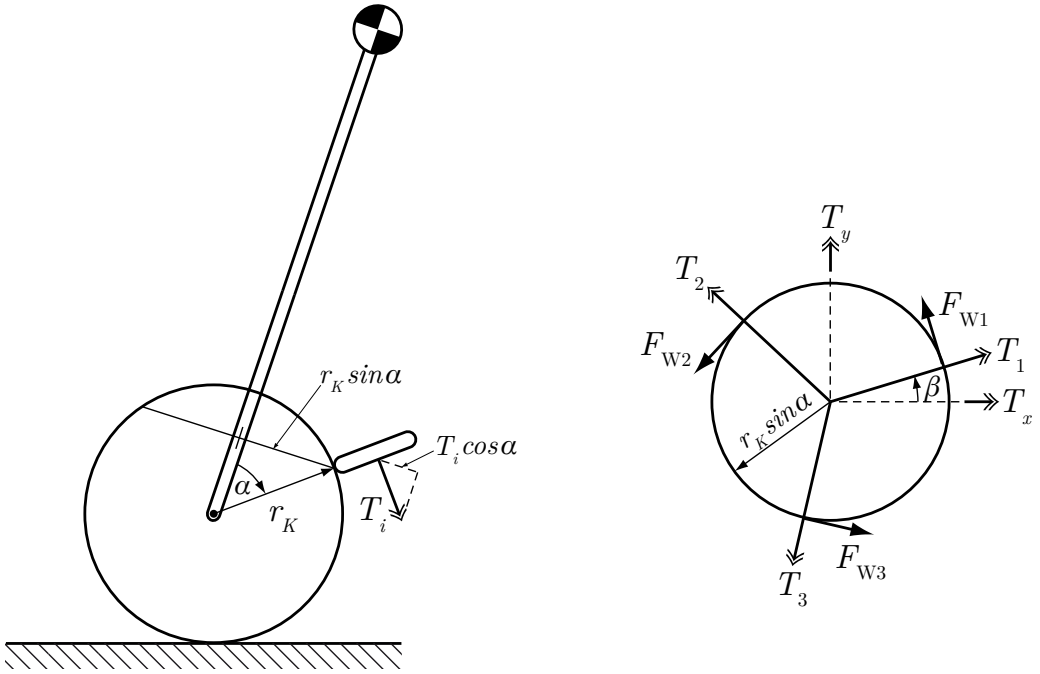



*Figure 2.4:* On the left side the arrangement of one real omniwheel is shown. The sketch on the right side shows the torques and tangential forces generated by the real actuating system, the dashed arrows mark the virtual torques.

$F_{W,1}$, $F_{W,2}$, $F_{W,3}$ in figure 2.4 denote the tangential forces on the ball generated by torques of the motors in the real configuration, consequently these are functions of $T_{1,2,3}$. The levers for this configuration are denoted as $r_{KW,1}$, $r_{KW,3}$, $r_{KW,3}$. For the virtual configuration, the notation $F_{W,x}$, $F_{W,y}$, $F_{W,z}$ is used for the tangential forces (as functions of $T_{x,y,z}$) and $r_{KW,x}$, $r_{KW,y}$, $r_{KW,z}$ for the levers. All those are geometrical relationships one can read out of figure 2.4. In the appendix C.1 the relationships are written out in full.

#### Torque on the ball

Once the relationships are given, the torque on the ball from the real and the virtual driving mechanism can be calculated. The torque is equal to the product of force and lever. Hence the torques for the real and the virtual configuration are calculated as follows.

$$T_{KW,i} = r_{KW,i} \times F_{W,i} \qquad \text{for } i = 1,2,3 \tag{2.28}$$

$$T_{KW,j} = r_{KW,j} \times F_{W,j} \qquad \text{for } j = x,y,z \tag{2.29}$$

#### Solution

Of course, the overall torque needs to be conserved:

$$T_{KW,1} + T_{KW,2} + T_{KW,3} = T_{KW,x} + T_{KW,y} + T_{KW,z} \tag{2.30}$$

Solving the relations for the real torques yields:

$$T_1 = \frac{1}{3} \cdot \left( T_z + \frac{2}{\cos\alpha} \cdot (T_x \cdot \cos\beta - T_y \cdot \sin\beta) \right) \tag{2.31}$$

$$T_2 = \frac{1}{3} \cdot \left( T_z + \frac{1}{\cos\alpha} \cdot \left( \sin\beta \cdot (-\sqrt{3}T_x + T_y) - \cos\beta \cdot (T_x + \sqrt{3}T_y) \right) \right) \tag{2.32}$$

$$T_3 = \frac{1}{3} \cdot \left( T_z + \frac{1}{\cos\alpha} \cdot \left( \sin\beta \cdot (\sqrt{3}T_x + T_y) + \cos\beta \cdot (-T_x + \sqrt{3}T_y) \right) \right) \tag{2.33}$$

Since the system of equation is linear in $T_{x,y,z}$, it is straightforward to solve for them as well (see appendix appendix C).

### 2.3.2 Inertia Calculations

The aim of this section is to find a suitable approximation for the moments of inertia of the modeled drive mechanism, which is not the same as in reality. The following approach is made by comparing the energies stored in the real omniwheels and the virtual actuating wheel respectively when driving at a constant velocity.

#### Approximations for sphere and body

The inertia of the ball is approximated by a hollow sphere, which is a good approximation:

$$\Theta_K = \frac{2}{3} \cdot m_K \cdot r_K^2$$

The moment of inertia of the body was initially approximated by that of a cylinder. Later, it was directly calculated in the CAD model.

**Conversion for motors and omniwheels**

The real rotor inertia of a motor is given by

$$\Theta_M = 3.33 \cdot 10^{-6}\,\mathrm{kgm}^2$$

and the real inertia of an omniwheel is calculated as:

$$\Theta_{OW} = \frac{1}{2} \cdot m_{OW} \cdot r_W^2 = 900 \cdot 10^{-6}\,\mathrm{kgm}^2$$

Since the reduction of the gear box is $i = 26$, the rotor of the motor turns 26 times faster. Therefore, a factor $i^2$ is added to calculate the energy ($E = \frac{1}{2} \cdot \Theta \cdot v^2$). So both, omniwheel and motor inertias are taken into account. The inertia of the gear rotor is $9.1 \cdot 10^{-6}$ kgm$^2$ and therefore negligible.
In order to obtain an approximation of the inertia of the virtual actuating wheel $\Theta_{OW,x}$, the rotation speed of each real omniwheel is calculated, while turning the virtual omniwheel at a specific speed. The rotational energy of the virtual wheel is then equated to the sum of the energies of the real wheels. This equation can be solved for the virtual inertia. The following results were obtained:

$$\Theta_{W,x} = \Theta_{W,y} = \frac{3}{2} \cdot \cos^2(\alpha)(\Theta_{OW} + i^2 \cdot \Theta_M) \tag{2.34}$$

$$\Theta_{W,z} = 3 \cdot (\Theta_{OW} + i^2 \cdot \Theta_M) \tag{2.35}$$

Comparing the numerical values for the energies of the ball, the actuating wheel and the body (assuming speeds of $v = 3$ m/s and $\dot{\vartheta}_x = 1$ rad/s) shows that all three values have the same order of magnitude (equations 2.36–2.38). Hence, it is appropriate to take all these energies into account for the modeling.

$$E_K = 6.88\,\mathrm{J} \tag{2.36}$$

$$E_A = 1.48\,\mathrm{J} \tag{2.37}$$

$$E_W = 2.95\,\mathrm{J} \tag{2.38}$$

## 2.4 Development of a Planar Model Based Controller

### 2.4.1 System Analysis

Based on the model introduced above, a linear controller is designed. Therefore, the equations have to be linearized. In order to obtain the linear equations, the equations of motion are written in state space form (2.39).

$$\dot{\vec{x}} = \begin{bmatrix} \vec{q} \\ \dot{\vec{q}} \end{bmatrix}^{\cdot} = \begin{bmatrix} \dot{\vec{q}} \\ M_x^{-1}(f_{NP} - (C_x + G_x)) \end{bmatrix} \tag{2.39}$$

Since the expected operating point values and input signals have the same order of magnitude, no normalization is used.[2]
With the state vector $\vec{x} = [\varphi_x, \vartheta_x, \dot{\varphi}_x, \dot{\vartheta}_x]$, the input $u = T_x$ and the system output $\vec{y} = \vec{x}$, meaning that the output is equal to the states, because all states are measured with the sensors, the linearized state space description is

$$\dot{\vec{x}} = A \cdot \vec{x} + B \cdot \vec{u} \ , \qquad \vec{y} = C \cdot \vec{x} + D \cdot \vec{u} \ . \tag{2.40}$$

Next, the matrices can be derived using a Taylor expansion of the equation of motion. Substituting the variables with numerical values, one gets the following matrices for the $yz$-plane/$xz$-plane model:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -31.4603 & 0 & 0 \\ 0 & 16.3333 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0 \\ 0 \\ 13.4993 \\ -2.8567 \end{bmatrix} C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The controllability matrix as well as the observability matrix have full rank, which means that the system is fully controllable and fully observable and thus stabilizable. One pole of the system with positive real part show its instability. Similar to an inverted pendulum, a ballbot has a non-minimum phase behavior.[1]
Carrying out the linearization for the $xy$-plane model in an equivalent way yields:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -17.2267 \end{bmatrix} C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

### 2.4.2 Controller Design

For the two planar inverted pendulums a linear quadratic state feedback controller (LQR) is designed. In the LQR problem a solution minimizing the cost criterion

$$J(u) = \int_0^\infty \left( x^T(t) \cdot Q \cdot x(t) + u^T(t) \cdot R \cdot u(t) \right) \mathrm{d}t \tag{2.41}$$

is sought.[2] In order to apply LQR, full state feedback is used. In many systems it is hard to measure all states and therefore a model based observer has to be used. If all states are available, the solution is $u(t) = -K \cdot x(t)$, where $K = R^{-1} \cdot B \cdot \Phi$ and $\Phi$ is the only positive definite solution of the Riccati equation:

$$\Phi \cdot B \cdot R^{-1} \cdot B^T \cdot \Phi - \Phi \cdot A - A^T \cdot \Phi - Q = 0 \tag{2.42}$$

[2]Normalization is used for preventing numerical problems.

Since all states are estimated by the sensors in the system discussed, the application of an LQR controller is straightforward. The weighting matrices $Q$ and $R$ are experimentally tuned (view section 2.4.3). For given $Q$ and $R$ matrices, the controller gains $K$ can be derived using the MATLAB `lqr` command. Since the LQR controller is designed for a set point equal to zero, an additional factor to the set point is needed, in order to achieve a correct tracking behavior. This gain is calculated as proposed in [3]:

$$N = -\left(C(A - BK)^{-1}B\right)^{-1} \tag{2.43}$$

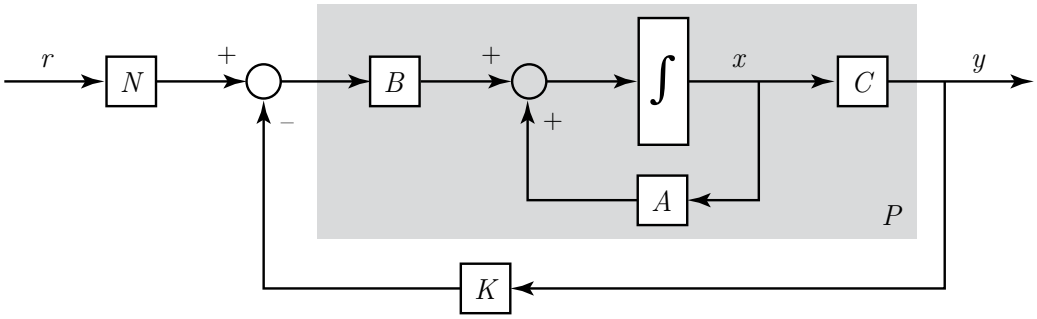



*Figure 2.5:* Block schema of the LQR controller with the linear model $P$ (shaded in grey)

Three different controllers are implemented on the system. First, a position controller which uses all states to stabilize the system and accepts position set points. Furthermore, a velocity controller that does not stabilize the position, thus only uses the states $x = [\vartheta_x, \dot{\varphi}_x, \dot{\vartheta}_x]$, is implemented. Finally, an angle controller which uses solely the states $x = [\vartheta_x, \dot{\vartheta}_x]$ and no information about position and speed.
For the relatively simple $xy$-plane model, a PD controller is experimentally developed. With the parameters $k_p = -0.5$ and $k_d = -0.3$, the rotation angle $\vartheta_z$ can be stabilized. Because of developing a controller for the full three dimensional system, these parameters were not tuned any further.

### 2.4.3 Tuning on the Real System

The initial guess for the weighting matrices $Q$ and $R$, which worked well for the simulation, was found to be much too aggressive to be applied on the real system. The problem is, that the noise on the gyroscope values (corresponding to the state $\dot{\vartheta}$) is amplified by the controller and thus causes the system to start trembling also linked with backlash of the gear box. Therefore, in order to lower the controller gains, the parameter $R$ was iteratively increased. In addition, the damping coefficients, i.e. the factors in the matrix $Q$, which weights the velocities, were raised too. Finally, the following matrices are found

to generate controller gains which stabilize the ballbot without causing trembling:

$$Q = \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix} \qquad R = 200$$

The controller gains calculated with these weighting matrices are:

$$K = \begin{bmatrix} -0.3162 & -22.8608 & -0.5359 & -6.5726 \end{bmatrix}$$

In order to improve the performance of the controller, a finite impulse response (FIR) low-pass filter of fourth order with a cutoff frequency of approximately 15 Hz (view section 3.4.2) is applied on the controller output. Therefore the gains can be increased, while the system remains stable.
Finally, the controller gains are slightly modified by hand to improve the performance. The gains used in the end are the following:

$$K = \begin{bmatrix} -0.8944 & -60 & -1.5 & -10 \end{bmatrix}$$

# 3 Simulation and Verification of the Planar Model

## 3.1 Implementation in Simulink

The nonlinear equations of the planar model are implemented in Simulink (fig. 3.1) for model verification. First, simulations without controller showed that the system behaves as expected. The simulation is also used to develop and test the controller and any further software that is going to be implemented on the real system. Therefore, the simulation should fit the reality as precisely as possible. In section 3.3 the approach to improve the simulation in that way is explained.

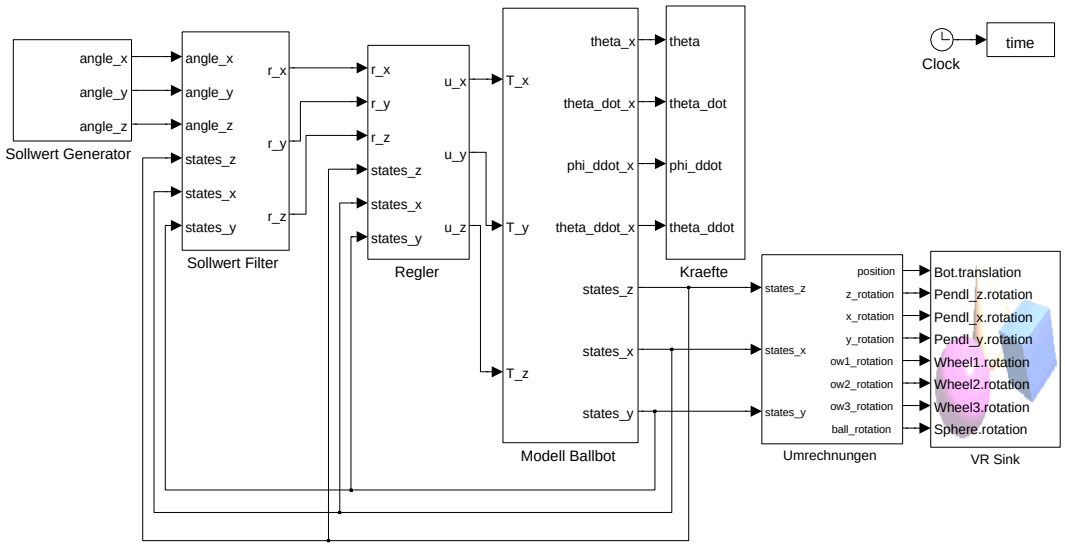



*Figure 3.1:* Simulink block scheme of the complete simulation environment

For handling the simulation a simple to use graphical user interface (GUI) is programmed in MATLAB (figure 3.2), where the control method and the input can be chosen. When starting the simulation, the real time visualization (section 3.2) is shown. After the simulation is stopped, the plots are generated.

## 3.2 Visualization

To visualize the data generated in the simulation, the *3D Animation Toolbox* of MATLAB is used, which is a relatively simple way to achieve a three-dimensional visualization of the model in real time. Figure 3.3 shows a screenshot of the visualization.

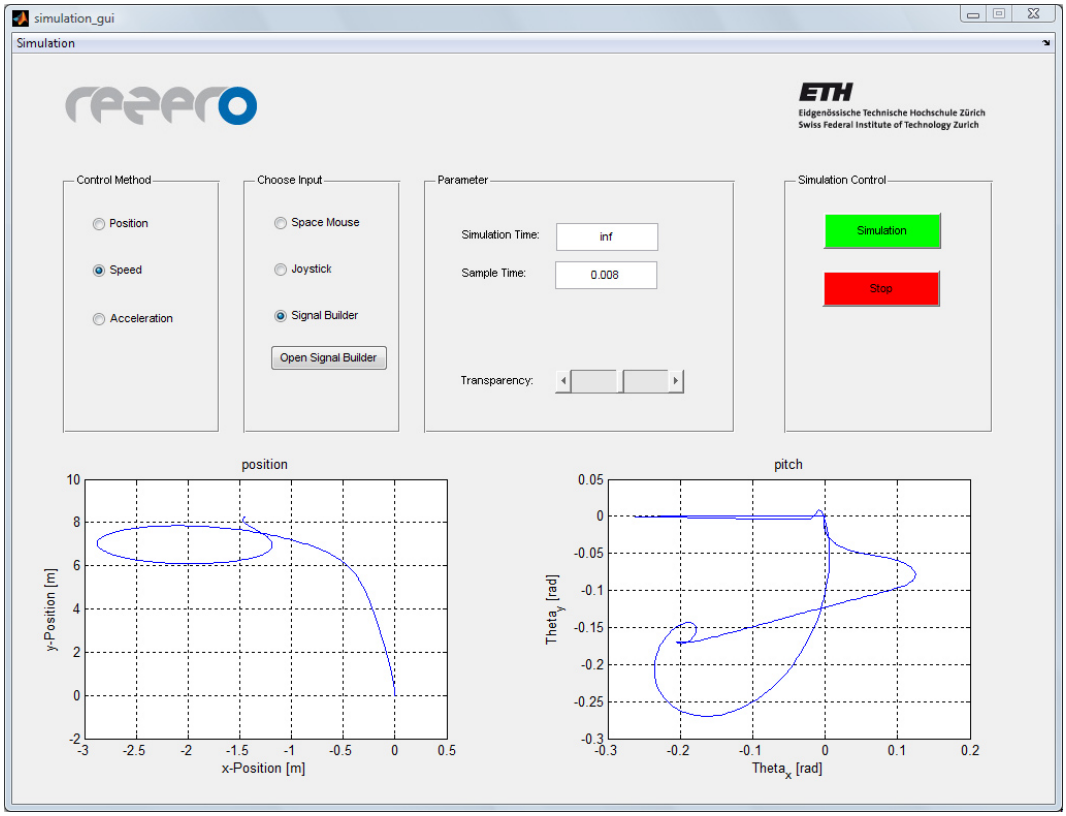



*Figure 3.2:* Screenshot of the GUI for the simulation environment

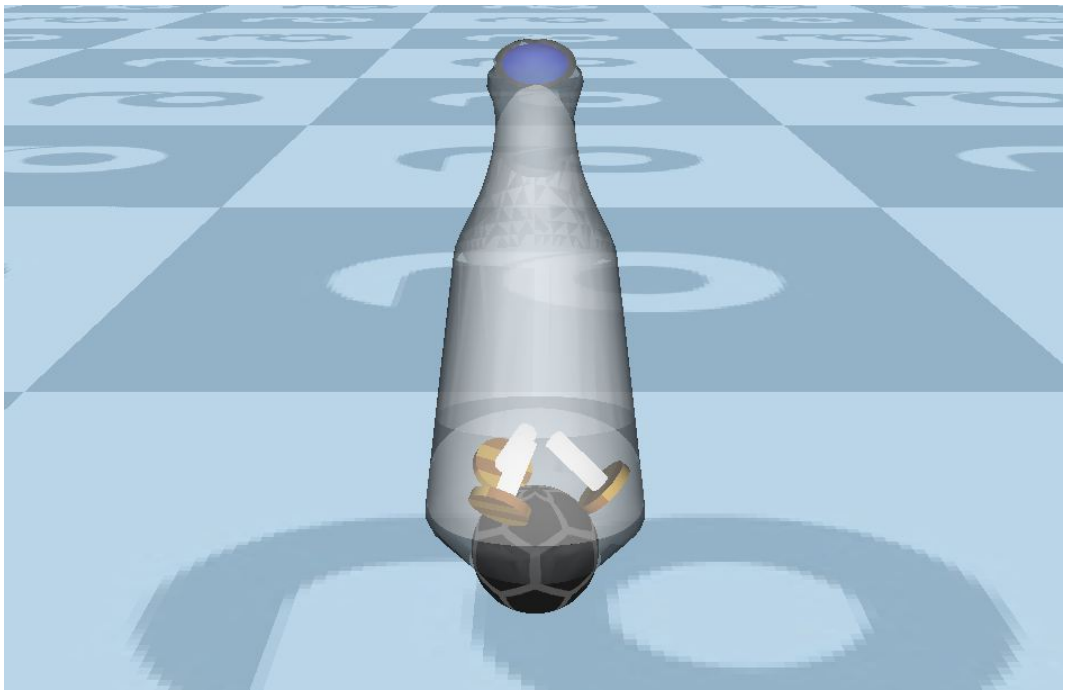


*Figure 3.3:* Screenshot of the virtual reality environment in Simulink

## 3.3 Error Model

To get an idea of the effect of noise and delay occurring in the real system, simple assumptions are made and simulated in Simulink.

### 3.3.1 Noise

The only noise that is assumed to be relevant is the sensor noise from the angle estimation and from the gyroscopes of the IMU. The discretization error of the motor encoder is neglected. The dynamic IMU noise is measured by rotating the IMU constantly around each axis separately. It is assumed that the rotation contains no high frequency oscillating, therefore, by subtracting the rotation speed from the measured data, the magnitude of the dynamic noise can be estimated. As shown in figure 3.5, the magnitude measured in this test is approximately $\pm 0.05$ rad/s at a sample time of 157.8 Hz. This data is then directly added to the angle rate (signal $n$ in figure 3.4).

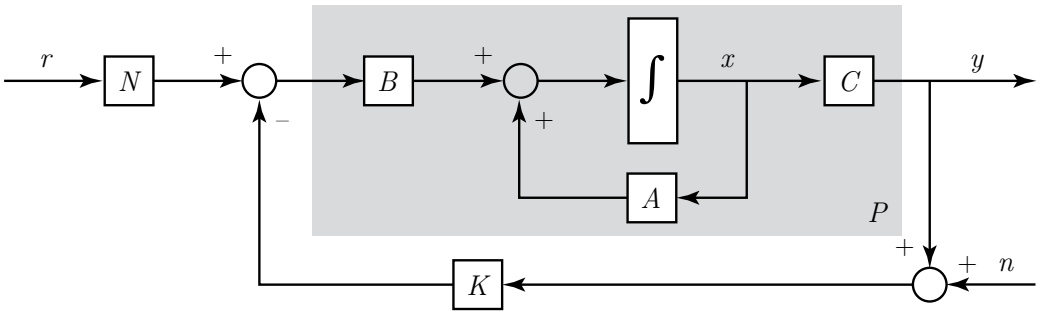



*Figure 3.4:* LQR feedback control schema with sensor noise $n$

### 3.3.2 Delay

In addition to the sensor noise, a delay of 0.01 s is added to the plant, representing the delay occurring in the real system by signal processing (a sample rate of 160 Hz means a maximum delay of 0.007 s).

## 3.4 Filtering

As already discussed in section 2.4.3, noise and other non-ideal effects are the main reason why the controller cannot be tuned in the desired way. Particularly the noise is limiting the magnitude of the controller gains and therefore the performance of the controller. In order to counteract these undesired effects, different solutions are tried.

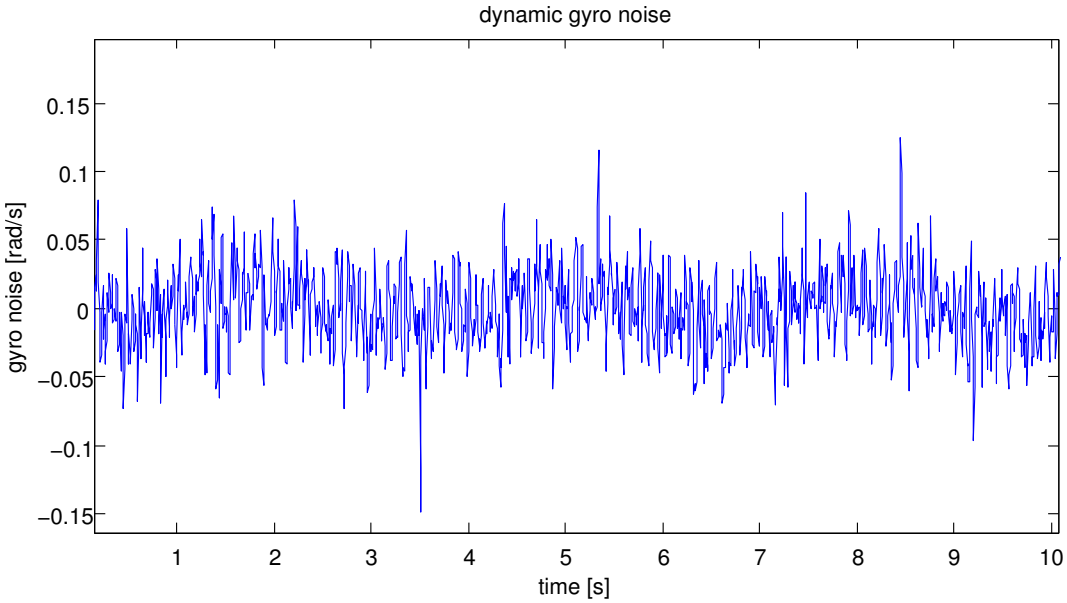



*Figure 3.5:* Measured data from the IMU gyroscopes, showing the noise.

### 3.4.1 Offset

Because of the friction, the motors do not start turning unless they get more than a specified amount of torque as input. The current needed to start the motors turning is experimentally determined and amounts to approximately 80 mA, what corresponds to a input torque of 42 mNm. An offset with that threshold was tested on the ballbot without improvement and is therefore not used.

### 3.4.2 Low Pass Filter

The noticeable amount of noise on the gyroscope values (fig. 3.5) causes the output of the controller to be very noisy too, which is shown by a trembling of the system. By applying a low pass filter on the controller output, this effect is eliminated. The low pass is derived using the *Filter Design Toolbox* `fdatool` of MATLAB. It is a fourth order finite impulse response (FIR) filter with a cutoff frequency of approximately 15 Hz.

### 3.4.3 Kalman Filter

By applying a Kalman filter on the sensor values and estimating the gyroscope values using the model, it is also possible to eliminate the trembling. As can be see in figure 3.7, it has the same effect as the low pass filter, whereas it does not cause as much delay as the low pass filter. The development of a Kalman filter is part of the bachelor thesis [4].

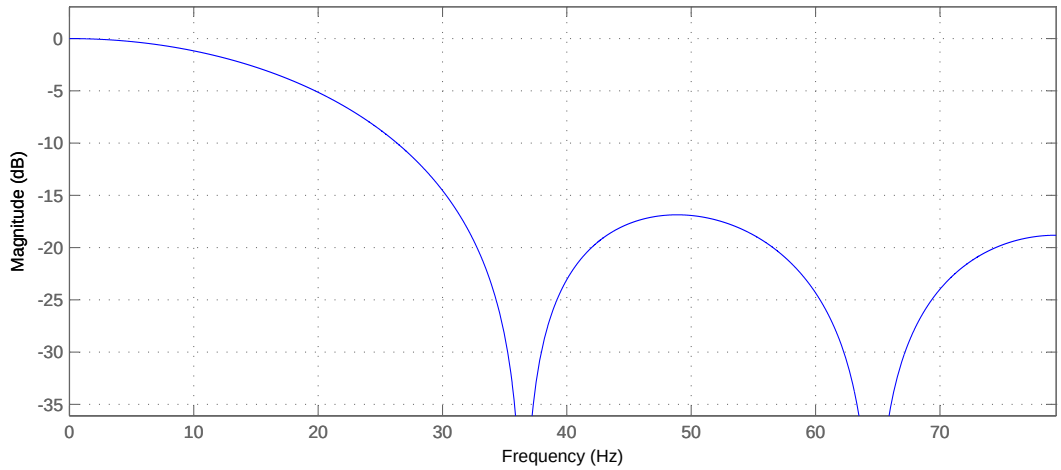



*Figure 3.6:* Magnitude of the transfer function of the fourth order lowpass filter

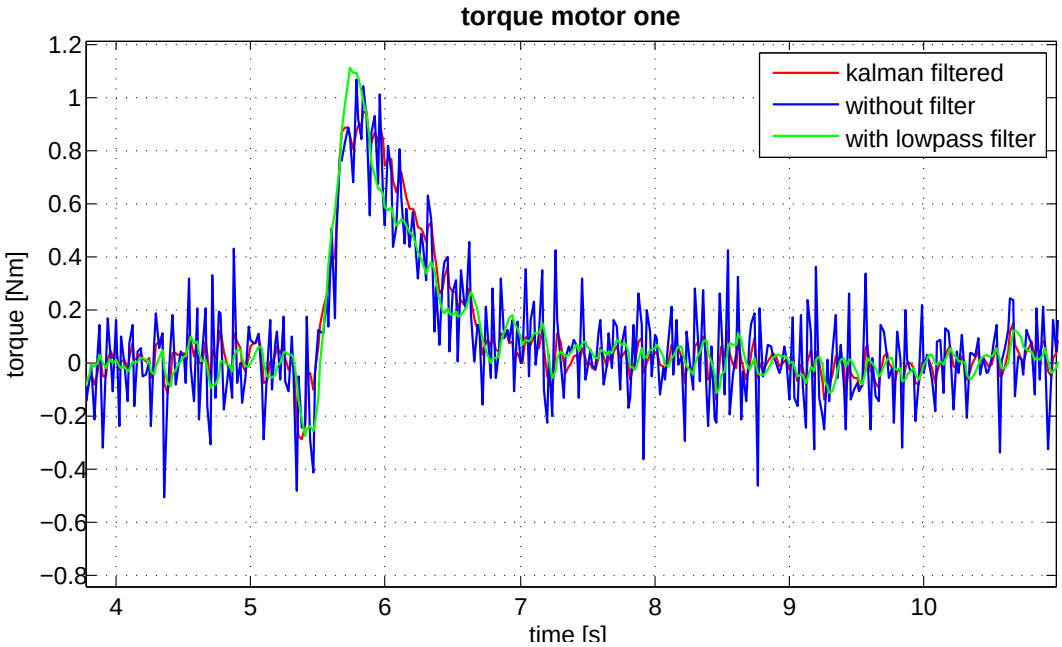



*Figure 3.7:* The raw signal (blue), the low pass filter signal (green) and the Kalman filter signal (red) of the system input

## 3.5 Verification of the Planar Model

A velocity step applied to the real system and to the planar model is shown in figure 3.8. The velocity step in this case is generated using a positive set point for $\dot{\varphi}_x$, which is equal to an actual translation in the direction of the $y$-axis. So actually it is a planar problem. The agreement of the step responses is generally quite good. The small deviations between the measured data and the simulated data are mainly due to noise and simplifications in the model. To perform this step with the prototype, a controller

based on the planar model was implemented. The same controller is used in the model. Therefore, it is not a feed forward verification as usually done. However, that would not be possible on an unstable system. Obviously, this controller is able to stabilize the system and to achieve an acceptable performance. Therefore, the model is a reasonable representation of the system. Further plots can be found in the appendix F.1.

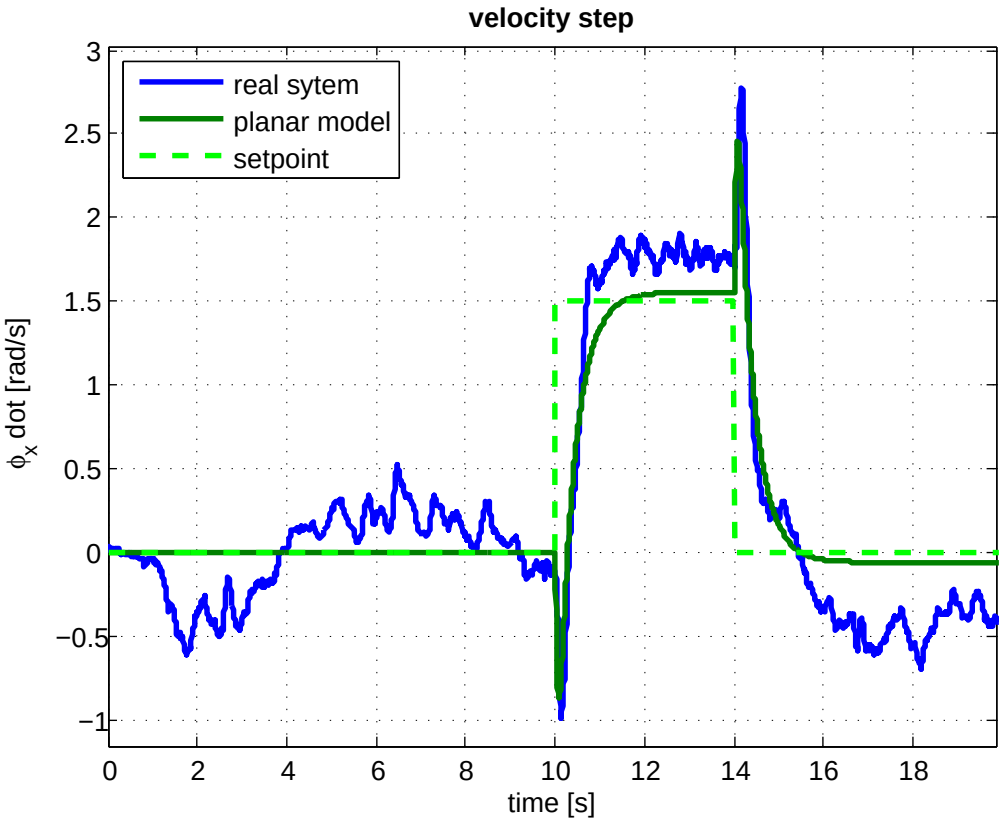



*Figure 3.8:* System and model response on a velocity step from 0 rad/s to 1.5 rad/s for 4 s

## 3.6 Drawbacks on the Planar Model

By splitting the system into three separate planar models, the coupling effects between the models are neglected. It can be shown that when driving into the $y$-direction ($yz$-plane) and then turning about the $z$-axis ($xy$-plane), there is no effect to the $xz$-plane model (fig. 3.9). It is intuitively clear that such a behavior does not represent reality.
Another drawback of the planar system modeling is that a lot of conversions have to be done (see chapter 2.3) to get the real system working. And for example the moments of inertia calculations are simple approximations.

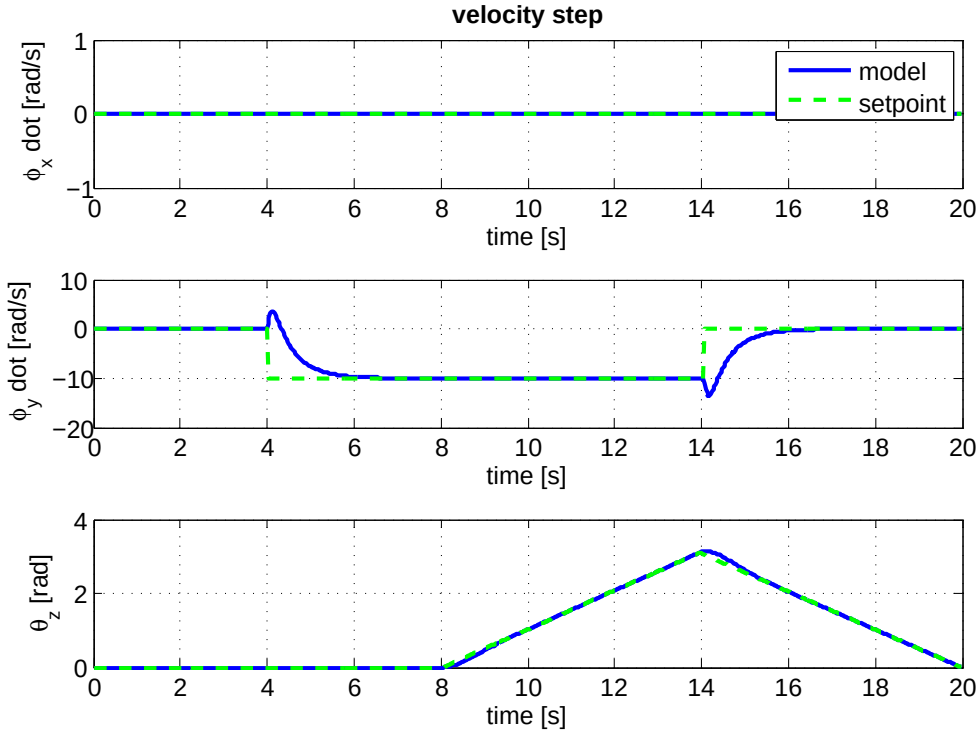



*Figure 3.9:* The response of the model to a velocity step in $x$ direction (center) and a ramp on the rotation around the $z$-axis (bottom)

# 4 3D System Modeling and Control

The representation of the real system of a ballbot as a planar model implies multiple drawbacks as described in section 3.6. The interconnecting effects between the planar models could be added separately to the dynamic equations and the results be verified subsequently. However, the approach described in this chapter takes the real, full three-dimensional geometry and parameters into account. With this model, all coupling effects are considered and no artificial conversions are required.

## 4.1 Model

### 4.1.1 Model Description

#### Geometry

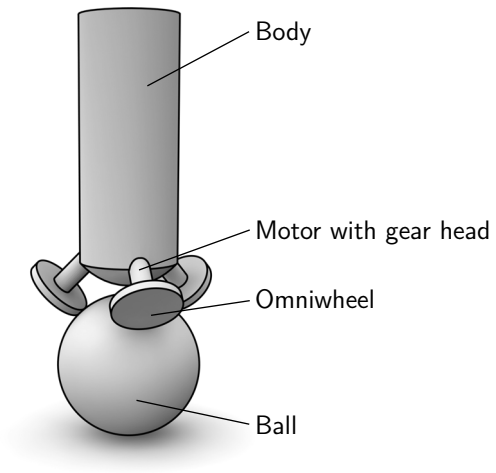



*Figure 4.1:* Geometrical representation of the 3d model

The proposed 3d model as shown in figure 4.1 consists of five solid bodies:

- 1 ball,
- 3 omniwheels,
- 1 body with 3 motors with gear head.

The ball can rotate about any axis parallel to the ground but not about the vertical axis. It is always in contact with the ground in one arbitrary point. Three omniwheels link the upper system to the ball in three points at any time. Each of these wheels is fixed to the

body over a motor with a gear head. The omniwheels can solely rotate about the motor axis in reference to the body coordinate system. The linkage between the wheels and the ball is holonomic and the wheel can only apply forces to the ball in the tangential direction of its rotation in the body reference system.
This system can be described with the following five degrees of freedom (DoF):

- 2 DoF for the position of the ball,
- 3 DoF for the rotation of the body.

All other geometrical coordinates, e.g., the rotation of the omniwheels, can be calculated directly from the given five states. A detailed description of these binding equations is given in section 4.1.4.

#### Input and Output

The inputs to the system are the three torques $T_1$, $T_2$, $T_3$ which are generated by the motors and transformed by the gear heads.
The sensors of the system are chosen in a way that all states can be measured directly. Therefore, each output represents a state separately. Details on the output variables are given in 4.3.2.
This system has several inputs and outputs and can be characterized as a MIMO[1] system.

### 4.1.2 Assumptions

In addition to the geometrical simplification of the system, the same assumptions as in the planar model are made (see 2.1.2).
Since the rotation of the ball about the vertical axis is not modeled, high rotational stiction at the contact point between ball and ground has to be assumed or guaranteed by limiting the torque around the $z$-axis by appropriate motor torque setting.

### 4.1.3 Coordinates

The inertial reference frame is denoted as $I$ (see fig. 4.2). By turning about the inertial $z$-axis $\mathbf{e}_z^I$ with $\vartheta_z$, the reference system denoted as $L$ is described. The reference systems $A'$ is derived from the rotation about the $y$-axis in the $L$-system $\mathbf{e}_y^L$ with $\vartheta_y$. Finally, the body fixed reference system is denoted as $A$ and is derived from the rotation about the $x$-axis in the $A'$-system $\mathbf{e}_x^{A'}$ with $\vartheta_x$.
The representation of the coordinate systems in the Euler form is chosen because the following calculations are well know in this form. Furthermore, the system is supposed to operate in the range of tilt angles of maximum $\pm 20°$, in which singularities do not

[1] Multiple Input, Multiple Output

have to be considered. The sequence of the transformations as Tait-Bryan angles[2] are matched with the convention of the sensor output, making conversions unnecessary.

$$\mathbf{I} \xrightarrow{\vartheta_z} \mathbf{L} \xrightarrow{\vartheta_y} \mathbf{A`} \xrightarrow{\vartheta_x} \mathbf{A}$$

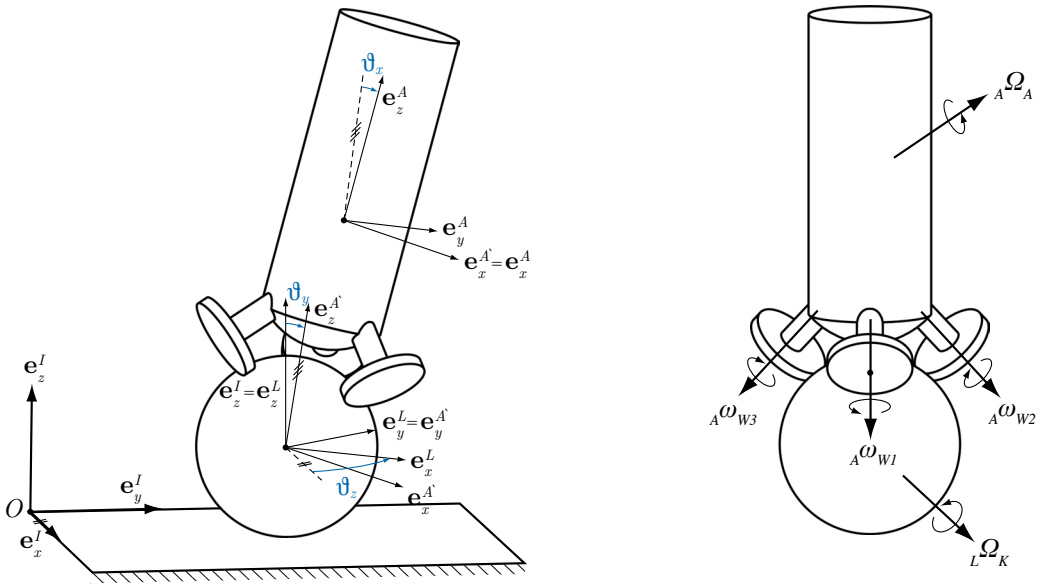



*Figure 4.2:* Coordinate systems (left) and angular velocities (right)

The angular velocity vectors of the bodies are denoted as follows:

- $\vec{\Omega}_K$**: Angular velocity of the ball**
  Parametrization of $\vec{\Omega}_K$ in the reference system $L$:

$$ {}_L\vec{\Omega}_K = \begin{bmatrix} \dot{\varphi}_x \\ \dot{\varphi}_y \\ 0 \end{bmatrix} \tag{4.1}$$

  $\dot{\varphi}_x$, $\dot{\varphi}_y$ are not the variations of any orientation angles of the ball. They are the parametrization of the angular velocity of the ball ${}_L\vec{\Omega}_K$ and are named to match the notation of the planar system.

- $\omega_{Wi}$**: Angular speed of an omniwheel i** relative to the reference system $A$ in direction of the motor axis
  Parametrization of $\omega_{Wi}$:

$$ {}_A\omega_{W1} = \dot{\psi}_1 \qquad {}_A\omega_{W2} = \dot{\psi}_2 \qquad {}_A\omega_{W3} = \dot{\psi}_3 \tag{4.2}$$

[2] specific kind of Euler angles often used in aerospace applications

- $\vec{\Omega}_A$**: Angular velocity of the body**
  In order to express the angular velocity of the body with the variables $\vartheta_x$, $\vartheta_y$ and $\vartheta_z$, the time variation of the Tait-Bryan angles $\dot{\vec{\vartheta}}$ needs to be converted with the Jacobian matrix $\underline{J}$ (see [6] for further explanations).

  Parametrization of $\vec{\Omega}_A$ in the reference system $A$:

$$ {}_A\vec{\Omega}_A = \underline{J} \cdot \dot{\vec{\vartheta}} = \begin{bmatrix} \dot{\vartheta}_x - \sin\vartheta_y \cdot \dot{\vartheta}_z \\ \cos\vartheta_x \cdot \dot{\vartheta}_y + \cos\vartheta_y \cdot \sin\vartheta_x \cdot \dot{\vartheta}_z \\ -\sin\vartheta_x \cdot \dot{\vartheta}_y + \cos\vartheta_x \cdot \cos\vartheta_y \cdot \dot{\vartheta}_z \end{bmatrix} \tag{4.3} $$

### 4.1.4 Binding Equations

The system described has five degrees of freedom, as discussed in 4.1.1. Geometrical dependencies have to be formulated with the aim to express all other geometrical variables as terms of the primary representation of the system.
The primary representation is chosen as $\vartheta_x$, $\vartheta_y$, $\vartheta_z$, $\varphi_x$, $\varphi_y$. This choice renders in a solution more compact than other alternatives (see 4.3.1). It is important to distinguish that while $\vartheta_x$, $\vartheta_y$ and $\vartheta_z$ are spatial orientation angles, $\varphi_x$ and $\varphi_y$ are simply the integral of the parametrization of the angular velocity of the ball ${}_L\vec{\Omega}_K$. $\varphi_x$ and $\varphi_y$ do not represent the orientation of the ball but the rolled angles in the respective direction.

#### Absolute rotation of the omniwheels

As a first step, the absolute rotation of the omniwheels are calculated. The vectors from the intersection of the motor directions $M$ to the center point of the omniwheels $W_1$, $W_2$ and $W_3$ in the body fixed reference system $A$ are denoted as ${}_A\overrightarrow{MW}_1$, ${}_A\overrightarrow{MW}_2$ and ${}_A\overrightarrow{MW}_3$ (fig. 4.3).

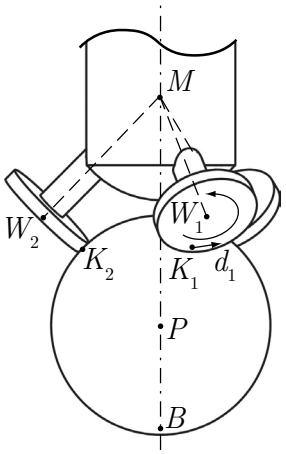



*Figure 4.3:* Geometrical points for the binding equations

The absolute angular speed of the omniwheels about the motor axis in the body reference frame $A$ can now be written as

$$ {}_A\Omega_{Wi} = {}_A\omega_{Wi} + \frac{{}_A\overrightarrow{MW}_i}{||{}_A\overrightarrow{MW}_i||} \cdot {}_A\vec{\Omega}_A \qquad \text{for } i = 1, 2, 3. \tag{4.4} $$

**Dependency on the rotation of the omniwheels**

The vectors from the center of the ball $P$ to the contact point with the omniwheels $K_1$, $K_2$ and $K_3$ in the reference system $A$ are denoted as ${}_A\overrightarrow{PK}_1$, ${}_A\overrightarrow{PK}_2$ and ${}_A\overrightarrow{PK}_3$, the directions of the tangential velocity of the rotation of the omniwheels as unit vectors ${}_A\vec{d}_1$, ${}_A\vec{d}_2$ and ${}_A\vec{d}_3$.
The angular velocity of the ball relative to the body in the reference system $A$ can be calculated as

$$ {}_A\vec{\omega}_K = \underline{R}_{AL} \cdot {}_L\vec{\Omega}_K - {}_A\vec{\Omega}_A \tag{4.5} $$

where $\underline{R}_{AL}$ is the transformation matrix from the reference system $L$ to $A$.

Based on the no slip assumption, it follows: The speed on the surface of the ball in omniwheel direction has to be the same speed as the tangential speed of the omniwheel itself. This can be formulated as a scalar equation for each omniwheel:

$$ ({}_A\vec{\omega}_K \times {}_A\overrightarrow{PK}_i) \cdot {}_A\vec{d}_i = {}_A\omega_{Wi} \cdot r_W \qquad \text{for } i = 1, 2, 3, \tag{4.6} $$

where $r_W$ denotes the radius of the omniwheel. Solving these equations for ${}_A\omega_{W1}$, ${}_A\omega_{W2}$ and ${}_A\omega_{W3}$ results in explicit statements depending on $\vartheta_x$, $\vartheta_y$, $\vartheta_z$, $\dot{\vartheta}_x$, $\dot{\vartheta}_y$, $\dot{\vartheta}_z$, $\dot{\varphi}_x$ and $\dot{\varphi}_y$.

Finally, ${}_A\Omega_{W1}$, ${}_A\Omega_{W2}$ and ${}_A\Omega_{W3}$ can easily be calculated. The solution for ${}_A\Omega_{W2}$ is given as an example:

$$ \begin{aligned} {}_A\Omega_{W2} =& \frac{1}{2\sqrt{2} \cdot r_W} \cdot (r_K \cdot (\cos\vartheta_y + (2\cos\vartheta_x - \sqrt{3}\sin\vartheta_x) \cdot \sin\vartheta_y) \cdot \dot{\varphi}_x \\ & - r_K \cdot (\sqrt{3}\cos\vartheta_x + 2\sin\vartheta_x) \cdot \dot{\varphi}_y + (r_K + r_W) \cdot (-\dot{\vartheta}_x + (\sqrt{3}\cos\vartheta_x + \\ & 2\sin\vartheta_x) \cdot \dot{\vartheta}_y + (\cos\vartheta_y \cdot (-2\cos\vartheta_x + \sqrt{3}\sin\vartheta_x) + \sin\vartheta_y) \cdot \dot{\vartheta}_z)) \end{aligned} $$

All detailed calculations and solutions are shown in E.

Based on exactly the same ideas but calculated and solved in a different order, a solution for the odometry calculation is found. The odometry calculations and solutions are shown in detail in D.

#### Translation of the ball

It is important to know the translational speed of the ballbot for later calculations. Also, integrating the translational speed generates information about the exact location of the ballbot relative to the inertial reference system $I$.
Knowing the vector from the ground $B$ to the center of the ball $P$, ${}_I\overrightarrow{BP}$, the speed vector ${}_I\dot{\vec{r}}_P$ of the center of the ball $P$ can be calculated as

$$ {}_I\dot{\vec{r}}_P = {}_I\Omega_K \times {}_I\overrightarrow{BP}. \tag{4.7} $$

### 4.1.5 Parameters

In the following section all parameters additional to table 2.1 are described.

| Description | Variable | Value |
|---|---|---|
| Inertia of the ball | $\Theta_K$ | 0.0239 kgm$^2$ |
| Inertia of the body | $\Theta_{A,x}$ | 2.026 kgm$^2$ |
| Inertia of the body | $\Theta_{A,y}$ | 2.025 kgm$^2$ |
| Inertia of the body | $\Theta_{A,z}$ | 0.092 kgm$^2$ |
| Inertia of one motor and wheel | $\Theta_W$ | 0.00315 kgm$^2$ |
| Angle of the omniwheel contact point | $\alpha$ | 45° |
| Angles of the omniwheel directions | $\beta_1, \beta_2, \beta_3$ | 0°, 120°, 240° |

*Table 4.1:* Additional parameters of the 3d system

The moments of inertia for the ball and the body, $\underline{\Theta}_K$ and $\underline{\Theta}_A$ respectively, are tensors and are given at the center of the ball. They are chosen in a reference system (see 4.1.3), where the principal moments of inertia become zero.

$$ {}_I\underline{\Theta}_K = {}_L\underline{\Theta}_K = \begin{bmatrix} \Theta_K & 0 & 0 \\ 0 & \Theta_K & 0 \\ 0 & 0 & \Theta_K \end{bmatrix} \tag{4.8} $$

$$ {}_A\underline{\Theta}_A = \begin{bmatrix} {}_A\underline{\Theta}_{A,x} & 0 & 0 \\ 0 & {}_A\underline{\Theta}_{A,y} & 0 \\ 0 & 0 & {}_A\underline{\Theta}_{A,z} \end{bmatrix} \tag{4.9} $$

The moment of inertia for a motor with an attached omniwheel $\Theta_W$, is a scalar and is given solely for the rotation about the motor axis in the body fixed coordinate system. In order to consider the rotation of the motors and omniwheels about all axes in the inertial reference frame, their moment of inertia is simply added to the moment of inertia for the body. This results in a simplified description for the system energies (see 4.2.1). However, gyroscopic effects, especially at high angular speed, are not taken into account with this simplification.

Of course, as described in section 2.3.2, it is necessary to take the gear ratio between motor and omniwheel into consideration. Equation (4.10) describes the relation for the moment of inertia of an omniwheel $\Theta_{OW}$ and the corresponding motor $\Theta_M$.

$$\Theta_W = \Theta_{OW} + i_{gear}^2 \cdot \Theta_M \tag{4.10}$$

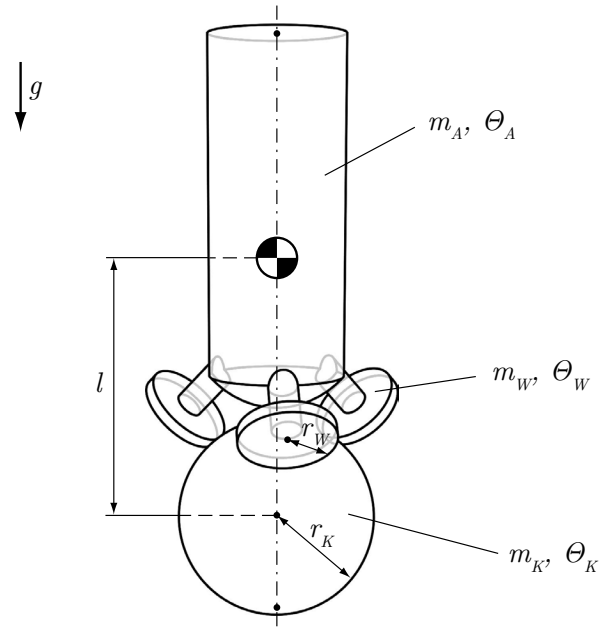



*Figure 4.4:* Geometrical parameters of the 3d systems

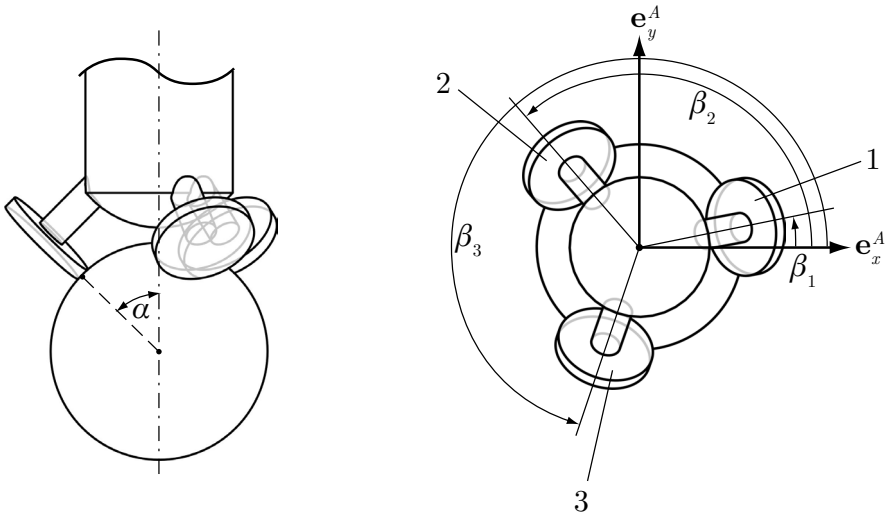



*Figure 4.5:* Angles of the motor configuration

## 4.2 Dynamics

This section describes the derivation of the dynamic equations which are needed for analysis, simulation and the design of a suitable controller.
As with the planar system modeling, the Lagrangian method is used to derive the equations of motion (see 2.2.1). As minimal coordinates $\vec{q}$ the following set of variables is used:

$$\vec{q} = \begin{bmatrix} \vartheta_x & \vartheta_y & \vartheta_z & \varphi_x & \varphi_y \end{bmatrix}^T \tag{4.11}$$

### 4.2.1 Energies

For each body, translational, rotational and potential energies have to be considered. The plane with zero potential energy is chosen to concur the center of the ball.

#### Ball

The kinetic energy of the ball consists of a translational and a rotational part.

$$T_K = \underbrace{\frac{1}{2} \cdot m_K \cdot {}_I\dot{\vec{r}}_P^T \cdot {}_I\dot{\vec{r}}_P}_{Translation} + \underbrace{\frac{1}{2} \cdot {}_L\vec{\Omega}_K^T \cdot {}_L\underline{\Theta}_K \cdot {}_L\vec{\Omega}_K}_{Rotation} \tag{4.12}$$

The ball is only moving in the horizontal directions (see 2.1.2), meaning that its potential energy $V_K$ is zero at any time.

#### Body

As explained in section 4.1.5, in $m_A$ and $\underline{\Theta}_A$ the motors and omniwheels are also considered.
The point of reference is chosen in $P$, which is not the center of mass, in comparison to the calculation of the ball. As a result, a coupling term has to be included.

$$T_A = \underbrace{\frac{1}{2} \cdot m_A \cdot {}_I\dot{\vec{r}}_P^T \cdot {}_I\dot{\vec{r}}_P}_{Translation} + \underbrace{m_A \cdot (\underline{R}_{AI} \cdot {}_I\dot{\vec{r}}_P) \cdot ({}_A\vec{\Omega}_A \times {}_A\vec{r}_{PSA})}_{Coupling} + \underbrace{\frac{1}{2} \cdot {}_A\vec{\Omega}_A^T \cdot {}_A\underline{\Theta}_A \cdot {}_A\vec{\Omega}_A}_{Rotation} \tag{4.13}$$

${}_A\vec{r}_{PSA}$ denotes the vector from the center of the ball $P$ to the center of gravity of the body $SA$ in the reference system $A$.
The potential energy of the body including motors and omniwheels is calculated as

$$V_A = -m_A \cdot \vec{G} \cdot \underline{R}_{IA} \cdot {}_A\vec{r}_{PSA} \qquad \text{with } \vec{G} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{4.14}$$

**Omniwheels**

The additional rotation of the omniwheels and the motors around the respective motor axis is modeled here. As a result of the simplification made in section 4.1.5, only the rotational energies have to be considered.

$$T_{Wi} = \frac{1}{2} \cdot {}_A\Theta_{Wi} \cdot {}_A\Omega_{Wi}^2 \qquad \text{for } i = 1, 2, 3 \tag{4.15}$$

### 4.2.2 Non-Potential Forces

The non-potential forces $f_{NP}$ are of course the motor torques $T_1$, $T_2$ and $T_3$, which are used to stabilize the system. They act directly on ${}_A\omega_{W1}$, ${}_A\omega_{W2}$ and ${}_A\omega_{W3}$ respectively. Since the parameters $\dot{\psi}_1$, $\dot{\psi}_2$ and $\dot{\psi}_3$ are not components of the minimal coordinates $\vec{q}$, a polynomial separation has to be done, such that

$${}_A\omega_{Wi} = \underline{J}_{Ti} \cdot \dot{\vec{q}} \qquad \text{for } i = 1, 2, 3 \tag{4.16}$$

is fulfilled.
Also, it is important to consider the counter torques acting on the body $\vec{T}_{C,1}$, $\vec{T}_{C,2}$ and $\vec{T}_{C,3}$. They act on the opposite direction of $T_1$, $T_2$ and $T_3$ respectively and are formulated as shown in (4.17).

$$\vec{T}_{C,i} = \frac{{}_A\overrightarrow{MW}_i}{||{}_A\overrightarrow{MW}_i||} \cdot (-T_i) \qquad \text{for } i = 1, 2, 3 \tag{4.17}$$

The associated Jacobian matrix is derived in (4.3) and is here denoted as $\underline{J}_{TC}$.
Hence, the non-potential forces $f_{NP}$ can be written as

$$f_{NP} = \underline{J}_{T1}^T \cdot T_1 + \underline{J}_{T2}^T \cdot T_2 + \underline{J}_{T3}^T \cdot T_3 + \underline{J}_{TC}^T \cdot (\vec{T}_{C,1} + \vec{T}_{C,2} + \vec{T}_{C,3}). \tag{4.18}$$

### 4.2.3 Equations of Motion

The equations of motion are now derived by solving the Lagrange equation, analogously to the planar procedure.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{\vec{q}}}\right)^T - \left(\frac{\partial T}{\partial \vec{q}}\right)^T + \left(\frac{\partial V}{\partial \vec{q}}\right)^T - \vec{f}_{NP} = 0 \tag{4.19}$$

with

$$T = T_K + T_A + T_{W1} + T_{W2} + T_{W3}$$
$$V = V_A$$

At this point it is important to note that these calculations could not be done directly in a software package like Mathematica. Instead, the calculations had to be separated into multiple, smaller computations and simplified after each step.

## 4.3 System Analysis

### 4.3.1 Nonlinear System

The equations of motion for the system described in section 4.2.3 are a set of vast, nonlinear, ordinary differential equations. They consist of thousands of terms and cannot be overviewed by any means.
The first 3d models also considered the rotation of the ball about the vertical axis and used different minimal coordinates. As a consequence, the results rendered in a file size over 40 MB when saved as a simple text file. With the model introduced in chapter 4.1.1, the file size is brought down to 0.5 MB and can be simulated in real time.
Because of the size of this complex system, its analysis is limited to a linearized version of the system.

### 4.3.2 Linearized System

For the state space representation (4.20), the state vector $\vec{x}$, the input vector $\vec{u}$ and the output vector $\vec{y}$ are chosen as shown in (4.21)–(4.23). As in section 2.4.2, the process of normalization is left out, because all states, inputs and outputs have the same order of magnitude.

$$\dot{\vec{x}} = A \cdot \vec{x} + B \cdot \vec{u} \qquad \vec{y} = C \cdot \vec{x} + D \cdot \vec{u} \tag{4.20}$$

$$\vec{x} = \begin{bmatrix} \vartheta_x & \dot{\vartheta}_x & \vartheta_y & \dot{\vartheta}_y & \vartheta_z & \dot{\vartheta}_z & \varphi_x & \dot{\varphi}_x & \varphi_y & \dot{\varphi}_y \end{bmatrix}^T \tag{4.21}$$

$$\vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \tag{4.22}$$

$$\vec{y} = \vec{x} \tag{4.23}$$

#### Linearization at zero

As a first step, it makes sense to linearize all states at zero, because the ballbot is controlled to keep the body upright and the position constant. Furthermore, a ballbot can tilt and drive in any direction, so linearization around zero is a reasonable trade-off. Additionally, it is the unstable equilibrium of a ballbot.

$$A_0 = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
37.69 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 37.73 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
-73.02 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & -73.09 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$B_0 = \begin{bmatrix}
0 & 0 & 0 \\
4.02 & -2.01 & -2.01 \\
0 & 0 & 0 \\
0 & 3.485 & -3.485 \\
0 & 0 & 0 \\
-10.76 & -10.76 & -10.76 \\
0 & 0 & 0 \\
-13.48 & 6.738 & 6.738 \\
0 & 0 & 0 \\
0 & -11.68 & -11.68
\end{bmatrix}$$

*Figure 4.6:* Matrices $A_0$ and $B_0$ for the linearization at zero for all states. The subsystem for the planar model in the $yz$-plane is marked blue, for the $xz$-plane red and for the $xy$-plane black.

The matrix $C_0$ is a $10 \times 10$ identity matrix and $D_0$ is a $10 \times 3$ zero matrix. The stability and non-minimum phase behavior is similar to the planar model (see 2.4.1).
Figure 4.6 shows that the 3d system linearized at zero can be split into three independent planar systems. The red and blue marked entries of $A_0$ and $B_0$ are the two identical systems for the $yz$- and $xz$-plane (see 2.1.2) and the black marked entries represent the system for the $xy$-plane. The slight numerical difference in $A_0$ for the $yz$- and $xz$-systems originates in the difference of their respective moment of inertia.
Also, it can be clearly seen by observing the matrix $B_0$ that the motors exert different influences on each system. For example, $u_1$ (from motor 1) has the main effect on the $yz$-system, while $u_2$ and $u_3$ (from motor 2 and 3) have a smaller and opposite input. Likewise, $u_1$ has no influence on the $xz$-system, because its omniwheel stands perpendicular to the ball turning direction of this configuration.

### Linearization at an arbitrary point

To show the influence of the choice of the linearization point, an arbitrary operation point (4.24) is chosen as linearization point.

The parameters $\vartheta_z$, $\varphi_x$ and $\varphi_y$ reflect the orientation and position of the ballbot and are therefore irrelevant for its dynamics. Hence, their linearization point has no influence on the state space matrices.
For all other states, the following linearization point is chosen:

$$\begin{bmatrix} \vartheta_x \\ \dot{\vartheta}_x \\ \vartheta_y \\ \dot{\vartheta}_y \\ \dot{\vartheta}_z \\ \dot{\varphi}_x \\ \dot{\varphi}_y \end{bmatrix} = \begin{bmatrix} 0.32 \text{ rad} \\ -0.5 \text{ rad/s} \\ 0.1 \text{ rad} \\ 0.35 \text{ rad/s} \\ -0.2 \text{ rad/s} \\ 6.3 \text{ rad/s} \\ 2.3 \text{ rad/s} \end{bmatrix} \qquad \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1.1 \text{ Nm} \\ -3.3 \text{ Nm} \\ -2.7 \text{ Nm} \end{bmatrix} \tag{4.24}$$

With $n$ denoting an arbitrary operation point, the state space matrices $C_n$ and $D_n$ are the same as before, $A_n$ and $B_n$ are shown in 4.25 and 4.26 respectively. By comparing the matrices to figure 4.6, it is obvious that $A_n$ and $B_n$ describe different systems and exhibit strong coupling terms.

$$A_n = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 12.85 & -0.5234 & \mathbf{64.1} & 1.164 & 0 & 2.361 & 0 & 0.09295 & 0 & 0.07759 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{-62.57} & 3.008 & 32.44 & -5.953 & 0 & -3.789 & 0 & -0.1806 & 0 & -0.4806 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \mathbf{-38.94} & -8.147 & 4.429 & \mathbf{21.06} & 0 & 6.915 & 0 & 1.066 & 0 & 1.703 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \mathbf{-13.2} & -1.236 & \mathbf{-28.15} & 1.551 & 0 & -2.268 & 0 & 0.03377 & 0 & 0.1916 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \mathbf{45.18} & 0.3581 & -42.06 & -3.098 & 0 & 3.54 & 0 & -0.3492 & 0 & -0.206 \end{bmatrix} \tag{4.25}$$

$$B_n = \begin{bmatrix} 0 & 0 & 0 \\ \mathbf{2.327} & -2.863 & -3.038 \\ 0 & 0 & 0 \\ \mathbf{3.925} & \mathbf{6.862} & \mathbf{1.02} \\ 0 & 0 & 0 \\ -10.17 & \mathbf{-7.92} & -11.51 \\ 0 & 0 & 0 \\ -11.38 & 6.901 & 6.871 \\ 0 & 0 & 0 \\ -2.231 & -13.96 & \mathbf{8.24} \end{bmatrix} \tag{4.26}$$

This shows that a model based on a decoupled system is a very strong limitation and only feasible at an operating point where all states are near zero. A fully operating ballbot with high dynamics can solely be based on a system that considers coupling effects.

## 4.4 Development of a 3D Model Based Controller

As in chapter 2.4.2 above, a model based, linear full state-feedback LQR controller is proposed. Based on the linearized system with all operating points at zero, the following parameters for $Q$ and $R$ are chosen:

$$Q = \text{diag}\,(100, 50, 100, 50, 40, 20, 20, 10, 20, 10)$$
$$R = \text{diag}\,(100, 100, 100)$$

The values for $Q$ are chosen similarly to the parameters described in section 2.4.2. $R$ is chosen at a value so that during normal operation the input of the motors do not reach their limits.
The values in $Q$ for $\vartheta_z$ and $\dot{\vartheta}_z$ can be chosen without influencing the resulting gains for the other dynamics. They are chosen in a way, where at normal operation and disturbances, the ball does not start to slip about the vertical axis.
As an example, the resulting matrix $K$ for the position control is given:

$$K = \begin{bmatrix} 22.34 & 4.63 & -0.00 & -0.00 & -0.36 & -0.29 & 0.36 & 0.53 & -0.00 & -0.00 \\ -11.17 & -2.32 & 19.34 & 4.01 & -0.36 & -0.29 & -0.18 & -0.26 & 0.31 & 0.46 \\ -11.17 & -2.32 & -19.34 & -4.01 & -0.36 & -0.29 & -0.18 & -0.26 & -0.31 & -0.46 \end{bmatrix}$$

## 4.5 Nonlinear Controller Design for the 3D Model

Considering the strong nonlinearities of the system (see 4.3.2), the next step is to develop an appropriate nonlinear controller.

### 4.5.1 Concept

A suitable approach for a nonlinear controller is gain scheduling [8], regarding the size of the dynamic equations of the system (see 4.3.1). With that method, a linear controller is designed at different operating points and the controller gains are interpolated for the current states during operation, i.e. combined to a nonlinear controller.
The nonlinear controller is developed for the velocity controller, because this is the mostly used controller by the different input generating projects. The states causing the systems nonlinearity are $\vartheta_x$, $\dot{\vartheta}_x$, $\vartheta_y$, $\dot{\vartheta}_y$, $\dot{\vartheta}_z$, $\dot{\varphi}_x$, $\dot{\varphi}_y$, $u_1$, $u_2$ and $u_3$ while $\vartheta_x$, $\vartheta_y$, $\dot{\varphi}_x$ and $\dot{\varphi}_y$ having the strongest influence.
In order to design a full nonlinear controller, the 24 entries of the velocity state feedback controller matrix $K$ have to be interpolated in these 10 dimensions. With a discretization of only five linearization points per state, the controller look up table contains $5^{10} \cdot 24 \approx 23.4 \cdot 10^7$ terms (curse of dimensionality).

### 4.5.2 Implementation

Implementing this controller on the given processor [9] means that this number of terms has to be decreased drastically because of memory and computational power limitations. Therefore, as a first approach, the focus was set on the non-linearities resulting from the velocity of the ballbot.

The primary issue with the linear controller in real tests is its incapability to control $\vartheta_z$ at high speeds, meaning over $\sim$2 m/s. At this speed, the ballbot can start oscillating heavily about the inertial $z$-axis which can lead to an instability of the whole system.

This effect can also be observed in the simulation illustrated with the plot on the left side of figure 4.7. The set points are a steep ramp for the velocity in y-direction from 0 to 2 m/s and a flat ramp for $\vartheta_z$. Additionally, little disturbances are given for $\varphi_y$. The right side of figure 4.7 shows that a nonlinear controller, based on linearization points for $\dot{\varphi}_x$ and $\dot{\varphi}_y$, is able to ensure system stability while following the set points.

The same nonlinear control as shown is implemented on the real system and initial tests have proved, that the controller works as expected. However, an elaborate analysis has yet to be done.

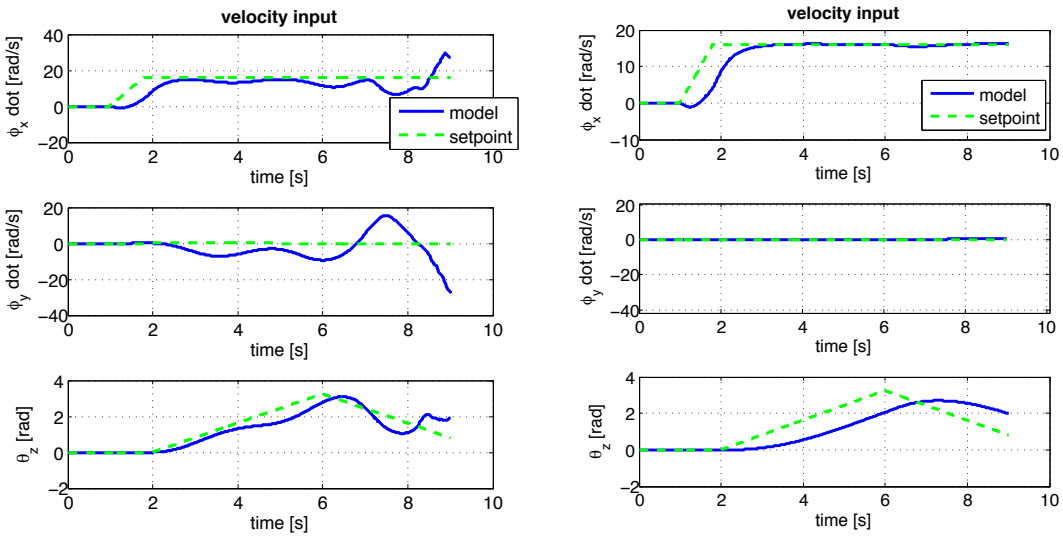



*Figure 4.7:* Performance of the linear (left) and the nonlinear controller (right) for the same set points

# 5 Simulation and Verification of the 3D Model

## 5.1 Implementation in Simulink

Similar to the planar model, the 3d model is also implemented in Simulink. However, the main difference is the way of implementation. Because of the size of the equations of motion, the model is implemented as S-function, which is optimized to make a real time simulation possible. All other functions are, whenever possible, implemented as MATLAB-functions. The advantage of this method is that the MATLAB-functions in the simulation are very similar to the C-functions in the real system. Therefore changes made in the simulation are transferred to the ballbot fast and with a low risk of migration errors.

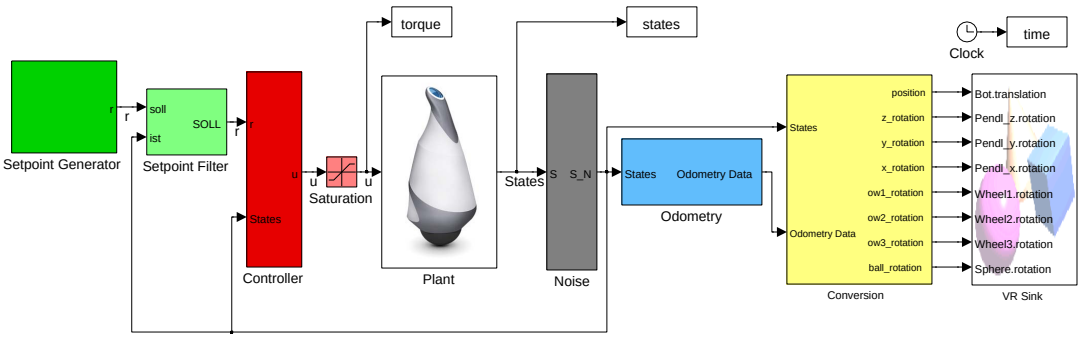



*Figure 5.1:* Block schematic of the 3d simulation environment in Simulink

## 5.2 Verification of the 3D Model

Figure 5.2 shows the responses of the real system and the simulation of the model to a step on the position. The set point step on $\varphi_x$ from 4 rad to -8 rad corresponds to a position difference of 1.5 m. The agreement of the step responses generally meet the expectations. The small deviations between the measured data and the simulated data are explained with the fact that the real system is not able to stand perfectly in place. The controller used for this experiment is based on the 3d model, and is able to achieve a very good performance. Therefore the model is a reasonable representation of the system.

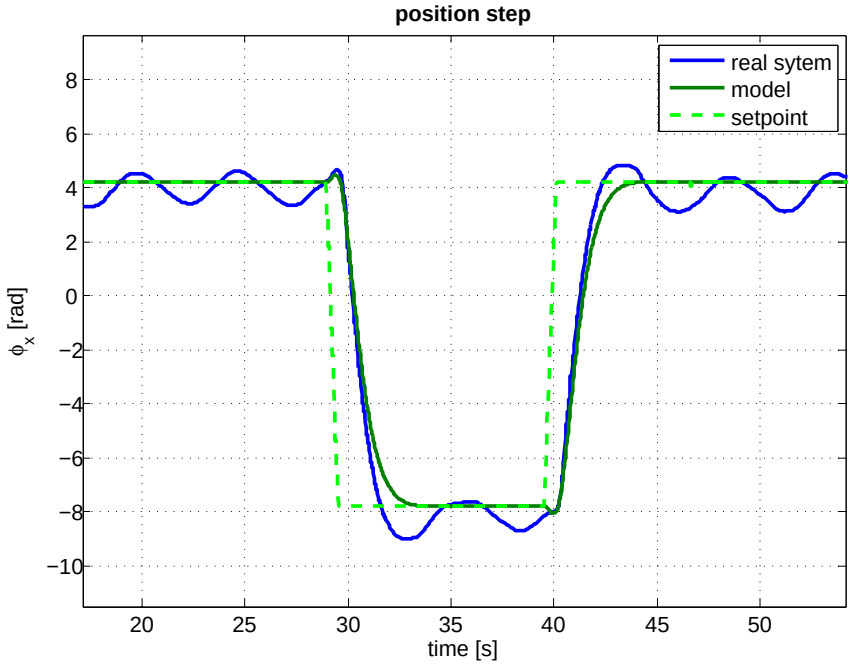



*Figure 5.2:* System and model response on a position step from 4 rad to -8 rad and backwards

Contrary to the planar model, figure 5.3 shows the coupling effects occurring in the 3d model. Driving in $x$ direction and then turning about the $z$-axis does now affect the motion in $y$ direction, although there is no set point.

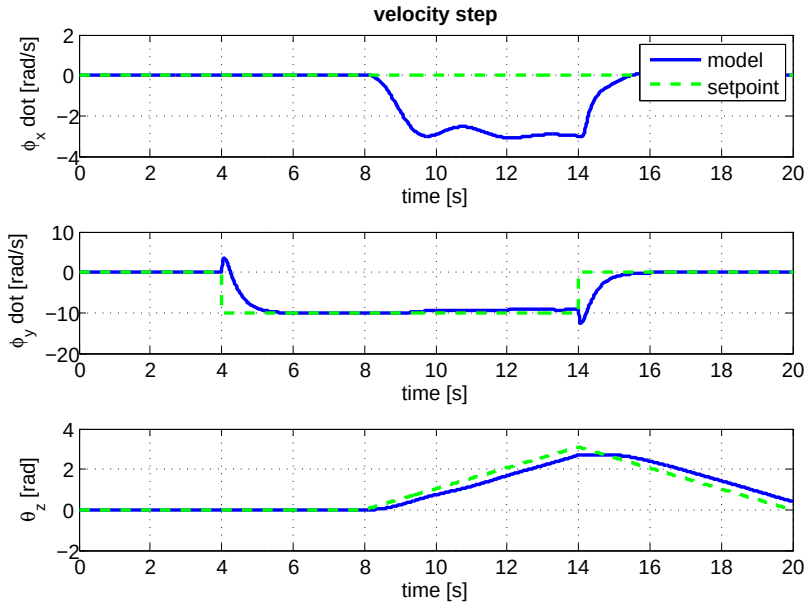



*Figure 5.3:* Response of the model on a velocity step in $x$ direction (center) and a ramp on the rotation around the $z$-axis (bottom).

The setting of more complex set points, generated with a joystick, shows that the agreement of the real system and the simulation is still good (fig. 5.4). Again, the coupling effects are well recognizable. Further plots are shown in the appendix F.2.

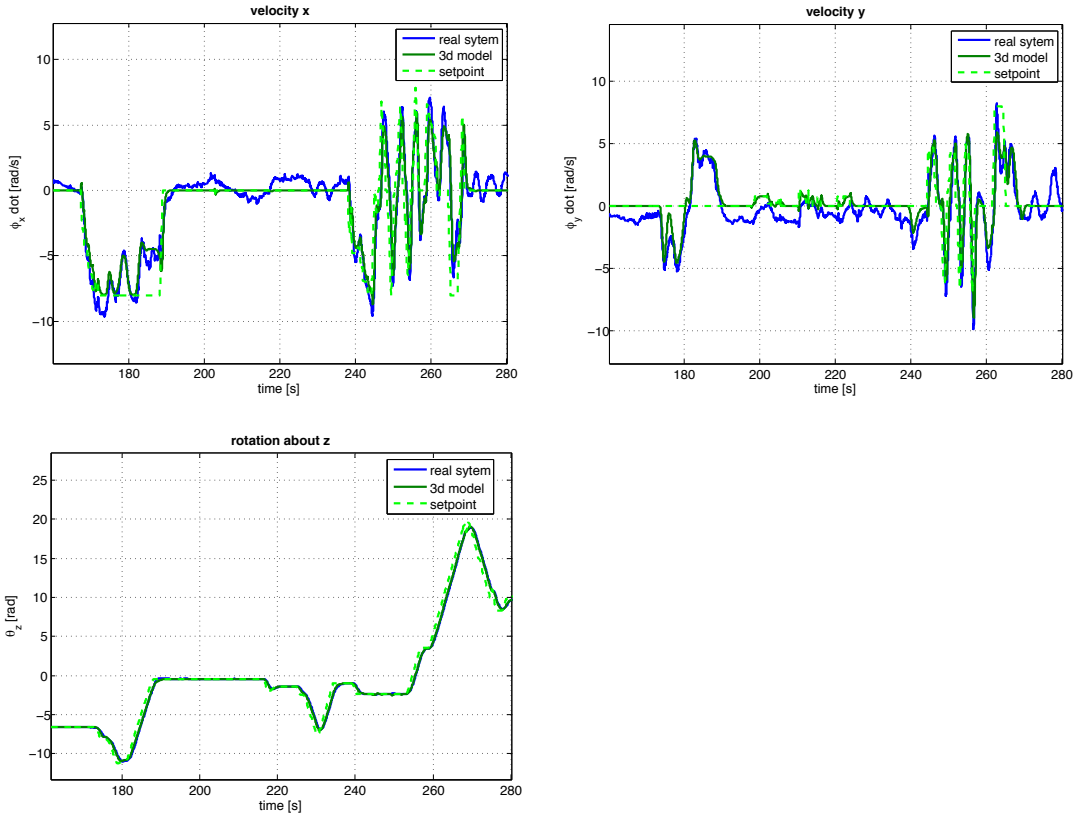



*Figure 5.4:* System and model response on set points generated by a joystick

## 5.3 Comparison

The main difference between the planar and the 3d model that affects the dynamics of the system are the couplings between the different axes. As already discussed in sections 3.6 and 5.2, this difference can be demonstrated easily. In figure 5.5, simulating the 3d model with the planar controller shows that this controller is not able to manage the couplings when driving at faster speed. Therefore the planar controller is not capable of performing complicated sequences of motion.

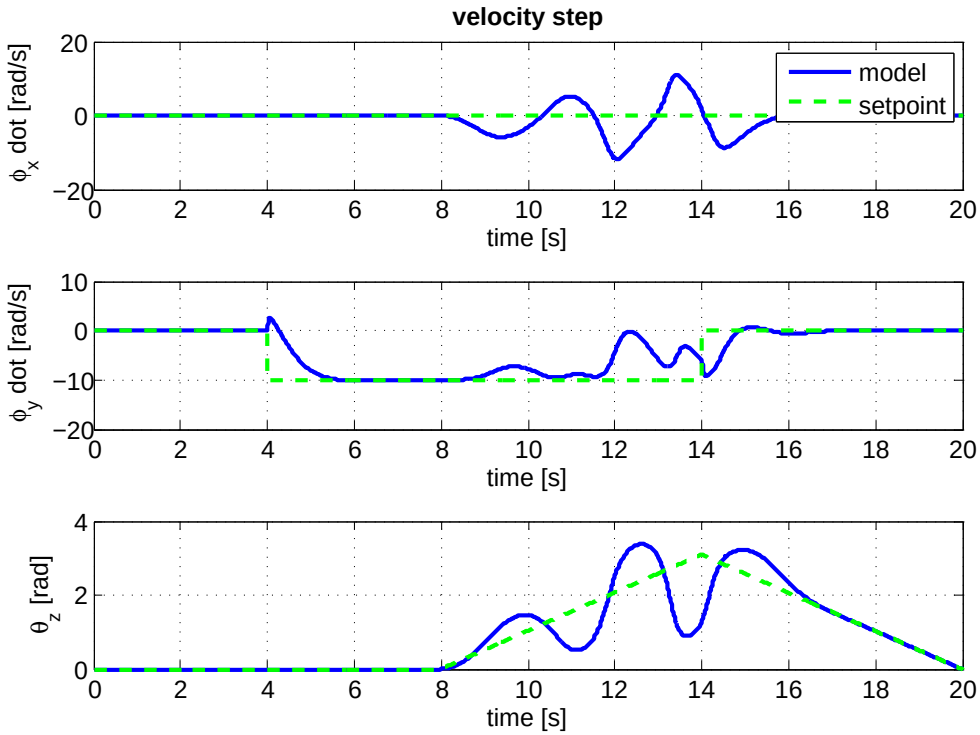



*Figure 5.5:* System response of the 3d model stabilized by the the planar controller

Figure 5.6 shows that the planar controller is not able to follow the complex set points generated by a joystick. When driving at relatively high speed and then turning around the $z$-axis the system starts oscillating.

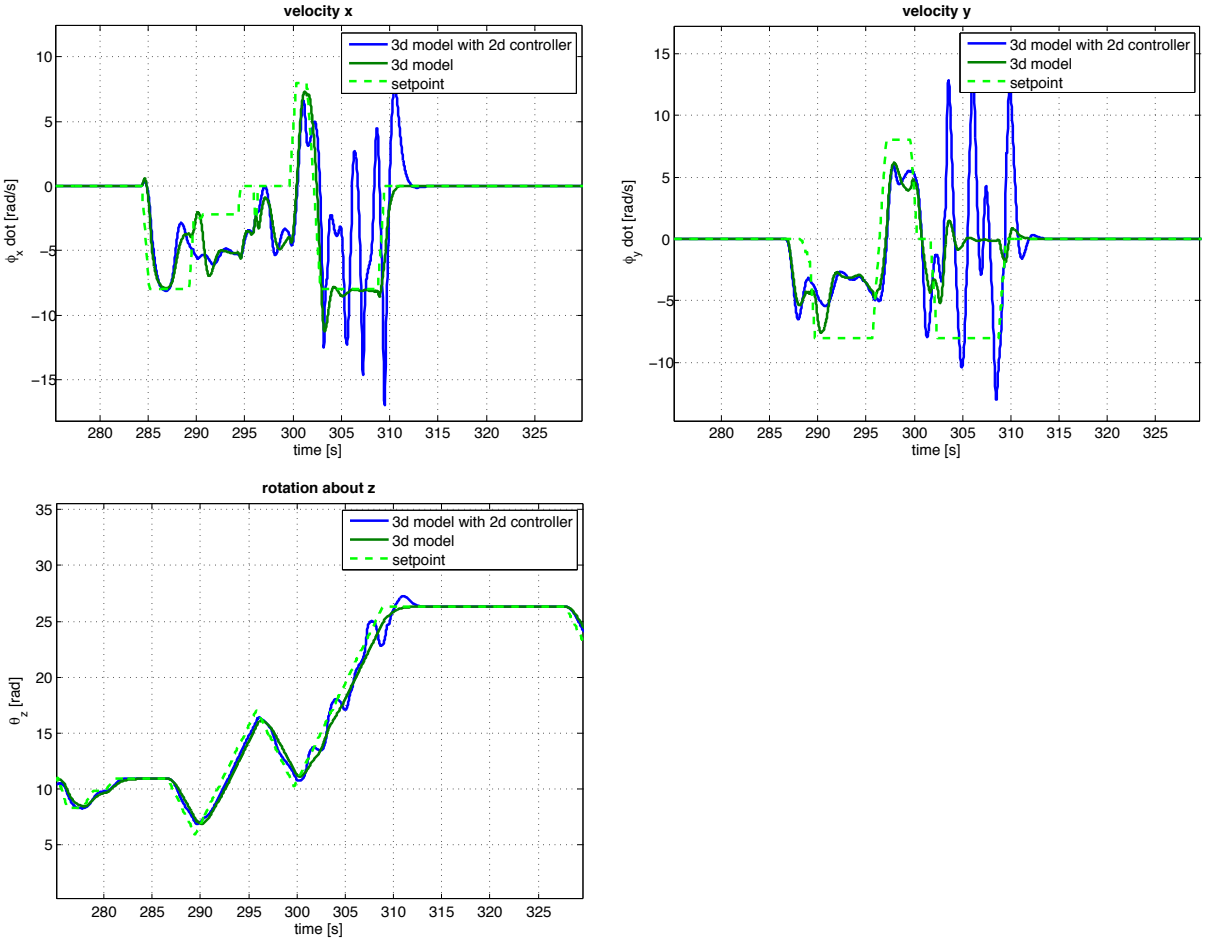



*Figure 5.6:* Comparison of the system responses of the 3d model with the planar and the 3d controller for set points generated by a joystick

# 6 Performance Analysis

During various tests, the performance of the controller was analyzed. Tests were performed on different coverings. Due to the additional damping and contact friction on carpets, the performance is slightly better than on hard floors.
Figure 6.1 shows the motion of the ballbot while balancing on a single point. For this test a special controller is used which contains a low pass filter on the controller output and therefore allows stronger gains. The maximum deviation from the initial point is 1 mm, compared to 20 mm, which is what other ballbots are capable of ([12],[7]).

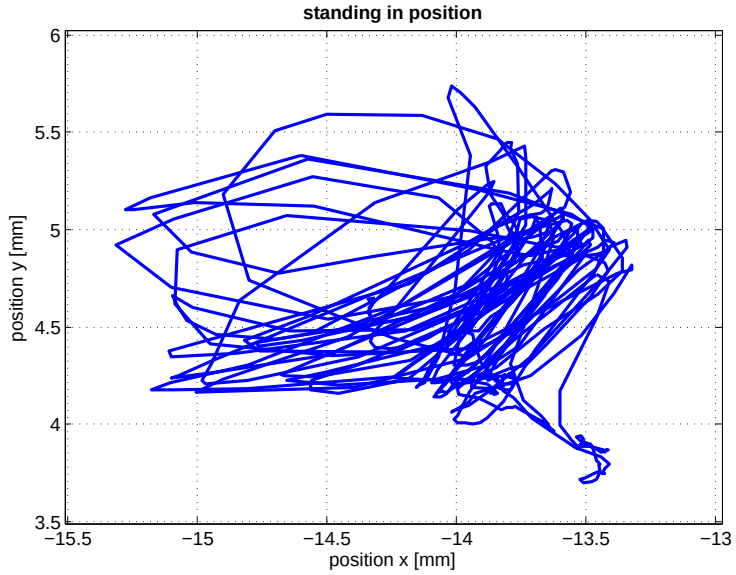



*Figure 6.1:* Deviation of the position for standing on a single point for 20 s

Greater differences between Rezero and other ballbots are shown in the following plots. Figure 6.2 shows the maximum speed that was reached with the prototype. This test was performed with an angular controller which stabilizes only the angle and does not include position or speed (view section 2.4). The plot shows the rotation speed of the ball in rad/s. Converting to a speed in m/s by multiplying with the radius of the ball yields a maximum speed of 3.5 m/s.
The maximum tilt angle reached so far is 17°. The tilt angle corresponds directly to the acceleration of the ballbot. Compared to already existing ballbots ([12],[7]), this angle is much bigger and shows the performance capability of the system and the controller. Figure 6.4 shows a picture of the prototype while tilting 17°. This test was done by generating velocity set points with a joystick.

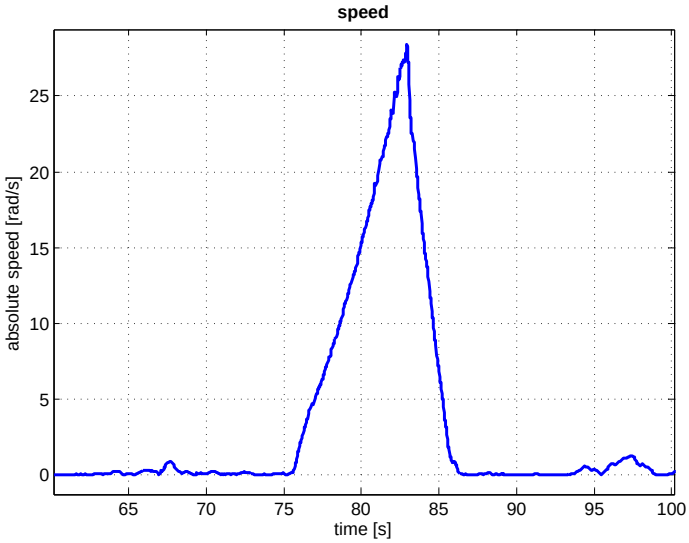



*Figure 6.2:* Speed during speed test reaching its maximum at 3.5 m/s

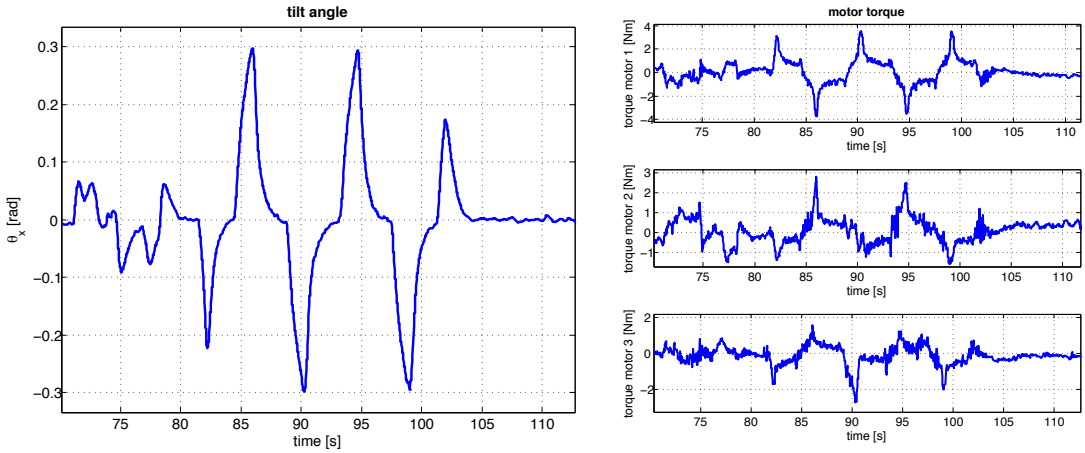



*Figure 6.3:* Maximum tilt angle (left) and corresponding motor torques (right) during system test

The described performance analysis tests were done with the linear controller derived from the 3d model. As the nonlinear controller shows promising potential (4.5), higher performance by using a nonlinear controller can be assumed.

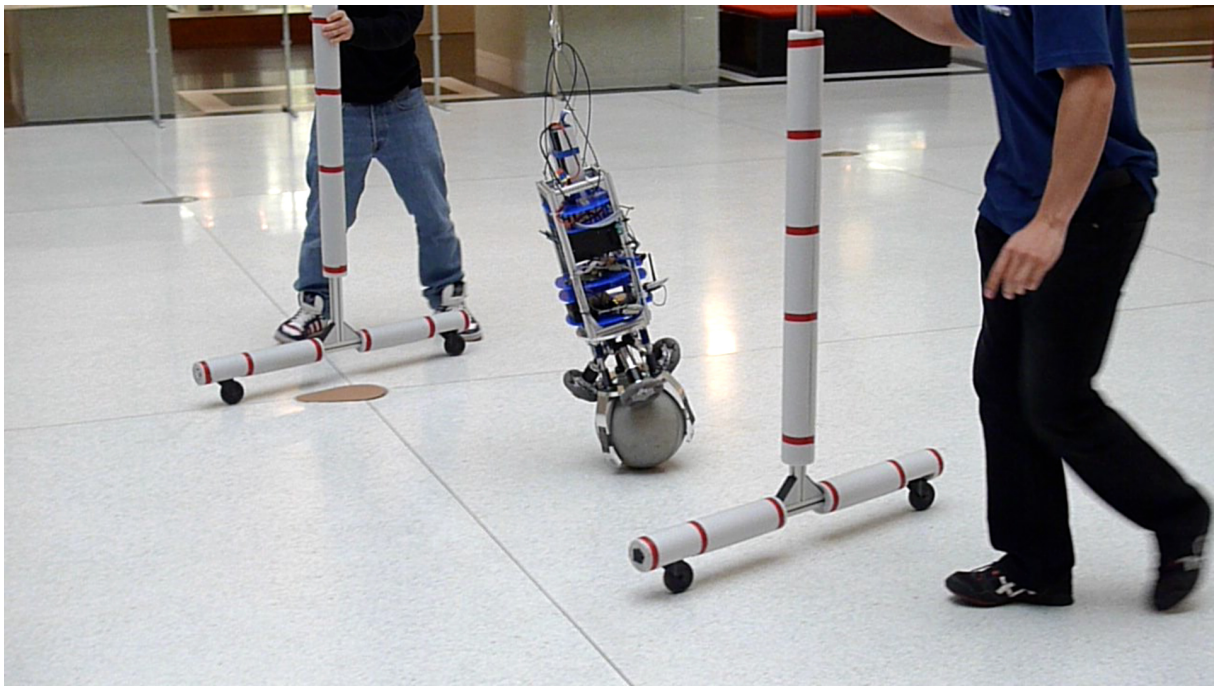


*Figure 6.4:* Performance demonstration, tilt angle of 17°

# 7 Implementation

## 7.1 Set Point Filter

To ensure that the set point jumps are feasible for the controller, the inputs generated by any input device are filtered by a set point filter. This filter makes a ramp from every input and limits its maximum (fig. 7.1). Consequently it has two tuning parameters: the maximum inclination of the ramp and the maximum allowed input. The filter works the way, that if the set point difference is higher than the maximum allowed difference, the allowed difference is added to the last set point in the direction of the demanded set point. This is done every time before the set points are sent to the controller.
The inclination of the ramp is chosen at a value at which the torque of the motors just reach the their limits. The maximum velocity is limited to a value at which the linear controller is sufficient to stabilize the system (cf. section 4.5).

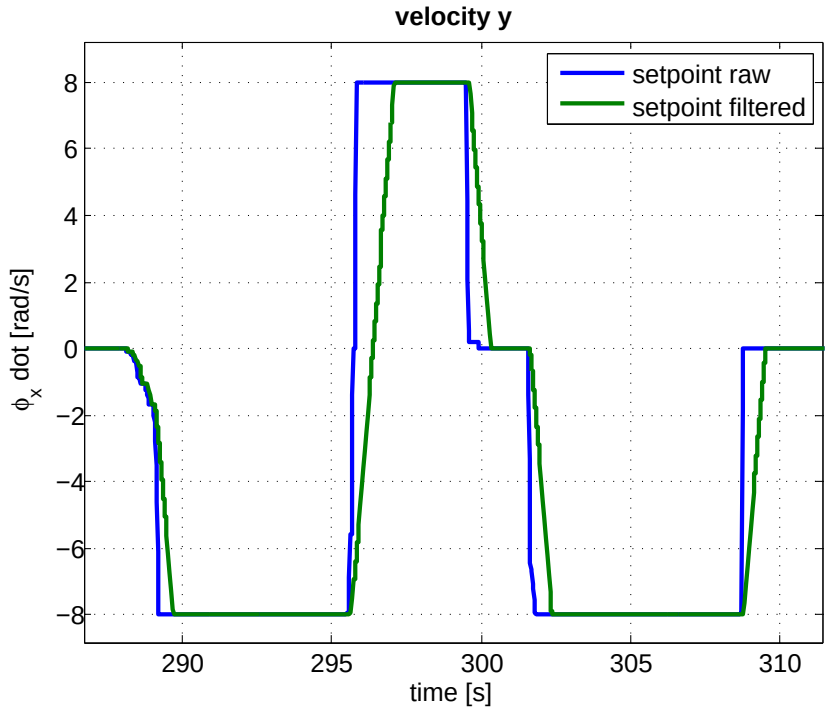



*Figure 7.1:* Raw set point and filtered value of the set point filter

## 7.2 Programming

The controller is implemented on a microcontroller running at 157.8 Hz. This control rate being sufficiently high, discrete control theory can be omitted and the continuous controller designed for the simulation is directly ported. The controller and all calculations are written in C-code.

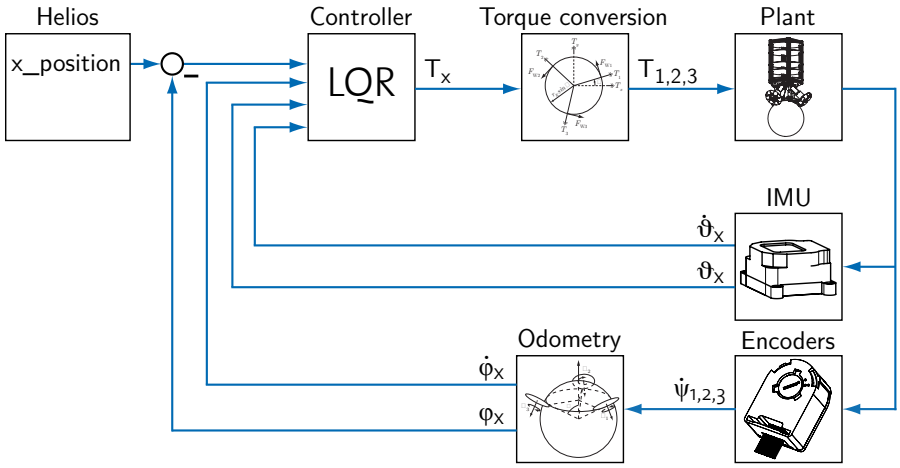



*Figure 7.2:* Structure of the implemented control system

Figure 7.2 shows an overview of the implemented control structure for the planar controller. The states of the plant are measured by the IMU and the encoder of the motors. The latter have to be converted with the odometry in order to match the states from the controller. Accounting for the set point, this dataset is processed by the controller and converted in the torque conversion to deliver the appropriate set point for the motor controllers.
Integration for $\varphi_x$, $\varphi_y$ and $\varphi_z$ is realized as numerical integration in trapezoidal form. Further details of the software and electronics are described in [11] and [9].

## 7.3 Safety System

In system verification and performance tests, the physical ballbot prototype must be operated extensively. In order to avoid system damage and human injury, safety is a very important aspect of running a ballbot.
At system tests with unknown outcome, the ballbot is secured from falling on the ground with a specially designed, rollable safety frame. A emergency button is attached to the safety frame, enabling the operator to turn off the system immediately at any time.
The motors are very powerful and capable of destroying the omniwheels or the ball. By limiting the maximum torques between 1.2 to 5 Nm, according to the situation, a harm

of equipment can be avoided. Also, to protect the gear heads, the motors are shut down as soon as they would accidentally rotate faster than at 8000 rpm.
Since the motors are limited to 5 Nm, the ballbot cannot stabilize tilting angles bigger than $\sim 20^{\circ}$. Running in a situation where the tilting angle exceeds this limit, the system is turned off at $25^{\circ}$ in order to prevent further damage.

# 8 Conclusion

This thesis has followed the goal of developing a controller for a highly agile and maneuverable ballbot. By relying on a simplified planar model first, a real-time simulation with a graphical user interface and three-dimensional visualization is developed. In order to control the real prototype, conversions for torques, moments of inertia and the odometry are calculated. A set of controllers for different inputs such as position, velocity and tilt angle are implemented. Advancing the system model, a full three-dimensional model is proposed and verified. A controller based on the 3d system is developed and shown to be able to stabilize the system for complex trajectories. Various experiments demonstrate the unique capabilities of the ballbot stabilized by the 3d controller. A nonlinear controller is proposed in order to expand the performance of the ballbot prototype.

## 8.1 Achievements

This thesis accomplishes the given goals and meets the requirements in agreement with other projects. At a very early stage, a working simulation has been developed and a controller implemented, stabilizing the real system. By advancing the system model in three dimensions and constant refining of the prototype, a unique controller has been successfully developed. With this controller, Rezero is the first ballbot to show the full potential of moving on a single ball. Rezero was presented in public for several times (see figure 8.1) and drew attention of many automation and control experts.
Having taken a step further, a nonlinear controller was proposed and shown to be more capable of stabilizing the system in high performance situations.

## 8.2 Prospect

The controller is able to push the system to its electrical and mechanical limitations in many respects. However, some desirable performance goals can be solved in a future work. For example, in a remote controlled mode, the speed limit is set to about 2 m/s at the moment, which could be improved by implementing a nonlinear controller.
Also, the system is very sensitive to noise and gear backlash. The motion precision could be improved by implementing a sophisticated filter like a Kalman filter and by replacing hardware elements, such as installing harmonic drives instead of planetary gears.
Finally, very interesting problems like an adaptive control for automatic IMU calibration, variable payloads and slopes could be solved.


*Figure 8.1:* Demonstration at Automatica in Munich

## 8.3 Personal Statement

This work on the modeling and control of our ballbot was a very interesting and educational journey to us. Based on a totally new and self constructed system, it is extremely rewarding to bring a system to life and develop the performance from the first trembling steps to a fast, flexible and robust system.
We think this thesis was a very worthy experience in many respects and we hope that the results will help the further development of Rezero and other ballbots.

# Bibliography

[1] L. Guzzella. *Analysis and Synthesis of Single-Input Single-Output Control Systems*. vdf Hochschulverlag AG, Zürich, Switzerland, 2nd edition, 2009.

[2] L. Guzzella. *Analysis and Synthesis of MIMO Control Systems*. Lecture notes, ETH Zürich, Switzerland, 2010.

[3] T. Holzhüter. *Zustandsregelung*. Germany, 2010.

[4] J. Hüssy. *Signalprocessing for a ballbot*. Bachelor thesis, ETH Zürich, Switzerland, TBP.

[5] J. Fong, S. Uppill. *Design and Build a Ballbot*. Report, The University of Adelaide, Australia, 2009.

[6] S. Leutenegger. *Unmanned Aircraft Design, Modeling and Control*. Lecture notes, ETH Zürich, Switzerland, 2010.

[7] M. Kumagai, T. Ochiai. *Development of a Robot Balancing on a Ball*. Paper, Tohoku Gakuin University, Japan, 2008.

[8] N.G.M. Rademakers. *Control of a Tailless Fighter using Gain-Scheduling*. Traineeship report, Eindhoven University of Technology, Netherlands, 2004.

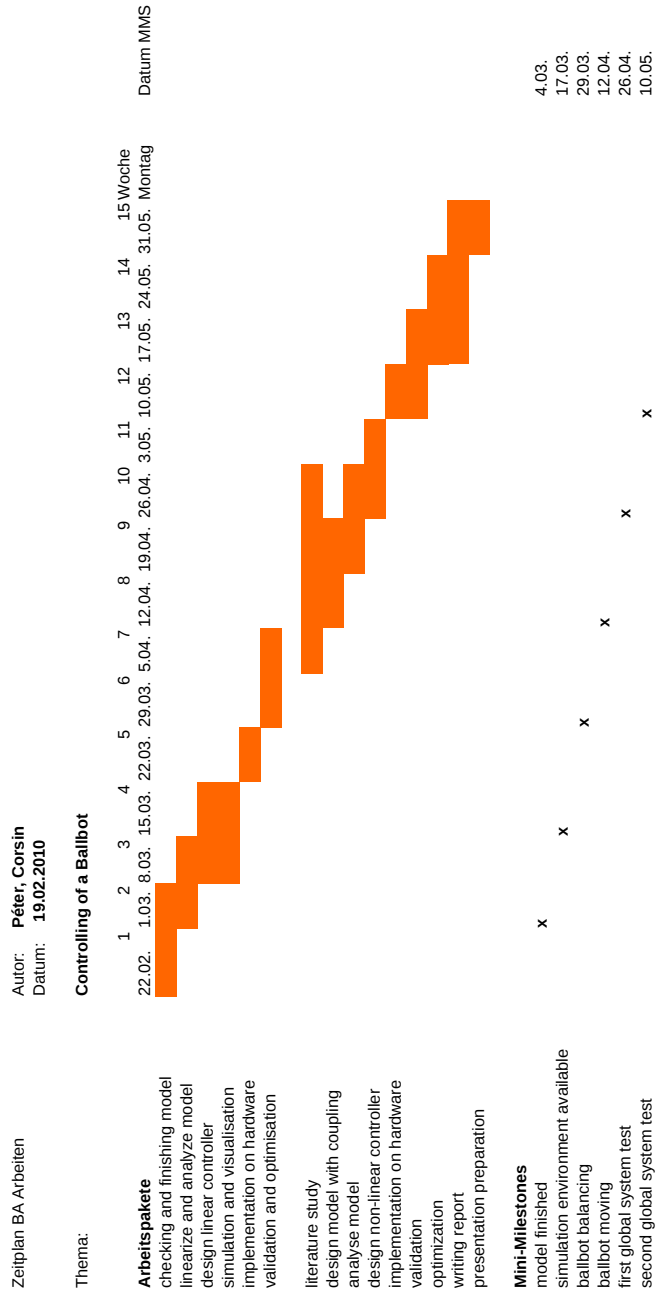
[9] S. Wigert S. Schüller. *Regelung eines auf einem Ball balancierenden Roboters*. Bachelor thesis, Zurich University of Applied Sciences, Switzerland, 2010.

[10] T. B. Lauwers, G. A. Kantor, R. L. Hollis. *A Dynamically Stable Single-Wheeled Mobile Robot with Inverse Mouse-Ball Drive*. Paper, Carnegie Mellon University, USA, 2006.

[11] Team Ballbot. *Rezero – dynamisch stabil auf einer Kugel*. Report, Swiss Federal Institute of Technology, Switzerland, 2010.

[12] U. Nagarajan, A. Mampetta, G. A. Kantor, R. L. Hollis. *State Transition, Balancing, Station Keeping, and Yaw Control for a Dynamically Stable Single Spherical Wheel Mobile Robot*. Paper, Carnegie Mellon University, USA, 2009.

# A Schedule


Zeitplan BA Arbeiten
Autor: Péter, Corsin
Datum: 19.02.2010
Thema:
Controlling of a Ballbot
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 Woche
22.02. 1.03. 8.03. 15.03. 22.03. 29.03. 5.04. 12.04. 19.04. 26.04. 3.05. 10.05. 17.05. 24.05. 31.05. Montag
Datum MMS
Arbeitspakete
checking and finishing model
linearize and analyze model
design linear controller
simulation and visualisation
implementation on hardware
validation and optimisation
literature study
design model with coupling
analyse model
design non-linear controller
implementation on hardware
validation
optimization
writting report
presentation preparation
Mini-Milestones
model finished
simulation environment available
ballbot balancing
ballbot moving
first global system test
second global system test
4.03.
17.03.
29.03.
12.04.
26.04.
10.05.

# B Planar System Modeling

## B.1 Model for the $yz$-/$xz$-Plane

### Binding Equations

One planar system has two degrees of freedom. Therefore, it is possible to write any coordinates as a function of the two minimal coordinates $\varphi_x$ und $\vartheta_x$.
The following equations are valid for the ball:

$$\begin{aligned} x_K &= \varphi_x r_K \\ z_K &= 0 \end{aligned}$$

For the actuating wheel:

$$\begin{aligned} x_W &= x_K + \sin\vartheta_x \cdot (r_K + r_W) \\ &= \varphi_x r_K + \sin\vartheta_x \cdot (r_K + r_W) \\ z_W &= \cos\vartheta_x \cdot (r_K + r_W) \end{aligned}$$

And for the body:

$$\begin{aligned} x_A &= x_K + \sin\vartheta_x \cdot l \\ &= \varphi_x r_K + \sin\vartheta_x \cdot l \\ z_A &= \cos\vartheta_x \cdot l \end{aligned}$$

The binding equation for $\psi_x$ is derived by equating the velocities (*Starrkörper-Geschwindigkeits-Formel*) in the contact point between ball and actuating wheel.

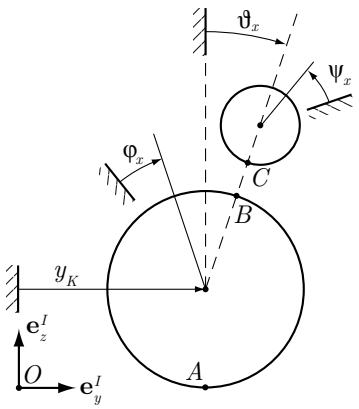


*Figure B.1:* Sketch of the planar model with coordinates and contact points

$$v_A = 0$$

$$v_B = \begin{bmatrix} \dot{x}_K \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\varphi}_x \\ 0 \end{bmatrix} \times \begin{bmatrix} r_K \sin\vartheta_x \\ 0 \\ r_K \cos\vartheta_x \end{bmatrix} = \begin{bmatrix} \dot{x}_K + \dot{\varphi}_x r_K \cos\vartheta_x \\ 0 \\ -r_K \dot{\varphi}_x \sin\vartheta_x \end{bmatrix}$$

$$\begin{aligned} v_C &= \begin{bmatrix} \dot{x}_K \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\vartheta}_x \\ 0 \end{bmatrix} \times \begin{bmatrix} (r_K + r_W)\sin\vartheta_x \\ 0 \\ (r_K + r_W)\cos\vartheta_x \end{bmatrix} + \begin{bmatrix} 0 \\ -\dot{\psi}_x \\ 0 \end{bmatrix} \times \begin{bmatrix} -r_W \sin\vartheta_x \\ 0 \\ -r_W \cos\vartheta_x \end{bmatrix} \\ &= \begin{bmatrix} \dot{x}_K + \dot{\vartheta}_x(r_K + r_W)\cos\vartheta_x + r_W\dot{\psi}_x \cos\vartheta_x \\ 0 \\ -\dot{\vartheta}_x(r_K + r_W)\sin\vartheta_x - r_W\dot{\psi}_x \sin\vartheta_x \end{bmatrix} \end{aligned}$$

To comply the no slip condition, the equation $v_B = v_C$ has to be true. It is then possible to equate the first and third term of $v_B$ and $v_C$. Solving these equations by $\dot{\psi}_x$ yields:

$$\dot{\psi}_x = \frac{r_K}{r_W}(\dot{\varphi}_x - \dot{\vartheta}_x) - \dot{\vartheta}_x \tag{B.1}$$

**Approach**

For the derivation of the equations of motion the Lagrangian method was selected as the common approach for that kind of problems. The following steps are needed:

- express kinetic (T) and potential (V) energy of all rigid bodies as a function of the minimal coordinates,
- write non-potential forces as a function of the minimal coordinates,
- solve the Lagrange equation for the second derivative of the minimal coordinates.

The calculations are made for the $yz$-plane model, which is identical to the $xz$-plane model, and for the $xy$-plane model as well.

**Energies**

Ball:

$$T_K = \frac{1}{2} m_K (r_K \dot{\varphi}_x)^2 + \frac{1}{2}\Theta_K \dot{\varphi}_x^2$$

$$V_K = 0$$

Actuating wheel:

$$
\begin{aligned}
T_W &= \frac{1}{2} m_W \dot{\vec{x}}_W \cdot \dot{\vec{x}}_W^\top + \frac{1}{2}\Theta_W \dot{\psi}_x^2, \quad \text{wobei } \vec{x}_W = \begin{bmatrix} r_K\varphi_x + (r_K + r_W)\sin\vartheta_x \\ 0 \\ (r_K + r_W)\cos\vartheta_x \end{bmatrix} \\
&= \frac{1}{2} m_W \begin{bmatrix} (r_K\dot{\varphi}_x) + (r_K + r_W)\cos\vartheta_x\dot{\vartheta}_x \\ 0 \\ -(r_K + r_W)\sin\vartheta_x\dot{\vartheta}_x \end{bmatrix}^2 + \frac{1}{2}\Theta_W \left( \frac{r_K}{r_W}(\dot{\varphi}_x - \dot{\vartheta}_x) - \dot{\vartheta}_x \right)^2 \\
&= \frac{1}{2} m_W \left( (r_K\dot{\varphi}_x)^2 + 2\cdot(r_K + r_W)\cos\vartheta_x\dot{\vartheta}_x(r_K\dot{\varphi}_x) + (r_K + r_W)^2\dot{\vartheta}_x^2 \right) \\
&\quad + \frac{1}{2}\Theta_W \left( \frac{r_K}{r_W}(\dot{\varphi}_x - \dot{\vartheta}_x) - \dot{\vartheta}_x \right)^2 \\
V_W &= m_W \cdot g \cdot (r_K + r_W) \cdot \cos\vartheta_x
\end{aligned}
$$

Body:

$$
\begin{aligned}
T_A &= \frac{1}{2} m_A \dot{\vec{x}}_A \cdot \dot{\vec{x}}_A^\top + \frac{1}{2}\Theta_A \dot{\vartheta}_x^2, \quad \text{wobei } \vec{x}_A = \begin{bmatrix} r_K\varphi_x + l\sin\vartheta_x \\ 0 \\ l\cos\vartheta_x \end{bmatrix} \\
&= \frac{1}{2} m_A \begin{bmatrix} (r_K\dot{\varphi}_x) + l\cos\vartheta_x\dot{\vartheta}_x \\ 0 \\ -l\sin\vartheta_x\dot{\vartheta}_x \end{bmatrix}^2 + \frac{1}{2}\Theta_A\dot{\vartheta}_x^2 \\
&= \frac{1}{2} m_A \left( (r_K\dot{\varphi}_x)^2 + 2\cdot l\cos\vartheta_x\dot{\vartheta}_x(r_K\dot{\varphi}_x) + l^2\dot{\vartheta}_x^2 \right) + \frac{1}{2}\Theta_A\dot{\vartheta}_x^2 \\
V_A &= m_A \cdot g \cdot l \cdot \cos\vartheta_x
\end{aligned}
$$

**Non-potential forces**

The only non-potential force is the torque from the actuating wheel:

$$
\vec{M} = \begin{bmatrix} 0 \\ -M \\ 0 \end{bmatrix}, \quad \vec{\Omega} = \begin{bmatrix} 0 \\ -\frac{r_K}{r_W}(\dot{\varphi}_x - \dot{\vartheta}_x) + \dot{\vartheta}_x \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 \\ -\frac{r_K}{r_W} & \left(\frac{r_K}{r_W} + 1\right) \\ 0 & 0 \end{bmatrix}}_{\vec{J}} \cdot \dot{\vec{q}}
$$

$$
\vec{f}_{NP,1} = \vec{J}^T\vec{M} = \begin{bmatrix} \frac{r_K}{r_W}M \\ -\left(1 + \frac{r_K}{r_W}\right)M \end{bmatrix}
$$

The counter torque acting on the coordinate $\vartheta_x$ of the body is expressed as:

$$\vec{M} = \begin{bmatrix} 0 \\ M \\ 0 \end{bmatrix}, \quad \vec{\Omega} = \begin{bmatrix} 0 \\ \dot{\vartheta}_x \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\vec{J}} \cdot \dot{\vec{q}}, \quad \vec{f}_{NP,2} = \vec{J}^T \vec{M} = \begin{bmatrix} 0 \\ M \end{bmatrix}$$

The total non-potential force is the sum of those two:

$$\vec{f}_{NP} = \vec{f}_{NP,1} + \vec{f}_{NP,2} = \begin{bmatrix} \frac{r_K}{r_W} M \\ -\left(1 + \frac{r_K}{r_W}\right) M + M \end{bmatrix}$$

**Equations of motion**

Now the equations of motion are derived by solving the Lagrangian equation for $\ddot{\vec{q}}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{\vec{q}}}\right)^T - \left(\frac{\partial T}{\partial \vec{q}}\right)^T + \left(\frac{\partial V}{\partial \vec{q}}\right)^T - \vec{f}_{NP} = 0 \tag{B.2}$$

where $T$ and $V$ stand for the total kinetic and potential energy respectively.

$$T = T_K + T_W + T_A$$
$$V = V_K + V_W + V_A$$

The solution of the Lagrange equation in matrix form is described as:

$$M_x(\vec{q}, \dot{\vec{q}})\ddot{\vec{q}} + C_x(\vec{q}, \dot{\vec{q}}) + G_x(\vec{q}) = f_{NP} \tag{B.3}$$

where $M_x, C_x, G_x$ are calculated as:

$$M_x = \begin{bmatrix} m_{tot} r_K^2 + \Theta_K + \left(\frac{r_K}{r_W}\right)^2 \Theta_W & -\frac{r_K}{r_W^2}(r_K + r_W)\Theta_W + \gamma r_K \cos\vartheta_x \\ -\frac{r_K}{r_W^2}(r_K + r_W)\Theta_W + \gamma r_K \cos\vartheta_x & \frac{(r_K + r_W)^2}{r_W^2}\Theta_W + \Theta_A + m_A l^2 + m_W (r_K + r_W)^2 \end{bmatrix}$$
$$C_x = \begin{bmatrix} -r_K \gamma \sin\vartheta_x \dot{\vartheta}_x^2 \\ 0 \end{bmatrix}$$
$$G_x = \begin{bmatrix} 0 \\ -g \sin\vartheta_x \gamma \end{bmatrix}$$

where

$$m_{tot} = m_K + m_A + m_W$$
$$\gamma = l m_A + (r_K + r_W) m_W$$

## B.2 Model for the $xy$-Plane

For the $xy$-plane model, the same approach is used. The complete derivation is given as follows.

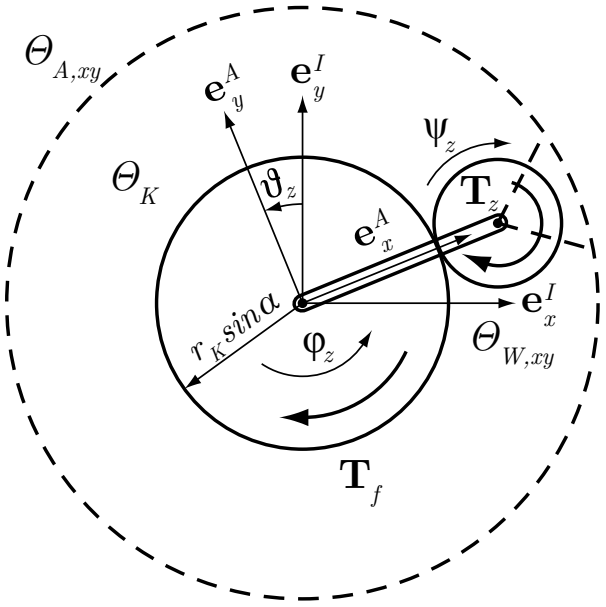



*Figure B.2:* Planar model for the $xy$-plane

### Minimal coordinates

The same minimal coordinates are used (see figure B.2).

$$\vec{q} = \begin{bmatrix} \varphi_z \\ \vartheta_z \end{bmatrix}$$

### Energies

Ball:

$$T_K = \frac{1}{2}\Theta_K \dot{\varphi}_z^2$$

Actuating wheel (Note: $\Theta_{W,xy}$ contains the inertias of motors and wheels around $\psi_z$).

$$T_W = \frac{1}{2}\Theta_{W,xy} \dot{\psi}_z^2$$

Body (Note: $\Theta_{A,xy}$ contains the inertias of motors and wheels around $\vartheta_z$).

$$T_A = \frac{1}{2}\Theta_{A,xy} \dot{\vartheta}_z^2$$

**Binding equation**

$$\dot{\psi}_z = \frac{r_K}{r_W} \cdot \sin\alpha \cdot (\dot{\varphi}_z - \dot{\vartheta}_z) \tag{B.4}$$

**Non-potential forces**

From the actuators:

$$T_1 = T_z$$
$$J_1 = \begin{bmatrix} \frac{r_K}{r_W} \cdot \sin\alpha \\ -\frac{r_K}{r_W} \cdot \sin\alpha \end{bmatrix}$$

Counter torque:

$$T_2 = -T_z$$
$$J_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Friction between ball and ground:

$$T_3 = -T_f$$
$$J_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$f_{NP}$:

$$f_{NP} = J_1 \cdot T_1 + J_2 \cdot T_2 + J_3 \cdot T_3 = \begin{bmatrix} -T_f + \frac{r_K}{r_W} \cdot \sin\alpha \cdot T_z \\ -\frac{r_K}{r_W} \cdot \sin\alpha \cdot T_z + T_z \end{bmatrix} \tag{B.5}$$

**Equation of motion**

Total kinetic energy:

$$T = T_K + T_W + T_A$$

Total potential energy:

$$V = 0$$

Solving the Lagrange equation for $\ddot{\varphi}$ and $\ddot{\vartheta}$ yields:

$$\ddot{\varphi} = -\frac{(r_W^2\Theta_{A,xy} + r_K^2\Theta_{W,xy}\sin^2\alpha) \cdot T_f + r_K r_W \Theta_{A,xy} \sin\alpha \cdot T_z}{r_W^2\Theta_{A,xy}\Theta_K + r_K^2(\Theta_{A,xy} + \Theta_K)\Theta_{W,xy}\sin^2\alpha} \tag{B.6}$$

$$\ddot{\vartheta} = -\frac{r_K \sin\alpha (r_K \Theta_{W,xy} \sin\alpha \cdot T_f + r_W \Theta_K \cdot T_z)}{r_W^2 \Theta_{A,xy} \Theta_K + r_K^2 (\Theta_{A,xy} + \Theta_K) \Theta_{W,xy} \sin^2\alpha} \tag{B.7}$$

### Friction

Condition for stiction:

$$\ddot{\varphi} = 0$$

Equation of motion solved for $T_f$:

$$T_f = \frac{r_K r_W \Theta_{A,xy} \sin\alpha \cdot T_z}{r_W^2 \Theta_{A,xy} + r_K^2 \Theta_{W,xy} \sin^2\alpha}$$

## B.3 Forces

### Free-body diagram

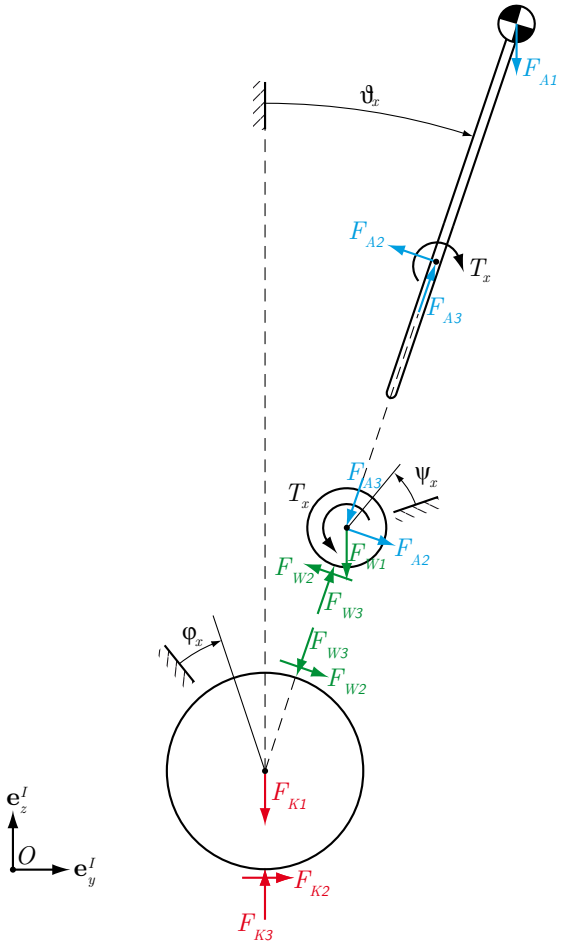



*Figure B.3:* Free-body diagram of the planar model

**Conservation of momentum**

For the Ball the following equations are valid.
Conservation of momentum in $x$-direction:

$$m_K \ddot{x}_K = F_{K2} + F_{W2} \cos \vartheta_x - F_{W3} \sin \vartheta_x \tag{B.8}$$

Conservation of momentum in $z$-direction:

$$m_K \ddot{z}_K = F_{K3} - F_{K1} - F_{W2} \sin \vartheta_x - F_{W3} \cos \vartheta_x \tag{B.9}$$

Conservation of angular momentum:

$$\Theta_K \ddot{\varphi}_x = F_{W2} r_K - F_{K2} r_K \tag{B.10}$$

For the actuating wheel the following equations are valid.
Conservation of momentum in $x$-direction:

$$m_W \ddot{x}_W = -F_{W2} \cos \vartheta_x + F_{W3} \sin \vartheta_x + F_{A2} \cos \vartheta_x - F_{A3} \sin \vartheta_x \tag{B.11}$$

Conservation of momentum in $z$-direction:

$$m_W \ddot{z}_W = -F_{W1} + F_{W2} \sin \vartheta_x + F_{W3} \cos \vartheta_x - F_{A2} \sin \vartheta_x - F_{A3} \cos \vartheta_x \tag{B.12}$$

Conservation of angular momentum:

$$\Theta_W \ddot{\psi}_x = -F_{W2} r_W + M_x \tag{B.13}$$

For the body the following equations are valid.
Conservation of momentum in $x$-direction:

$$m_A \ddot{x}_A = -F_{A2} \cos \vartheta_x + F_{A3} \sin \vartheta_x \tag{B.14}$$

Conservation of momentum in $z$-direction:

$$m_A \ddot{z}_A = -F_{A1} + F_{A2} \sin \vartheta_x + F_{A3} \cos \vartheta_x \tag{B.15}$$

Conservation of angular momentum:

$$\Theta_A \ddot{\vartheta}_x = F_{A2}(l - r_K - r_W) + M_x \tag{B.16}$$

### Potential forces

The following potential forces act in the system:

$$F_{K1} = m_K \cdot g$$
$$F_{W1} = m_W \cdot g$$
$$F_{A1} = m_A \cdot g$$

### Binding equations

The binding equations are derived in section B.1

### Binding forces

Solving the equations (B.14) and (B.15) for the forces $F_{A2}$ and $F_{A3}$ yields:

$$F_{A2} = m_A \cdot \Big(g \sin\vartheta_x - r_K \ddot{\varphi}_x \cos\vartheta_x - \ddot{\vartheta}_x l\Big)$$
$$F_{A3} = m_A \cdot \Big(g \cos\vartheta_x + r_K \ddot{\varphi}_x \sin\vartheta_x - \dot{\vartheta}_x^2 l\Big)$$

With this results and the equations (B.11) and (B.12) the forces $F_{W2}$ and $F_{W3}$ can be calculated:

$$F_{W2} = (m_A + m_W) \cdot \Big(g \sin\vartheta_x - r_K \ddot{\varphi}_x \cos\vartheta_x\Big) - \gamma \ddot{\vartheta}_x$$
$$F_{W3} = (m_A + m_W) \cdot \Big(g \cos\vartheta_x + r_K \ddot{\varphi}_x \sin\vartheta_x\Big) - \gamma \dot{\vartheta}_x^2$$

Finally, it is possible to solve the equations (B.8) and (B.9) for $F_{K2}$ and $F_{K3}$:

$$F_{K2} = \gamma\Big(\ddot{\vartheta}_x \cos\vartheta_x - \dot{\vartheta}_x^2 \sin\vartheta_x\Big) + r_K m_{tot} \ddot{\varphi}_x$$
$$F_{K3} = -\gamma\Big(\ddot{\vartheta}_x \sin\vartheta_x + \dot{\vartheta}_x^2 \cos\vartheta_x\Big) + g m_{tot}$$

Where

$$m_{tot} = m_K + m_A + m_W$$
$$\gamma = l m_A + (r_K + r_W) m_W$$

# C Conversions

## C.1 Torque Conversion

The planar model uses a virtual wheel to actuate the system. The real system has an actuating structure which differs strongly from the one assumed in the planar model. Since a controller for the planar model is going to be implemented on the real system, conversions have to be calculated. In order to be able to control the real system, the torques on the virtual motors have to be converted into the torques for the real motors.

### Definitions

One element of the real system (omniwheel driven by a motor) is shown on the left side in figure C.1. It actuates the ball by scrolling on a circle on the ball, characterized by the motor arrangement angle $\alpha$.
The right side of figure C.1 shows a top view on the real actuating system where the torque of each omniwheel generates a tangential force on the surface of the ball.

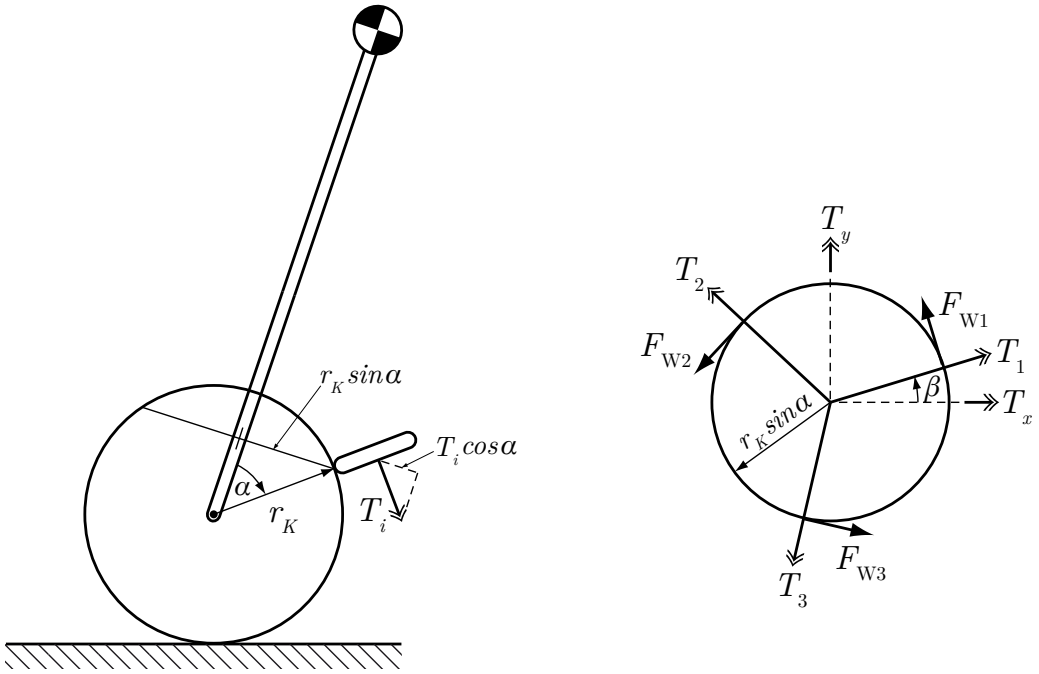



*Figure C.1:* Torques and tangential forces generated by the real actuating system

The following relationships are read out of figure C.1.

Tangential forces of the real configuration:

$$F_{W,1} = \frac{T_1}{r_W} \cdot \begin{bmatrix} -\sin\beta \\ \cos\beta \\ 0 \end{bmatrix} \tag{C.1}$$

$$F_{W,2} = \frac{T_2}{r_W} \cdot \begin{bmatrix} -\sin(\beta + \frac{2}{3}\pi) \\ \cos(\beta + \frac{2}{3}\pi) \\ 0 \end{bmatrix} \tag{C.2}$$

$$F_{W,3} = \frac{T_3}{r_W} \cdot \begin{bmatrix} -\sin(\beta - \frac{2}{3}\pi) \\ \cos(\beta - \frac{2}{3}\pi) \\ 0 \end{bmatrix} \tag{C.3}$$

Lever for real configuration:

$$r_{KW,1} = r_K \cdot \begin{bmatrix} \cos\beta\sin\alpha \\ \sin\beta\sin\alpha \\ \cos(\alpha \end{bmatrix} \tag{C.4}$$

$$r_{KW,2} = r_K \cdot \begin{bmatrix} \cos(\beta + \frac{2}{3}\pi)\sin(\alpha) \\ \sin(\beta + \frac{2}{3}\pi)\sin(\alpha) \\ \cos(\alpha) \end{bmatrix} \tag{C.5}$$

$$r_{KW,3} = r_K \cdot \begin{bmatrix} \cos(\beta - \frac{2}{3}\pi)\sin(\alpha) \\ \sin(\beta - \frac{2}{3}\pi)\sin(\alpha) \\ \cos(\alpha) \end{bmatrix} \tag{C.6}$$

Tangetial forces of the virtual configuration plane:

$$F_{W,x} = \frac{T_x}{r_W} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \text{(actually } y\text{-direction)} \tag{C.7}$$

$$F_{W,y} = \frac{T_y}{r_W} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \text{(actually } x\text{-direction)} \tag{C.8}$$

$$F_{W,z} = \frac{T_z}{r_W} \cdot \begin{bmatrix} -\sin\beta \\ \cos\beta \\ 0 \end{bmatrix} \tag{C.9}$$

Lever for the virtual configuration plane:

$$r_{KW,x} = r_{KW,y} = r_K \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{C.10}$$

$$r_{KW,z} = r_K \cdot \begin{bmatrix} \cos\beta \cdot \sin\alpha \\ \sin\beta \cdot \sin\alpha \\ 0 \end{bmatrix} \tag{C.11}$$

**Torque on the ball**

With given relationships the torque on the ball from the real and the virtual driving mechanism can be calculated.
Torques on the Ball from the real omniwheels:

$$T_{KW,1} = r_{KW,1} \times F_{W,1} \tag{C.12}$$

$$T_{KW,2} = r_{KW,2} \times F_{W,2} \tag{C.13}$$

$$T_{KW,3} = r_{KW,3} \times F_{W,3} \tag{C.14}$$

Torques on the Ball from the virtual omniwheels:

$$T_{KW,x} = r_{KW,x} \times F_{W,x} \tag{C.15}$$

$$T_{KW,y} = r_{KW,y} \times F_{W,y} \tag{C.16}$$

$$T_{KW,z} = r_{KW,z} \times F_{W,z} \tag{C.17}$$

**Solution**

Following equation describes that the two different systems need to introduce the same torque on the ball.

$$T_{KW,1} + T_{KW,2} + T_{KW,3} = T_{KW,x} + T_{KW,y} + T_{KW,z} \tag{C.18}$$

The relationship between real and virtual driving mechanism can be obtained by solving for the unknowns.

Solved for real motor torques:

$$T_1 = \frac{1}{3} \cdot \left( T_z + \frac{2}{\cos\alpha} \cdot (T_x \cdot \cos\beta - T_y \cdot \sin\beta) \right) \tag{C.19}$$

$$T_2 = \frac{1}{3} \cdot \left( T_z + \frac{1}{\cos\alpha} \cdot \left( \sin\beta \cdot (-\sqrt{3}T_x + T_y) - \cos\beta \cdot (T_x + \sqrt{3}T_y) \right) \right) \tag{C.20}$$

$$T_3 = \frac{1}{3} \cdot \left( T_z + \frac{1}{\cos\alpha} \cdot \left( \sin\beta \cdot (\sqrt{3}T_x + T_y) + \cos\beta \cdot (-T_x + \sqrt{3}T_y) \right) \right) \tag{C.21}$$

(C.22)

Solved for virtual motor torques:

$$T_x = \cos\alpha \cdot \left( T_1 \cdot \cos\beta - T_2 \cdot \sin(\beta + \frac{\pi}{6}) + T_3 \cdot \sin(\beta - \frac{\pi}{6}) \right) \tag{C.23}$$

$$T_y = \cos\alpha \cdot \left( -T_1 \cdot \sin\beta - T_2 \cdot \cos(\beta + \frac{\pi}{6}) + T_3 \cdot \cos(\beta - \frac{\pi}{6}) \right) \tag{C.24}$$

$$T_z = T_1 + T_2 + T_3 \tag{C.25}$$

## C.2 Inertia Calculations

The aim of this section is to find a suitable approximation for the moments of inertia of the modeled drive mechanism, which is not the same as in reality. The following approach is made by comparing the energies stored in the real omniwheels and the virtual actuating wheel respectively, when driving at a constant velocity.

### Approximations for ball and body

$$\Theta_K = \frac{2}{3} m_K r_K^2 \qquad \text{(hollow sphere)} \tag{C.26}$$

$$\Theta_A = \frac{1}{4} \cdot m_A \cdot r_A^2 + \frac{1}{2} \cdot m_A \cdot h^2 + m_A \cdot l^2 \qquad \text{(cylinder)} \tag{C.27}$$

### Motors and omniwheels

The real rotor inertia of a motor is given by

$$\Theta_M = 3.33 \cdot 10^{-6}\,\mathrm{kgm}^2$$

and the real inertia of an omniwheel is calculated as:

$$\Theta_{OW} = \frac{1}{2} m_{OW} r_W^2 = 900 \cdot 10^{-6}\,\mathrm{kgm}^2$$

Since the reduction of the gear box is $i = 26$, the rotor of the motor turns 26 times faster. Therefore, a factor $i^2$ is added to calculate the energy ($E = 1/2 \cdot \Theta \cdot v^2$). So

both, omniwheel and motor inertias are taken into account. The inertia of the gear rotor is $9.1 \cdot 10^{-6}$ kgm$^2$ and therefore negligible.

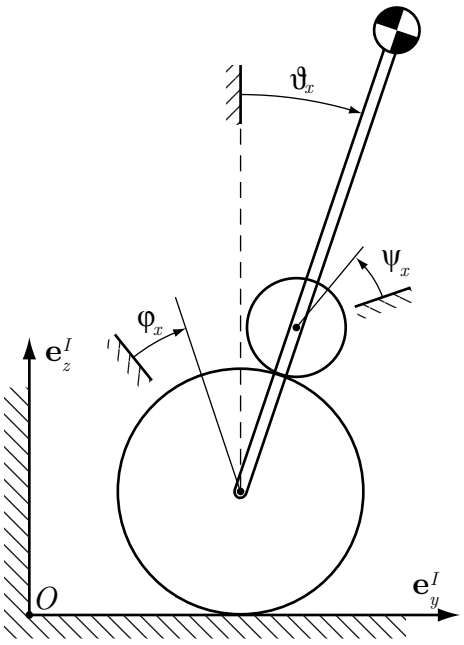



*Figure C.2:* planar model coordinates

Assuming any velocity $v$ in positive $y$ direction, the rotation speed of the sphere is:

$$\omega_{K,x} = \frac{v}{r_K} = \omega_{W,x}\frac{r_W}{r_K}, \quad \text{with } \omega_{W,x} = \frac{v}{r_W}$$

From the odometry we get the rotation speed of each omniwheel. For motion in $y$ direction:

$$\omega_{OW,1} = \omega_{W,x}\cos\alpha \tag{C.28}$$

$$\omega_{OW,2/3} = -\frac{1}{2}\omega_{W,x}\cos\alpha \tag{C.29}$$

in $x$ direction:

$$\omega_{OW,1} = 0 \tag{C.30}$$

$$\omega_{OW,2} = -\frac{\sqrt{3}}{2}\omega_{W,y}\cos\alpha \tag{C.31}$$

$$\omega_{OW,3} = \frac{\sqrt{3}}{2}\omega_{W,y}\cos\alpha \tag{C.32}$$

rotation around $z$-axis:

$$\omega_{OW,1/2/3} = \omega_{W,z}\sin\alpha \tag{C.33}$$

Equations for energy equilibrium in $y$ direction:

$$\begin{aligned}\frac{1}{2}\Theta_{W,x}\dot{\psi}_x^2 &= \frac{1}{2}\Theta_{OW}(\dot{\psi}_x\cos\alpha)^2 + \frac{1}{2}\Theta_M(i\cdot\dot{\psi}_x\cos\alpha)^2 \\ &\quad + \frac{1}{2}\cdot 2\left(\Theta_{OW}(-\frac{1}{2}\dot{\psi}_x\cos\alpha)^2 + \Theta_M(-\frac{1}{2}i\cdot\dot{\psi}_x\cos\alpha)^2\right) \\ \Theta_{W,x} &= \cos^2(\alpha)\left(\Theta_{OW} + i^2\Theta_M + \frac{1}{2}(\Theta_{OW} + i^2\Theta_M)\right) \\ &= \underline{\frac{3}{2}\cos^2(\alpha)(\Theta_{OW} + i^2\Theta_M)}\end{aligned}$$

in $x$ direction:

$$\begin{aligned}\frac{1}{2}\Theta_{W,y}\dot{\psi}_y^2 &= \frac{1}{2}\cdot 2\left(\Theta_{OW}(\frac{\sqrt{3}}{2}\dot{\psi}_y\cos\alpha)^2 + \Theta_M(-\frac{\sqrt{3}}{2}i\cdot\dot{\psi}_x\cos\alpha)^2\right) \\ \Theta_{W,y} &= \underline{\frac{3}{2}\cos^2(\alpha)(\Theta_{OW} + i^2\Theta_M)}\end{aligned}$$

**Note:** following notation was used:

$$\begin{aligned}\dot{\psi}_x &= \omega_{W,x} \\ \dot{\psi}_y &= \omega_{W,y} \\ \dot{\psi}_1 &= \omega_{OW,1} \\ \dot{\psi}_2 &= \omega_{OW,2} \\ \dot{\psi}_3 &= \omega_{OW,3}\end{aligned}$$

### Numerical comparison

Moments of inertia:

$$\begin{aligned}\Theta_K &= \frac{2}{3}\cdot 2.29\,\mathrm{kg}\cdot(0.125\,\mathrm{m})^2 = 0.0239\,\mathrm{kgm}^2 \\ \Theta_A &= \frac{1}{4}\cdot 12.2\,\mathrm{kg}\cdot(0.1\,\mathrm{m})^2 + \frac{1}{2}\cdot 12.2\,\mathrm{kg}\cdot(0.5\,\mathrm{m})^2 + 12.2\,\mathrm{kg}\cdot(0.339\,\mathrm{m})^2 = 2.96\,\mathrm{kgm}^2 \\ \Theta_W &= \frac{3}{2}\cdot\left(\frac{\sqrt{2}}{2}\right)^2(900\cdot 10^{-6}\,\mathrm{kgm}^2 + 26^2\cdot 3.33\cdot 10^{-6}\,\mathrm{kgm}^2) = 2.36\cdot 10^{-3}\,\mathrm{kgm}^2\end{aligned}$$

Rotation energies ($v = 3$ m/s and $\dot{\vartheta}_x = 1$ rad/s):

$$E_K = \frac{1}{2}\Theta_K \dot{\varphi}_x^2 = 0.5 \cdot 0.0239\,\mathrm{kgm}^2 \cdot \left(\frac{3\,\mathrm{m/s}}{0.125\,\mathrm{m}}\right)^2 = 6.88\,\mathrm{J}$$
$$E_A = \frac{1}{2}\Theta_A \dot{\vartheta}_x^2 = 0.5 \cdot 2.29\,\mathrm{kgm}^2 \cdot (1/\mathrm{s})^2 = 1.48\,\mathrm{J}$$
$$E_W = \frac{1}{2}\Theta_W \dot{\psi}_x^2 = 0.5 \cdot 0.00236\,\mathrm{kgm}^2 \cdot \left(\frac{3\,\mathrm{m/s}}{0.06\,\mathrm{m}}\right)^2 = 2.95\,\mathrm{J}$$

**Moments of inertia for the $xy$-plane**

$$\Theta_{A,xy} = \frac{1}{2} \cdot (m_A + m_W) \cdot r_A^2 \tag{C.34}$$
$$\Theta_{W,xy} = 3 \cdot (\Theta_{OW} + i^2 \cdot \Theta_M) \tag{C.35}$$

# D Odometry

# Odometry

## Parameters

- **Radius of the ball**

```
rK;
```

- **Radius of the Omniwheel**

```
rW;
```

- **Radius of the body**

```
rA;
```

- **Distance between center of the ball an center of gravity of the body**

```
l;
```

- **Angle of the motors**

```
alpha = 45 °;
beta1 = 0 °;
beta2 = 120 °;
beta3 = 240 °;
```

- **Rotation speed vector of the ball in the Lisa reference frame L**

```
L⌣OmegaK[t] = {phix'[t], phiy'[t], phiz'[t]};
```

- **Rotation speed of the omniwheel about the motor axis in the body reference frame A (scalar, encoder value)**

```
A⌣omegaW1[t] = psi1'[t];
A⌣omegaW2[t] = psi2'[t];
A⌣omegaW3[t] = psi3'[t];
```

## Coordinates and rotation matrices (I↔L↔A)

- **Rotation matrices I↔L↔A**

Rotations around z, y and x Axes

```
Rz = {{Cos[thetaz[t]], -Sin[thetaz[t]], 0},
    {Sin[thetaz[t]], Cos[thetaz[t]], 0}, {0, 0, 1}};
Ry = {{Cos[thetay[t]], 0,  Sin[thetay[t]]}, {0, 1, 0},
    {-Sin[thetay[t]], 0, Cos[thetay[t]]}};
Rx = {{1, 0, 0}, {0, Cos[thetax[t]], -Sin[thetax[t]]},
    {0, Sin[thetax[t]], Cos[thetax[t]]}};
```

Rotation Matrix (I to L and L to I)

```
RIL = Rz;
RLI = Transpose[RIL];
```

Rotation Matrix (L to A and A to L)

```
RLA = Ry.Rx;
RAL = Transpose[RLA];
```

Rotation Matrix (I to A and A to I)

```
RIA = RIL.RLA;
RAI = Transpose[RIA];
```

### ■ Ball

Rotation of the ball in I

```
OmegaK[t] = RIL.L⌣OmegaK[t];
```

### ■ Jacob

Jacobian Matrix

```
J = {{1, 0, -Sin[thetay[t]]}, {0, Cos[thetax[t]], Sin[thetax[t]] * Cos[thetay[t]]},
    {0, -Sin[thetax[t]], Cos[thetax[t]] * Cos[thetay[t]]}};
```

Time variation of Tait-Bryan angles

```
ThetaDot[t] = {thetax'[t], thetay'[t], thetaz'[t]};
```

Rotation Vector of the Aufbau (in A)

```
A⌣OmegaA[t] = J.ThetaDot[t];
```

## Rotation of the ball

Vector from center of the ball (P) to the contact point with the omniwheels (K1, K2, K3) in A

```
A⌣rPK1 = {rK * Sin[alpha] * Cos[beta1], rK * Sin[alpha] * Sin[beta1], rK * Cos[alpha]};
A⌣rPK2 = {rK * Sin[alpha] * Cos[beta2], rK * Sin[alpha] * Sin[beta2], rK * Cos[alpha]};
A⌣rPK3 = {rK * Sin[alpha] * Cos[beta3], rK * Sin[alpha] * Sin[beta3], rK * Cos[alpha]};
```

Direction of the tangential speed of the rotation of the omniwheels in A

```
A⌣d1 = {-Sin[beta1], Cos[beta1], 0};
A⌣d2 = {-Sin[beta2], Cos[beta2], 0};
A⌣d3 = {-Sin[beta3], Cos[beta3], 0};
```

Test : Absolute value has to be 1! (ok)

```
Norm[A⌣d1];
Norm[A⌣d2];
Norm[A⌣d3];
```

Rotation speed of the ball relative to the body in A

```
A⌣omegaK = {A⌣omegaKx, A⌣omegaKy, A⌣omegaKz};
```

Speed on the surface of the ball (in omniwheel direction) has to be the same speed as the tangetial speed of the omniwheel

```
E1 = Cross[A⌣omegaK, A⌣rPK1].A⌣d1 == A⌣omegaW1[t] * rW;
E2 = Cross[A⌣omegaK, A⌣rPK2].A⌣d2 == A⌣omegaW2[t] * rW;
E3 = Cross[A⌣omegaK, A⌣rPK3].A⌣d3 == A⌣omegaW3[t] * rW;

sol = Solve[{E1, E2, E3}, {A⌣omegaKx, A⌣omegaKy, A⌣omegaKz}];

A⌣omegaK = {A⌣omegaKx /. sol[[1]], A⌣omegaKy /. sol[[1]], A⌣omegaKz /. sol[[1]]};
```

Rotation speed of the ball in A

```
A‿OmegaK = A‿omegaK + A‿OmegaA[t];
```

Coordinate transformation from A to I System

```
L‿OmegaK[t] = FullSimplify[RLA.A‿OmegaK];
```

Solution

```
phix'[t] = L‿OmegaK[t][[1]]
```

$$\frac{1}{3\,rK}\left(\sqrt{6}\;rW\,\text{Sin}[\text{thetax}[t]]\,\text{Sin}[\text{thetay}[t]]\,(-\text{psi2}'[t]+\text{psi3}'[t])+\sqrt{2}\;rW\,\text{Cos}[\text{thetax}[t]]\,\text{Sin}[\text{thetay}[t]]\,(\text{psi1}'[t]+\text{psi2}'[t]+\text{psi3}'[t])+\text{Cos}[\text{thetay}[t]]\left(\sqrt{2}\;rW\,(-2\,\text{psi1}'[t]+\text{psi2}'[t]+\text{psi3}'[t])+3\,rK\,\text{thetax}'[t]\right)\right)$$

```
phiy'[t] = L‿OmegaK[t][[2]]
```

$$\frac{1}{3\,rK}\left(\sqrt{6}\;rW\,\text{Cos}[\text{thetax}[t]]\,(-\text{psi2}'[t]+\text{psi3}'[t])-\sqrt{2}\;rW\,\text{Sin}[\text{thetax}[t]]\,(\text{psi1}'[t]+\text{psi2}'[t]+\text{psi3}'[t])+3\,rK\,\text{thetay}'[t]\right)$$

```
phiz'[t] = L‿OmegaK[t][[3]]
```

$$\frac{1}{3\,rK}\left(\sqrt{2}\;rW\,(\text{Cos}[\text{thetax}[t]]\,\text{Cos}[\text{thetay}[t]]+2\,\text{Sin}[\text{thetay}[t]])\,\text{psi1}'[t]+\sqrt{2}\;rW\left(\sqrt{3}\;\text{Cos}[\text{thetay}[t]]\,\text{Sin}[\text{thetax}[t]]\,(-\text{psi2}'[t]+\text{psi3}'[t])+\text{Cos}[\text{thetax}[t]]\,\text{Cos}[\text{thetay}[t]]\,(\text{psi2}'[t]+\text{psi3}'[t])-\text{Sin}[\text{thetay}[t]]\,(\text{psi2}'[t]+\text{psi3}'[t])\right)+3\,rK\,(-\text{Sin}[\text{thetay}[t]]\,\text{thetax}'[t]+\text{thetaz}'[t])\right)$$

## Translation of the ball

Vector from the ground to center of the ball (P)

```
rBP = {0, 0, rK};
```

Substitution

```
OmegaK‿given[t] = RIL.{phix‿given'[t], phiy‿given'[t], phiz‿given'[t]};
```

Speed vector of the center of the ball (P) in I

Lightweight solution with use of solutions above

```
rPdot‿light = FullSimplify[Cross[OmegaK‿given[t], rBP]]
```

```
{rK (Sin[thetaz[t]] phix‿given'[t] + Cos[thetaz[t]] phiy‿given'[t]),
 -rK Cos[thetaz[t]] phix‿given'[t] + rK Sin[thetaz[t]] phiy‿given'[t], 0}
```

Explicit solution

```
rPdot = FullSimplify[Cross[OmegaK[t], rBP]]
```

$$\left\{\frac{1}{3}\left(\sqrt{2}\ \text{rW}\ \text{Sin}[\text{thetay}[t]]\ \text{Sin}[\text{thetaz}[t]]\left(\sqrt{3}\ \text{Sin}[\text{thetax}[t]]\ (-\text{psi2}'[t]+\text{psi3}'[t])+\right.\right.\right.$$
$$\left.\left.\text{Cos}[\text{thetax}[t]]\ (\text{psi1}'[t]+\text{psi2}'[t]+\text{psi3}'[t])\right)+\text{Cos}[\text{thetay}[t]]$$
$$\text{Sin}[\text{thetaz}[t]]\left(\sqrt{2}\ \text{rW}\ (-2\ \text{psi1}'[t]+\text{psi2}'[t]+\text{psi3}'[t])+3\ \text{rK}\ \text{thetax}'[t]\right)+$$
$$\text{Cos}[\text{thetaz}[t]]\left(\sqrt{6}\ \text{rW}\ \text{Cos}[\text{thetax}[t]]\ (-\text{psi2}'[t]+\text{psi3}'[t])-\right.$$
$$\left.\sqrt{2}\ \text{rW}\ \text{Sin}[\text{thetax}[t]]\ (\text{psi1}'[t]+\text{psi2}'[t]+\text{psi3}'[t])+3\ \text{rK}\ \text{thetay}'[t]\right)\Big),$$
$$\frac{1}{3}\left(\sqrt{2}\ \text{rW}\left((\text{Cos}[\text{thetaz}[t]]\ (2\ \text{Cos}[\text{thetay}[t]]-\text{Cos}[\text{thetax}[t]]\ \text{Sin}[\text{thetay}[t]])-\right.\right.$$
$$\left.\text{Sin}[\text{thetax}[t]]\ \text{Sin}[\text{thetaz}[t]])\ \text{psi1}'[t]-\left(\text{Cos}[\text{thetaz}[t]]\right.\right.$$
$$\left.\left(\text{Cos}[\text{thetay}[t]]+\left(\text{Cos}[\text{thetax}[t]]-\sqrt{3}\ \text{Sin}[\text{thetax}[t]]\right)\text{Sin}[\text{thetay}[t]]\right)+\right.$$
$$\left.\left(\sqrt{3}\ \text{Cos}[\text{thetax}[t]]+\text{Sin}[\text{thetax}[t]]\right)\text{Sin}[\text{thetaz}[t]]\right)\text{psi2}'[t]-$$
$$\left(\text{Cos}[\text{thetay}[t]]\ \text{Cos}[\text{thetaz}[t]]+\text{Cos}[\text{thetaz}[t]]\right.$$
$$\left.\left(\text{Cos}[\text{thetax}[t]]+\sqrt{3}\ \text{Sin}[\text{thetax}[t]]\right)\text{Sin}[\text{thetay}[t]]+\right.$$
$$\left.\left(-\sqrt{3}\ \text{Cos}[\text{thetax}[t]]+\text{Sin}[\text{thetax}[t]]\right)\text{Sin}[\text{thetaz}[t]]\right)\text{psi3}'[t]\Big)-$$
$$\left.3\ \text{rK}\ \text{Cos}[\text{thetay}[t]]\ \text{Cos}[\text{thetaz}[t]]\ \text{thetax}'[t]+3\ \text{rK}\ \text{Sin}[\text{thetaz}[t]]\ \text{thetay}'[t]\right),\ 0\Big\}$$

# E 3D System Modeling

# System Modeling 3D

## Parameters

- **Radius of the ball**

```
rK;
```

- **Radius of the Omniwheel**

```
rW;
```

- **Radius of the body**

```
rA;
```

- **Distance between center of the ball an center of gravity of the body**

```
l;
```

- **Mass of body and omniwheels**

```
mAW;
```

- **Mass of ball**

```
mK;
```

- **Inertia of body and omniwheels in the body reference frame A**

```
A⌣ThetaAW = {{A⌣ThetaAWx, 0, 0}, {0, A⌣ThetaAWy, 0}, {0, 0, A⌣ThetaAWz}};
```

- **Inertia of ball in the intertial reference frame I or L**

```
ThetaK = {{ThetaKi, 0, 0}, {0, ThetaKi, 0}, {0, 0, ThetaKi}};
```

- **Inertia of omniwheel and motor (gear ratio!) about the motor axis**

```
ThetaWi;
```

- **Angle of the motors**

```
alpha = 45 °;
beta1 = 0 °;
beta2 = 120 °;
beta3 = 240 °;
```

- **Rotation speed vector of the ball in the Lisa reference frame L (no Rotation of the ball around the z-axis in L, which is same as the z-axis in I)**

```
L⌣OmegaK[t] = {phix'[t], phiy'[t], 0};
```

- **Relative rotation speed between the omniwheels and body about the motor axis in the body reference frame A (scalar, encoder value)**

```
A‿omegaW1[t] = psi1'[t];
A‿omegaW2[t] = psi2'[t];
A‿omegaW3[t] = psi3'[t];
```

- **Gravity vector**

```
G = {0, 0, -g};
```

## Minimal coordinates

Phix and phiy are not orientation angles but "Abrollwinkel" of the ball! Phiz is not a state, since it's assumed to be always 0.

```
q[t_] = {thetax[t], thetay[t], thetaz[t], phix[t], phiy[t]};
```

## Coordinates and rotation matrices (I↔L↔A)

### Rotation matrices I↔L↔A

Rotations around z, y and x Axes

```
Rz = {{Cos[thetaz[t]], -Sin[thetaz[t]], 0},
    {Sin[thetaz[t]], Cos[thetaz[t]], 0}, {0, 0, 1}};
Ry = {{Cos[thetay[t]], 0,  Sin[thetay[t]]}, {0, 1, 0},
    {-Sin[thetay[t]], 0, Cos[thetay[t]]}};
Rx = {{1, 0, 0}, {0, Cos[thetax[t]], -Sin[thetax[t]]},
    {0, Sin[thetax[t]], Cos[thetax[t]]}};
```

Rotation Matrix (I to L and L to I)

```
RIL = Rz;
RLI = Transpose[RIL];
```

Rotation Matrix (I to A and A to I)

```
RIA = RIL.Ry.Rx;
RAI = Transpose[RIA];
```

### Ball

Rotation of the ball in I

```
OmegaK[t] = RIL.L‿OmegaK[t];
```

### Jacob

Jacobian Matrix

```
J = {{1, 0, -Sin[thetay[t]]}, {0, Cos[thetax[t]], Sin[thetax[t]] * Cos[thetay[t]]},
    {0, -Sin[thetax[t]], Cos[thetax[t]] * Cos[thetay[t]]}};
```

Time variation of Tait-Bryan angles

```
ThetaDot[t] = {thetax'[t], thetay'[t], thetaz'[t]};
```

Rotation Vector of the Aufbau (in A)

```
A‿OmegaA[t] = J.ThetaDot[t];
```

## Binding equations

### ■ Absolut rotation of the omniwheels

Vector from intersection of the motor directions M to the center point of the omniwheels (W1, W2, W3) in A

```
A‿MW1 = {Cos[beta1] * Sin[alpha], Sin[alpha] * Sin[beta1], -Cos[alpha]};
A‿MW2 = {Cos[beta2] * Sin[alpha], Sin[alpha] * Sin[beta2], -Cos[alpha]};
A‿MW3 = {Cos[beta3] * Sin[alpha], Sin[alpha] * Sin[beta3], -Cos[alpha]};
```

Absolute rotation speed of the omniwheels about the motor axis in the body reference frame A (scalar)

```
A‿OmegaW1[t] = FullSimplify[A‿omegaW1[t] + A‿MW1.A‿OmegaA[t]];
A‿OmegaW2[t] = FullSimplify[A‿omegaW2[t] + A‿MW2.A‿OmegaA[t]];
A‿OmegaW3[t] = FullSimplify[A‿omegaW3[t] + A‿MW3.A‿OmegaA[t]];
```

### ■ Dependency on the rotation of the omniwheels

Vector from center of the ball (P) to the contact point with the omniwheels (K1, K2, K3) in A

```
A‿rPK1 = {rK * Sin[alpha] * Cos[beta1], rK * Sin[alpha] * Sin[beta1], rK * Cos[alpha]};
A‿rPK2 = {rK * Sin[alpha] * Cos[beta2], rK * Sin[alpha] * Sin[beta2], rK * Cos[alpha]};
A‿rPK3 = {rK * Sin[alpha] * Cos[beta3], rK * Sin[alpha] * Sin[beta3], rK * Cos[alpha]};
```

Direction of the tangential speed of the rotation of the omniwheels in A

```
A‿d1 = {-Sin[beta1], Cos[beta1], 0};
A‿d2 = {-Sin[beta2], Cos[beta2], 0};
A‿d3 = {-Sin[beta3], Cos[beta3], 0};
```

Rotation speed of the ball in A

```
A‿OmegaK[t] = RAI.OmegaK[t];
```

Rotation speed of the ball relative to the body in A

```
A‿omegaK[t] = FullSimplify[A‿OmegaK[t] - A‿OmegaA[t]];
```

Speed on the surface of the ball (in omniwheel direction) has to be the same speed as the tangetial speed of the omniwheel

```
E1 = Cross[A‿omegaK[t], A‿rPK1].A‿d1 == A‿omegaW1[t] * rW;
E2 = Cross[A‿omegaK[t], A‿rPK2].A‿d2 == A‿omegaW2[t] * rW;
E3 = Cross[A‿omegaK[t], A‿rPK3].A‿d3 == A‿omegaW3[t] * rW;
```

```
sol = Solve[{E1, E2, E3}, {psi1'[t], psi2'[t], psi3'[t]}];
```

```
psi1'[t] = FullSimplify[psi1'[t] /. sol[[1]]];
psi2'[t] = FullSimplify[psi2'[t] /. sol[[1]]];
psi3'[t] = FullSimplify[psi3'[t] /. sol[[1]]];
```

Simplify

```
A‿OmegaW1[t] = FullSimplify[A‿OmegaW1[t]];
A‿OmegaW2[t] = FullSimplify[A‿OmegaW2[t]];
A‿OmegaW3[t] = FullSimplify[A‿OmegaW3[t]];
```

### ■ Translation of the ball

Vector from the ground to center of the ball (P)

```
rBP = {0, 0, rK};
```

Speed vector of the center of the ball (P) in I

```
rPdot = FullSimplify[Cross[OmegaK[t], rBP]];
```

## Energies

### Ball

#### Kinetic

```
T◡K = FullSimplify[1 / 2 * mK * rPdot.rPdot + 1 / 2 * L◡OmegaK[t].ThetaK.L◡OmegaK[t]];
```

#### Potential

```
V◡K = 0;
```

### Aufbau (und Omniwheels)

Vector from center of the ball to the center of gravity of the body in A

```
A◡rPSA = {0, 0, l};
```

#### Kinetic

```
T◡AW =
  FullSimplify[1 / 2 * mAW * rPdot.rPdot + mAW * (RAI.rPdot).(Cross[A◡OmegaA[t], A◡rPSA]) +
    1 / 2 * A◡OmegaA[t].A◡ThetaAW.A◡OmegaA[t]];
```

#### Potential

```
V◡AW = -mAW * G.RIA.A◡rPSA;
```

### Rotational energy of the omniwheels

Considering only the rotation energy of the omniwheel and motors about the motor axis

```
T◡W1 = FullSimplify[1 / 2 * ThetaWi * A◡OmegaW1[t]^2];
T◡W2 = FullSimplify[1 / 2 * ThetaWi * A◡OmegaW2[t]^2];
T◡W3 = FullSimplify[1 / 2 * ThetaWi * A◡OmegaW3[t]^2];
```

## Non potentional forces

### Torque on omniwheels

```
{JT1, R1} = FullSimplify[PolynomialReduce[A◡omegaW1[t], q'[t], q'[t]]];
{JT2, R2} = FullSimplify[PolynomialReduce[A◡omegaW2[t], q'[t], q'[t]]];
{JT3, R3} = FullSimplify[PolynomialReduce[A◡omegaW3[t], q'[t], q'[t]]];
```

Torques

```
T1[t];
T2[t];
T3[t];
```

### Counter torques on body

```
JCT = {PolynomialReduce[A⌣OmegaA[t][[1]], q'[t], q'[t]][[1]],
   PolynomialReduce[A⌣OmegaA[t][[2]], q'[t], q'[t]][[1]],
   PolynomialReduce[A⌣OmegaA[t][[3]], q'[t], q'[t]][[1]]};
```

Counter torques

```
TC1[t] = A⌣MW1 * (-T1[t]);
TC2[t] = A⌣MW2 * (-T2[t]);
TC3[t] = A⌣MW3 * (-T3[t]);
```

- **f_NP**

```
fNP[t_] = FullSimplify[JT1 * T1[t] + JT2 * T2[t] + JT3 * T3[t] +
     Transpose[JCT].TC1[t] + Transpose[JCT].TC2[t] + Transpose[JCT].TC3[t]];
```

## Equation of motion

- **Total kinetic energy**

```
T[t_] = FullSimplify[T⌣K + T⌣AW + T⌣W1 + T⌣W2 + T⌣W3];
```

- **Total potentional energy**

```
V[t_] = V⌣K + V⌣AW;
```

- **Lagrange II**

```
L1[t_] = FullSimplify[D[D[T[t], {q'[t]}], t]];

L2[t_] = FullSimplify[D[T[t], {q[t]}]];

L3[t_] = D[V[t], {q[t]}];

EQ = L1[t] - L2[t] + L3[t] - fNP[t] == 0;

EQ1 = FullSimplify[EQ[[1]][[1]]];
EQ2 = FullSimplify[EQ[[1]][[2]]];
EQ3 = FullSimplify[EQ[[1]][[3]]];
EQ4 = FullSimplify[EQ[[1]][[4]]];
EQ5 = FullSimplify[EQ[[1]][[5]]];
```

## Solve for second derivative

```
varList = {thetax''[t], thetay''[t], thetaz''[t], phix''[t], phiy''[t]};

{q1, r1} = FullSimplify[PolynomialReduce[EQ1, varList, varList]];
{q2, r2} = FullSimplify[PolynomialReduce[EQ2, varList, varList]];
{q3, r3} = FullSimplify[PolynomialReduce[EQ3, varList, varList]];
{q4, r4} = FullSimplify[PolynomialReduce[EQ4, varList, varList]];
{q5, r5} = FullSimplify[PolynomialReduce[EQ5, varList, varList]];
```

Solve system with substitution

```
q1S = {q11S, q12S, q13S, q14S, q15S};
q2S = {q21S, q22S, q23S, q24S, q25S};
q3S = {q31S, q32S, q33S, q34S, q35S};
q4S = {q41S, q42S, q43S, q44S, q45S};
q5S = {q51S, q52S, q53S, q54S, q55S};

varSolS =
  FullSimplify[LinearSolve[{q1S, q2S, q3S, q4S, q5S}, {-r1S, -r2S, -r3S, -r4S, -r5S}]];

HeavySubstitution = {q11S -> q1[[1]],
    q12S -> q1[[2]],
    q13S -> q1[[3]],
    q14S -> q1[[4]],
    q15S -> q1[[5]],
    q21S -> q2[[1]],
    q22S -> q2[[2]],
    q23S -> q2[[3]],
    q24S -> q2[[4]],
    q25S -> q2[[5]],
    q31S -> q3[[1]],
    q32S -> q3[[2]],
    q33S -> q3[[3]],
    q34S -> q3[[4]],
    q35S -> q3[[5]],
    q41S -> q4[[1]],
    q42S -> q4[[2]],
    q43S -> q4[[3]],
    q44S -> q4[[4]],
    q45S -> q4[[5]],
    q51S -> q5[[1]],
    q52S -> q5[[2]],
    q53S -> q5[[3]],
    q54S -> q5[[4]],
    q55S -> q5[[5]],
    r1S -> r1,
    r2S -> r2,
    r3S -> r3,
    r4S -> r4,
    r5S -> r5};

varSol = varSolS /. HeavySubstitution;
```

## Linearization

State variables

```
x = {x1[t], x2[t], x3[t], x4[t], x5[t], x6[t], x7[t], x8[t], x9[t], x10[t]};
```

Input variables

```
u = {u1[t], u2[t], u3[t]};
```

Output values

```
h = {x1[t], x2[t], x3[t], x4[t], x5[t], x6[t], x7[t], x8[t], x9[t], x10[t]};
```

State space representation

```
SPrep = {
    thetax[t] → x1[t], thetax'[t] → x2[t],
    thetay[t] → x3[t], thetay'[t] → x4[t],
    thetaz[t] → x5[t], thetaz'[t] → x6[t],
    phix[t] → x7[t], phix'[t] → x8[t],
    phiy[t] → x9[t], phiy'[t] → x10[t],
    T1[t] → u1[t], T2[t] → u2[t], T3[t] → u3[t]};
```

Equation of movement in state variables form

```
f = {x2[t], varSol[[1]], x4[t], varSol[[2]], x6[t],
     varSol[[3]], x8[t], varSol[[4]], x10[t], varSol[[5]]} /. SPrep;
```

Point of linearization

```
equilibrium = {x1[t] → 0, x2[t] → 0, x3[t] → 0, x4[t] → 0, x5[t] → 0, x6[t] → 0,
    x7[t] → 0, x8[t] → 0, x9[t] → 0, x10[t] → 0, u1[t] → 0, u2[t] → 0, u3[t] → 0};
arbitrary = {x1[t] → LP1, x2[t] → LP2, x3[t] → LP3, x4[t] → LP4,
    x5[t] → LP5, x6[t] → LP6, x7[t] → LP7, x8[t] → LP8, x9[t] → LP9,
    x10[t] → LP10, u1[t] → LPu1, u2[t] → LPu2, u3[t] → LPu3};
```

- **A**

```
MatrixForm[AA0 = D[f, {x}] /. equilibrium];
MatrixForm[AA = D[f, {x}] /. arbitrary];
```

- **B**

```
MatrixForm[BB0 = D[f, {u}] /. equilibrium];
MatrixForm[BB = D[f, {u}] /. arbitrary];
```

- **C**

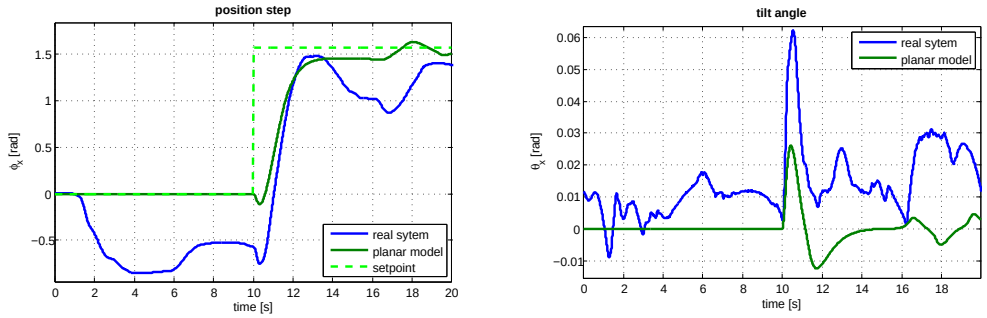
```
MatrixForm[CC0 = D[h, {x}] /. equilibrium];
MatrixForm[CC = D[h, {x}] /. arbitrary];
```

- **D**

```
MatrixForm[DD0 = D[h, {u}] /. equilibrium];
MatrixForm[DD = D[h, {u}] /. arbitrary];
```

# F Verification Plots

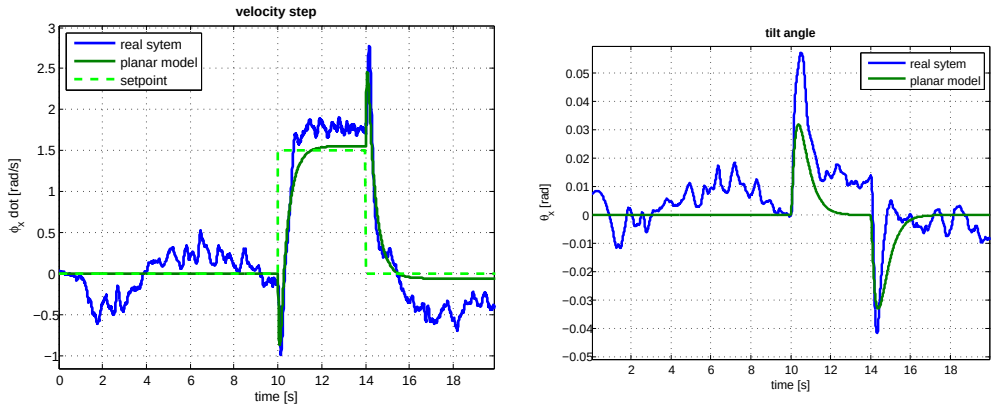
## F.1 Planar Model




(a) System and model responses, both with the same controller.

(b) Corresponding tilt angle of the system (blue) and of the model (green).

*Figure F.1:* Position step of 0.2 m in $y$ direction

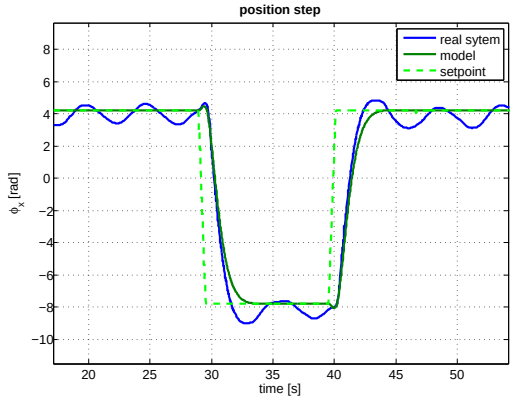



(a) Response of the system and of the model

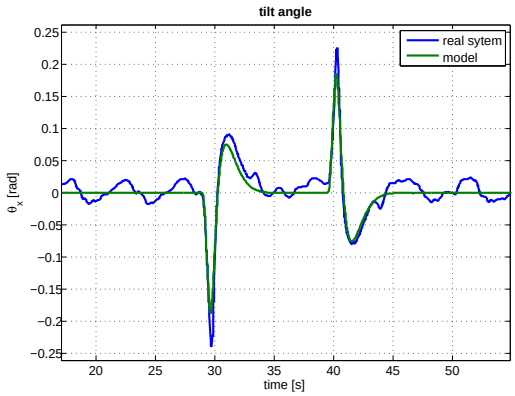
(b) Corresponding tilt angles

*Figure F.2:* Velocity step in $y$ direction from 0 m/s to 0.19 m/s and backwards

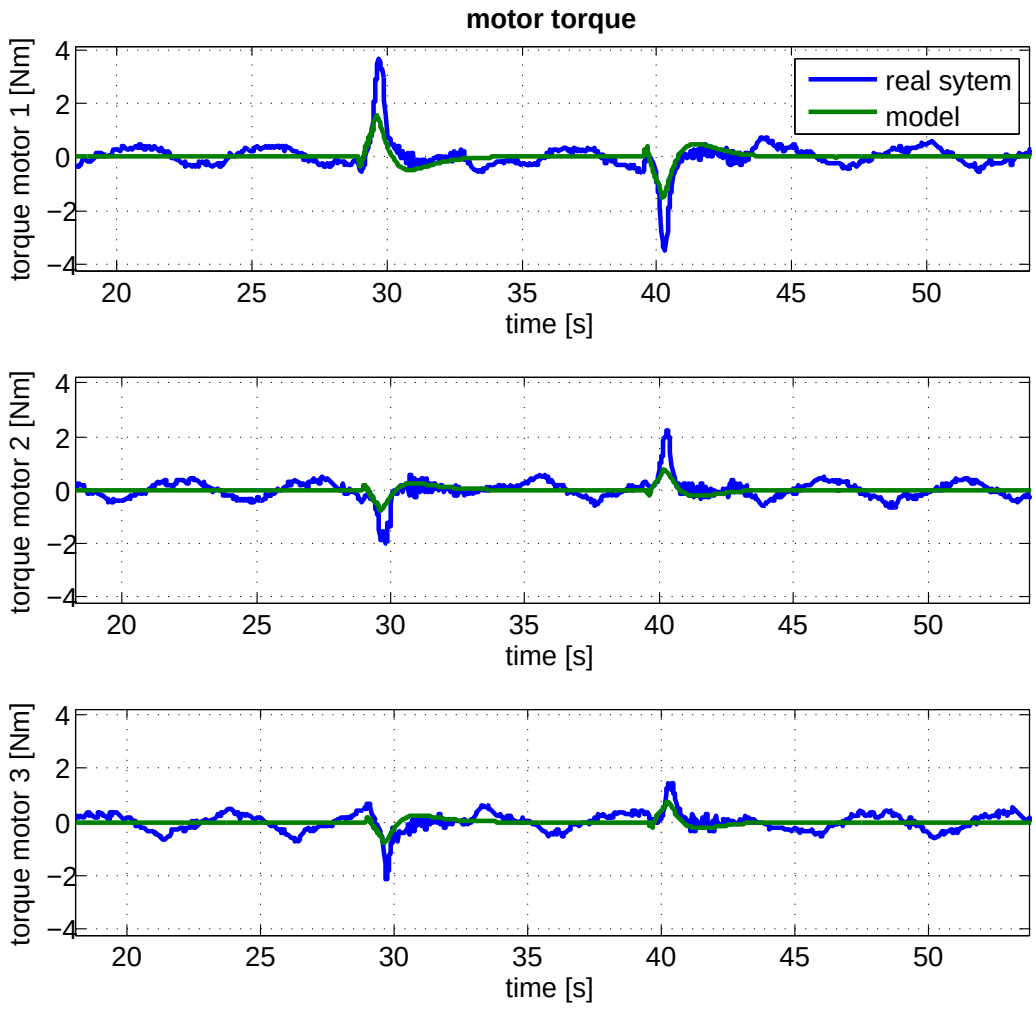
## F.2 3D Model




(a) System and model responses

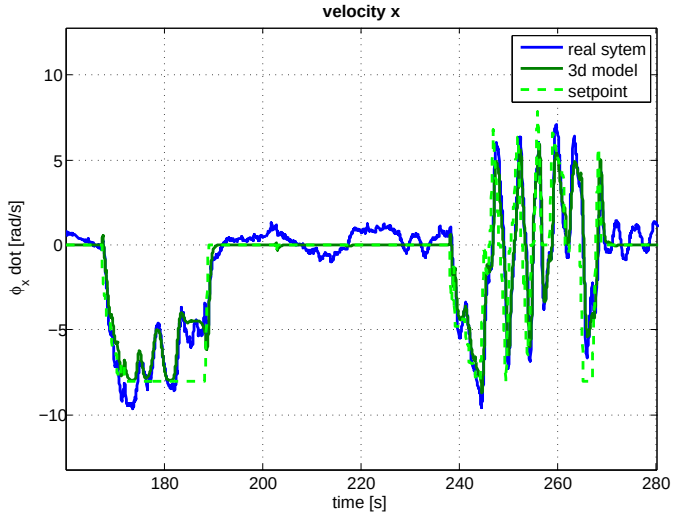



(b) System and model tilt angles

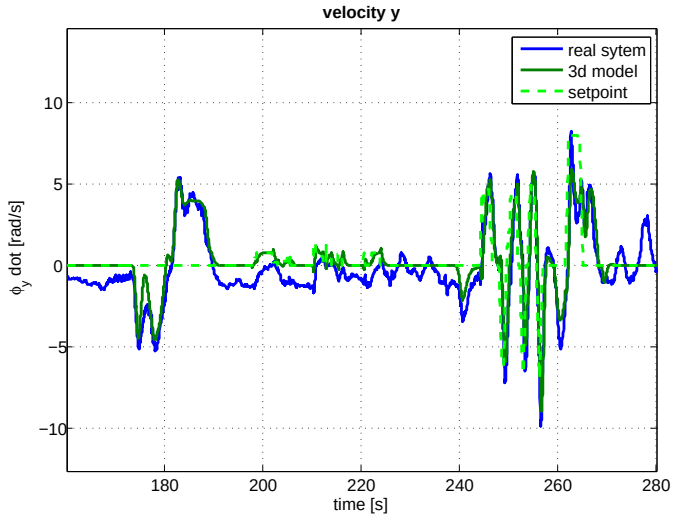



(c) System and model torques

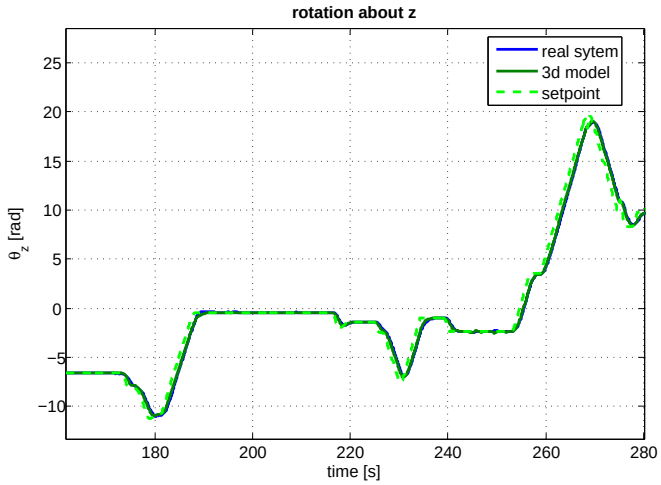
*Figure F.3:* Position step of 1.5 m in $y$ direction with the controller based on the 3d model.




(a) Set point for $\dot{\varphi}_x$ and corresponding responses.

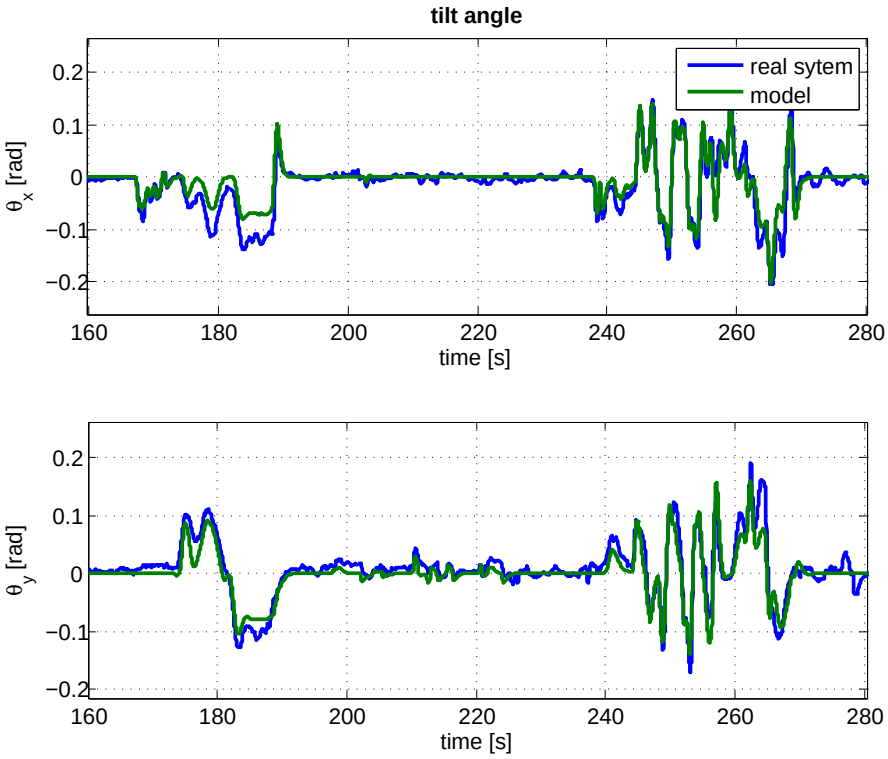



(b) Set point for $\dot{\varphi}_y$ and corresponding responses.

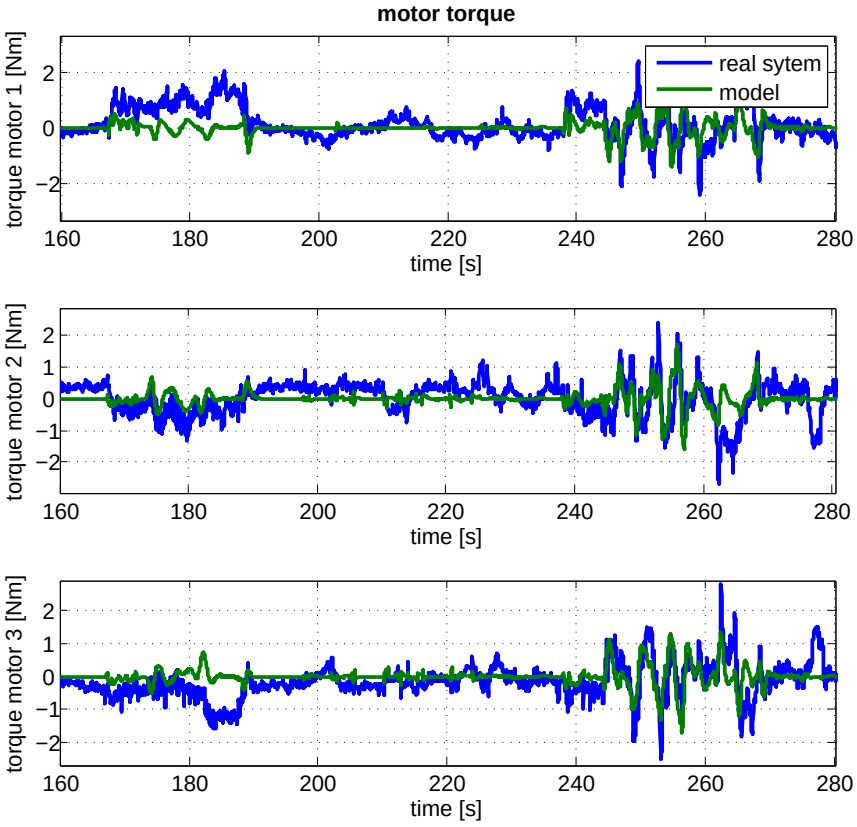



(c) Set point for $\vartheta_z$ and corresponding responses.

*Figure F.4:* Joystick input (dashed) and response of the system and of the model.




(a) System and model tilt angles




(b) System and model torques

*Figure F.5:* Tilt angles and system input for the joystick input (figure F.4).