

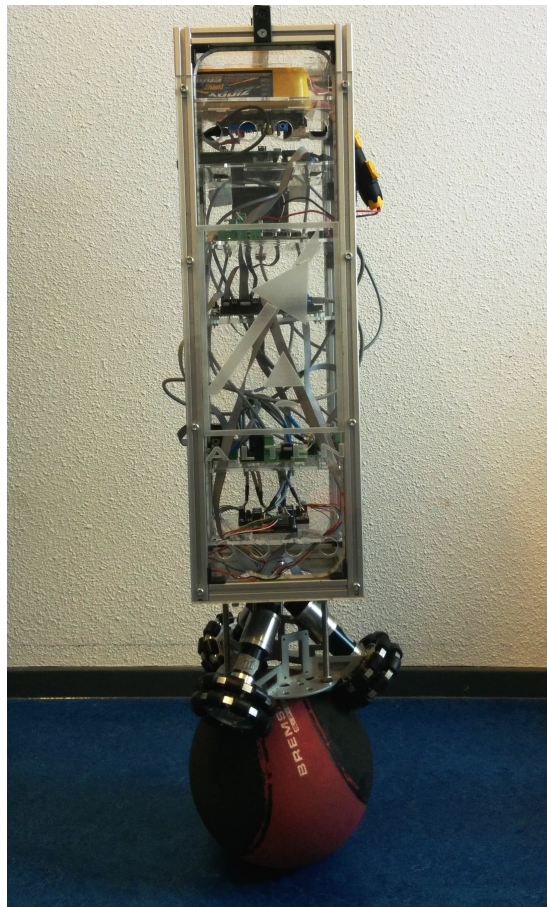
Internship & Master thesis

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)

Modeling and Control of a Ball-Balancing Robot

Internship & Master thesis at ALTEN Mechatronics

July, 2014



Author:

Koos van der Blonk

Supervisors:

Dr. ir. J.H. Sandee

Prof. dr. A.A. Stoorvogel

Ir. A.J.H. Verboord



UNIVERSITY OF TWENTE.

ABSTRACT

A Ball-Balancing Robot (BBR) is an omni-directional robot balancing on a single ball, which makes it inherently unstable. This project follows the goal of developing a model and a controller for this robot in order to make it balance and move around as a demonstrator on fairs.

To achieve this goal, the three-dimensional dynamic behaviour of the robot is approximated by three independent two-dimensional models. Due to neglected dependencies between the two-dimensional models and the conversions that had to be made from the two-dimensional models to the three-dimensional system, a three-dimensional model is developed to describe the full dynamic behaviour of the BBR and it is linearized around the position the BBR stands upright. Based on the linearized three-dimensional model, a linear controller is designed. The controller calculates the appropriate motor torques based on the measured tilt angle, required to keep the BBR dynamically stable. First a simple, easily implementable controller is designed with LQR control theory to make the BBR 'fair ready' as soon as possible. Later a more advanced controller is designed with SISO loopshaping. Simulations prove that the performance of the system with regard to balancing is significantly higher with the controller, designed with SISO loopshaping. During the implementation of the first controller, it turns out that noisy sensor data forms a serious restriction on the performance of the BBR. In particular, noisy gyroscope data limits the magnitude of the controller gains. The noise is significantly reduced by attenuating system vibrations, oversampling of the gyroscope and filtering the motor inputs, which results in a robot that is able to balance.

Furthermore, based on the developed three-dimensional model, research is done to investigate under what requirements the system remains stable with the developed controller. The aim of this research is to analyze the influence of uncertainties in the developed model and to predict the influence of future changes on the stability of the system.

ACKNOWLEDGEMENTS

I really enjoyed working on this project and I would like to thank everyone who contributed to that. I would like to thank a number of people in particular that have been of great importance for this report.

First of all, I would like to thank professor Anton Stoorvogel for supervising me during this project. His valuable feedback pointed me in the right direction and his critical eye helped me to stay scientific in an environment with all engineers.

I would also like to thank ALTEN that made this project possible and offered excellent support. Special thanks to my supervisors at ALTEN, Heico Sandee for his valuable feedback and availability for support and Arjan Verboord, for his daily guidance and extensive reviewing of this thesis.

Many thanks to all the members of the team that have spent three quarters of a year on developing and building the Ball-Balancing Robot. The team work was great and the team spirit made me enjoy this project a lot.

Koos van der Blonk
Eindhoven, July 2014

CONTENTS

1	Introduction	1
1.1	Motivation	1
1.2	Description of the BBR	2
1.3	Goal	2
1.4	History of BBR's	2
1.5	Structure of the report	3
2	2D Model	7
2.1	Assumptions	7
2.2	Description of 2D model	8
2.3	Coordinates	11
2.4	Equations of motion for YZ/XZ-plane	12
2.5	Equations of motion for XY-plane	18
2.6	Torque conversion	21
2.7	Calculation of parameters	25
2.8	Simulation of 2D model	32
3	3D Model	37
3.1	Assumptions	37
3.2	Description of 3D-model	37
3.3	Coordinates	37
3.4	Variables	40
3.5	Parameters	45
3.6	Equations of motion	48
3.7	Simulation of 3D model	51
4	Design of a controller	55
4.1	Literature review	55
4.2	Control approach	57
4.3	Design of a LQR controller	59
4.4	Design of a SISO controller	68
4.5	Derivation of the ball's position	89
4.6	Conclusion	90
5	Controller implementation	93
5.1	LQR controller for balancing	94
5.2	LQR controller for yaw control	104
5.3	LQR controller for position control	104
5.4	Conclusion	104
6	System stability research	109
6.1	Research approach	109
6.2	Sensitivity analysis with LQR controller	111
6.3	Sensitivity analysis with SISO controller	116
6.4	Conclusion	124

7 Conclusion & Recommendations	127
7.1 Conclusion	127
7.2 Recommendations & Future work	128
Bibliography	132
A Parameters	135
B Derivations	137
B.1 Derivations 2D model	137

LIST OF FIGURES

1.1	The BBR as demonstrator on a fair.	1
1.2	Sketch of the BBR and the relevant outputs.	2
1.3	Overview of different BBR's.	4
2.1	Modeling the 3D system with three 2D models.	9
2.2	Sketches of the 2D models.	10
2.3	Sketch of ball torque and body torque, generated by the virtual motor torque, the tangential forces and corresponding levers in yz -plane.	15
2.4	Sketch of body torque, generated by the virtual motor torque, the tangential force and corresponding lever in xy -plane.	20
2.5	Sketch of body torques, generated by the real motor torques, the tangential forces and the corresponding levers.	21
2.6	Sketch of body torques, generated by the virtual motor torques, the tangential forces and the corresponding levers.	22
2.7	Illustration of the method used to estimate the moment of inertia of the body about the rotation axis.	27
2.8	Sketch of the decomposed angular rates of the omni-wheels.	29
2.9	Free response of the linear and nonlinear 2D model for the yz -plane with an initial pitch angle of 0.1°	34
2.10	Forced response of the linear and nonlinear 2D model for the yz -plane with a constant external torque of 0.01 Nm.	35
2.11	Forced response of the 2D model for the xy -plane with a constant external torque of 0.3 Nm	35
3.1	Sketch of the 3D model.	38
3.2	Sketch of the different coordinate frames.	39
3.3	Twist of the ball.	42
3.4	Sketch of the position of the omni-wheels.	45
3.5	Free response of the linear and nonlinear 3D model with an initial pitch angle of 0.1°	52
3.6	Forced response of the linear and nonlinear 3D model with a constant external torque of 0.01 Nm.	53
4.1	The closed loop system with a LQR controller.	60
4.2	The inverse step response of the pitch angle simulated for different values of ρ , that determines the ratio between optimizing the control effort and optimizing the states.	62
4.3	Bode magnitude plot of the sensitivity functions of the minimal coordinates (in case $\rho = 1$).	63
4.4	Bode magnitude plot of the sensitivity functions of the minimal coordinates (in case $\rho = 0.25$).	64
4.5	Bode magnitude plot of the closed loop transfer functions of the minimal coordinates (in case $\rho = 0.25$).	65
4.6	The required torque in case of a natural disturbance signal with a frequency of 0.5 Hz and an amplitude of 5°	66
4.7	The required torque in case of a exceptional disturbance signal with a frequency of 0.5 Hz and an amplitude of 15°	66

4.8	Simulation of position of the BBR from position (0,0) to (3,1) with the LQR controller.	67
4.9	Sketch of the closed loop system with a SISO controller.	68
4.10	Bode plot of plant transfer function $P_{\psi_x}(s)$	69
4.11	Bode plot of closed loop transfer function $T_{\psi_x}(s)$	70
4.12	Simulation for different values of n , that sets the frequencies of the pole and the zero of the lead compensator. a) The impulse response of the closed loop transfer function $T_{\psi_x}(s)$. b) Bode plot of the sensitivity function $S_{\psi_x}(s)$	71
4.13	Bode magnitude plot of the sensitivity function $S_{\psi_x}(s)$	72
4.14	Bode magnitude plot of closed loop transfer function $T_{\psi_x}(s)$	73
4.15	Bode plot of plant transfer function $P_{\psi_z}(s)$	74
4.16	Bode magnitude plot of the sensitivity function $S_{\psi_z}(s)$ with a bandwidth of 3 Hz.	74
4.17	Bode magnitude plot of the sensitivity function $S_{\psi_z}(s)$ with a bandwidth of 9 Hz.	75
4.18	Bode magnitude plot of closed loop transfer function $T_{\psi_z}(s)$	76
4.19	Sketch of the closed loop system for x_S	77
4.20	Bode plot of plant transfer function $P_{x_S}(s)$	77
4.21	Bode magnitude plot of plant transfer function $P_{x_S}(s)$, zoomed in around the crossing of the 0 dB line.	79
4.22	Bode plot of the open loop transfer function $L_{x_S}(s)$	79
4.23	Simulations for different values of ω_z . a) Bode plot of the open loop transfer function $L_{x_S}(s)$. b) The step response of the closed loop transfer function $T_{x_S}(s)$	80
4.24	The step response of the closed loop transfer function $T_{x_S}(s)$ for three different feedback controllers.	82
4.25	The step response versus the response to a 4 th order reference signal of the closed loop transfer function with controller $K1_{x_S}$	82
4.26	Sketch of the closed loop system including a 4 th order reference signal and feedforward control.	83
4.27	The x -position of the ball moving from x -position 0 to 1, including a 4 th order reference signal and feedforward control, displayed for feedforward gains from 0.1 until 0.5.	83
4.28	The x -position of the ball moving from x -position 0 to 1, including a 4 th order reference signal and feedforward control, displayed for feedforward gains from 0.2 until 0.3.	84
4.29	Comparison of the response of the system in moving from x -position 0 to 1, with feedback controllers $K1_{x_S}$, $K2_{x_S}$ and $K3_{x_S}$	85
4.30	The inverse step response of the pitch angle.	86
4.31	The required torque in case of a natural disturbance signal with a frequency of 0.5 Hz and an amplitude of 5°.	87
4.32	The required torque in case of a exceptional disturbance signal with a frequency of 0.5 Hz and an amplitude of 15°.	87
4.33	Simulation of position of the BBR from position (0,0) to (3,1) with the SISO controller.	88
5.1	The BBR attached to the moving test frame.	93
5.2	Sketch of the working of a complementary filter.	94
5.3	The angular rates of the omni-wheels.	95
5.4	The torques generated by the LQR controller.	96
5.5	a) The pitch and roll angle. b) The pitch and roll angular rates.	97
5.6	The magnitude spectra of the pitch and roll angular rates of the tests, where the BBR is hanging in the moving test frame.	98

5.7	The pitch and roll angular rates of the test, where the BBR is hanging in the moving test frame, with the motors switched on.	98
5.8	The omni-wheels, where the gap between the rollers is clearly visible.	99
5.9	The magnitude spectra of the pitch and roll angular rates of the test, where the BBR is trying to balance.	100
5.10	The magnitude spectra of the torques of the test, where the BBR is trying to balance.	100
5.11	The magnitude spectra of the torques of the test, where the BBR is trying to balance after the foam is added.	101
5.12	The magnitude spectra of the torques of the test, where the BBR is trying to balance after averaging the gyroscope data.	102
5.13	Bode magnitude plot of the 4 th order FIR filter, used to filter the torques.	102
5.14	The unfiltered and filtered torque of motor one.	103
5.15	The magnitude spectra of the torques of a test, after the FIR filter is implemented.	103
5.16	The pitch, roll and yaw angle in case: a) Yaw control is applied. b) Yaw control is not applied.	105
6.1	LQR controller: The phase margin for different values of the moment of inertia. a) I_S . b) $I_{B,x}$	112
6.2	LQR controller: Stability of the system for uncertainties in model parameters m_{B1} and l	113
6.3	LQR controller: The phase margin for different values of m_{B1} and l	114
6.4	LQR controller: Required torque for the inverse step response of the roll angle with $m_{B1} = 30$ kg and $l = 85$ cm.	115
6.5	LQR controller: The phase margin for different values of m_S	115
6.6	LQR controller: Required torque for the inverse step response of the roll angle with $m_S = 5$ kg.	116
6.7	LQR controller: The phase margin for different values of α	117
6.8	LQR controller: Required torque for the inverse step response of the roll angle with $\alpha = 80^\circ$	117
6.9	SISO controller: The phase margin for different values of the moment of inertia. a) I_S . b) $I_{B,x}$	119
6.10	SISO controller: Stability of the system for uncertainties in plant parameters m_{B1} and l	120
6.11	SISO controller: The phase margin for different values of m_{B1} and l	120
6.12	SISO controller: Required torque for the inverse step response of the roll angle with $m_{B1} = 30$ kg and $l = 95$ cm.	121
6.13	SISO controller: The phase margin for different values of m_S	122
6.14	SISO controller: Required torque for the inverse step response of the roll angle with $m_S = 5$ kg.	122
6.15	SISO controller: The phase margin for different values of α	123
6.16	SISO controller: Required torque for the inverse step response of the roll angle with $\alpha = 80^\circ$	123

LIST OF TABLES

2.1	Description of parameters used in sketches of 2D models.	11
-----	--	----

2.2	Description and values of parameters used for the estimation of the moment of inertia of the body.	27
3.1	Description and notation of variables and parameters used in the 3D model. . . .	41
6.1	Description of the uncertain model parameters with their nominal value.	110
A.1	Model parameters.	135
A.2	Model variables.	136

LIST OF ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
BBR	Ball-Balancing Robot
CMU	Carnegie Mellon University
COM	Centre Of Mass
DC	Direct Current
DOF	Degrees Of Freedom
ETH	Swiss Federal Institute of Technology
FF	FeedForward
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
IMU	Inertia Measurement Unit
LQR	Linear Quadratic Regulator
MIMO	Multiple-Input-Multiple-Output
PD	Proportional-Derivative
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
RHP	Right-Half Plane
SISO	Single-Input-Single-Output
TGU	Tohoku Gakuin University
UA	University of Adelaide

1. INTRODUCTION

1.1 Motivation

Traditional wheeled mobile robots are equipped with two independent driving wheels. Since these robots have two degrees of freedom (DOFs), they can rotate about any point, but cannot perform immediate motion in every direction. To overcome this type of motion limitation, omni-directional mobile robots were proposed. They can move in an arbitrary direction without changing the direction of the wheels, because they can achieve three DOF-motion on a two-dimensional plane. A special kind of such a robot is a Ball-Balancing Robot (BBR). This system is inherently unstable, as it will immediately fall down when no active control is applied to the wheels on the ball.

ALTEN Mechatronics is developing such a robot for promotional purposes. The robot shall be a dynamically stable robot, designed to balance on a ball and to move in every direction. It shall also be designed to be robust to disturbances and to navigate autonomously. In the future, more intelligence will be added, such as following and approaching people.



Figure 1.1: The BBR as demonstrator on a fair.

1.2 Description of the BBR

The BBR is a robot on top of a ball, that drives the ball using three actuators. The BBR is balancing by measuring its tilt angle and calculating the appropriate motor torques needed to keep the BBR upright. The ball is a medicine ball and the actuators are three 2-row omni-wheels driven by brushless DC motors. The body of the robot includes the battery at the top, the Inertia Measurement Unit (IMU) that measures the angles and angular rates of the body of the robot with a sample rate of 200 Hz, see Fig. 1.2 and a microprocessor that receives data from the IMU to control the actuators. Furthermore, the body includes the motor encoders that measure the angles of the omni-wheels (see also Fig. 1.2), which are needed to derive the position of the ball for position control of the BBR.

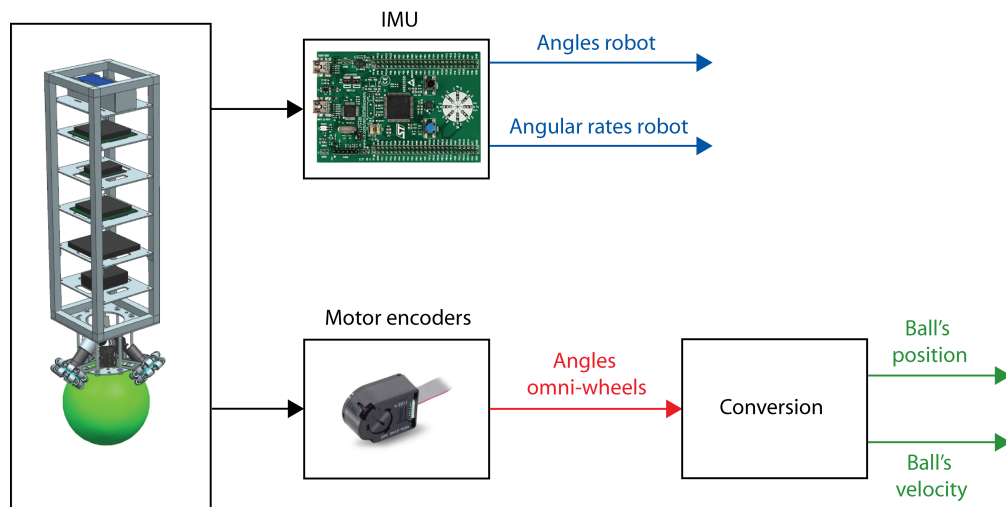


Figure 1.2: Sketch of the BBR and the relevant outputs.

1.3 Goal

The goal of the internship project is to develop a model of the BBR's dynamic behaviour. The goal of the master thesis is to design and implement a controller that stabilizes the system, such that the BBR is able to balance with tilt angles of at most 10° and is able to track given position set points. Furthermore, research will be done to investigate under what requirements the system will remain stable with the developed controller.

1.4 History of BBR's

Several BBR's have already been developed. On the following BBR's quite some research is done:

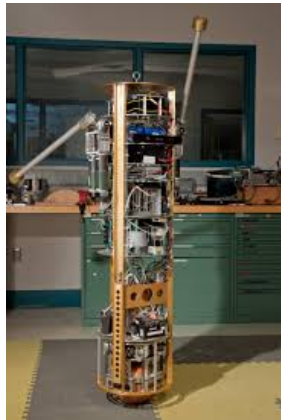
1. The first BBR (shown in Fig. 1.3a)) is developed in 2006 at Carnegie Mellon University (CMU) in the United States [8]. The robot has about human size with the aim to let it interact with humans. Later also arms were added to the BBR [17]. Special about this

BBR is that it has three legs for static stability and that it has a drive mechanism that consists of four rollers: two active rollers, driven by DC motors, that actuate the ball and two spring-loaded passive rollers opposite the drive rollers, that apply a force to the ball to maintain contact between the drive rollers and the ball. That implies that the robot cannot rotate around the vertical axis. Later, the two passive rollers were replaced by two active rollers, due to unequal friction for forward and backward motion and also a drive system was added for rotation around the vertical axis [15]. This means that in total 5 DC motors were needed, which is quite expensive. A lot of research has been done on this BBR, which can be found in [8],[15],[13],[16].

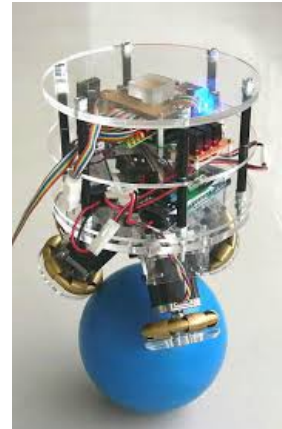
2. The Tohoku Gakuin University (TGU) in Japan developed a BBR (shown in Fig. 1.3b)) in 2008 [7],[6]. This one is small compared to the BBR of CMU, but special about this BBR is that it can perform the same motion (including rotation around the vertical axis) with only three motors connected to three omni-wheels that drive the ball. Also this robot can carry loads of at least 10 kg.
3. The University of Adelaide (UA) in Australia developed a BBR using LEGO Mindstorms NXT (shown in Fig. 1.3c)) in 2009 [3]. It is a small robot of about 20 cm high, completely built of LEGO. It has only two wheels to drive the ball.
4. The National Chung Hsing University in Taiwan developed a BBR (shown in Fig. 1.3d)) in 2012, similar to the one of the TGU [19]. It also has three omni-wheels and is of about the same height.
5. ETH Zürich in Switzerland developed a BBR, called Rezero (shown in Fig. 1.3e)), in 2010. Like the robot of TGU, it has three omni-wheels to drive the ball. It has a high dynamic robustness, achieves a linear speed of up to 2 m/s and a tilt angle of 20° [2].f

1.5 Structure of the report

The report consists of seven chapters, including this chapter. In Chap. 2, a two-dimensional (2D) dynamical model of the BBR is derived. Due to shortcomings of the 2D model, a three-dimensional (3D) dynamical model is derived in Chap. 3. In Chap. 4, a controller for the BBR is designed, which stabilizes the system, such that the BBR is able to track given position set points. The implementation of this controller is described in Chap. 5. In Chap. 6, research is done to investigate under what requirements the system will remain stable with the developed controller. Finally, conclusions and recommendations can be found in Chap. 7.



a) BBR CMU.



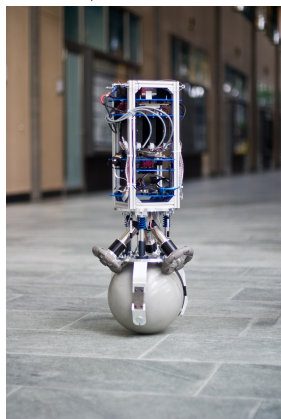
b) BBR TGU.



c) BBR UA.



d) BBR Taiwan.



e) BBR Rezero.

Figure 1.3: Overview of different BBR's.

2. 2D MODEL

A dynamical model is required to get a deeper insight in the system. Also to control the BBR, equations of motion are required, which can be derived using a dynamical model of the BBR. The aim of this chapter is to model the 3D system as three 2D models and to verify if the combination of 2D models is representative for the three-dimensional system.

Different 2D models for a BBR are already derived by [3], [2], [15]. However, it is still useful to derive the 2D models to get a qualitative understanding of the dynamics of the BBR.

In the first paragraph assumptions will be made. In the second paragraph a description of the 2D models is given. After that coordinates will be defined in the third paragraph. In the fourth paragraph the equations of motion for the vertical planes will be derived. The equations of motion for the horizontal plane will be derived in the fifth paragraph. In the sixth paragraph the relation between the torques of the virtual motors and the torques of the real motors will be derived. Several parameters of the model, including the moments of inertia, are estimated in the seventh paragraph. Finally, in the eighth paragraph, simulations with the 2D models will be done.

The values of the parameters used in this chapter can be found in Appendix A. Extensive derivations of equations in this chapter can be found in Appendix B.1.

2.1 Assumptions

Before a model of the BBR can be made, some assumptions will be made in this paragraph.

Independent vertical planes

When modeling the two-dimensional planes, the vertical planes are assumed to be independent.

Rigid bodies/floor

The total system is assumed to consist of two rigid bodies, namely the ball, the body of the robot with the drive system and the omni-wheels attached to it. The assumption of rigid bodies neglects deformation of the bodies. Furthermore, also deformation of the floor is neglected.

Friction

It is assumed that, besides static friction, all other types of friction, like rolling and kinetic friction, are negligible.

No slip

It is assumed that there is no slip between the ball and the floor and between the ball and the omni-wheels. This assumption implicates that

- The applied torques of the motors are restricted to the range in which there occurs no slip between the ball and the omni-wheels.
- It has to be guaranteed that the static friction is high enough, i.e. that the static friction never will be overcome, because then slipping will happen. Particularly, high static friction that prevents the ball to rotate around the vertical axis has to be guaranteed by limiting the torque around the vertical axis.
- The ball is always in contact with the floor, so no jumping occurs. Also the floor is rough enough to prevent slipping and it has to be taken care off that there are no obstacles present that cause the ball to slip.

Horizontal floor

It is assumed that the floor, on which the BBR moves, is horizontal, which implicates that the ball has no potential energy.

Negligible time delay

It is assumed that the time delay between the measurements of the sensors and the control of the actuators is negligible.

Omni-wheels

It is assumed that the 2-row omni-wheels, which have more than one contact point, can be modeled as 1-row omni-wheels that have a single contact point.

2.2 Description of 2D model

Fig. 2.1 shows the cross section of the 3D system.

In order to make adequate 2D models of the 3D system, the following modeling choices are made:

- The omni-wheels and the motors are modeled as virtual actuating wheels that include the motors, like is done in [2]. Each 2D model contains one virtual actuating wheel that rotates around the axis orthogonal to that plane. The relation between the torques of the real and virtual system will be derived later in section Sec. 2.6.
- The 2D models of the different planes only model 2D motion that is described by the generalized coordinates of that plane. So the 2D models are not a projection of the 3D motion in 2D spaces, but they describe the 2D motion in 2D spaces. For example, the 2D model of the xy -plane does not describe the translational motion of the ball, but only the rotational motion around the z -axis of the body.

The next modeling choices only hold for the model of the vertical planes:

- The body is modeled as a solid cuboid. Obviously the mass of the body is not uniformly distributed. This is modeled by dividing the body into an upper and lower part that both have a uniform mass distribution, but a different density. This makes it possible to model a

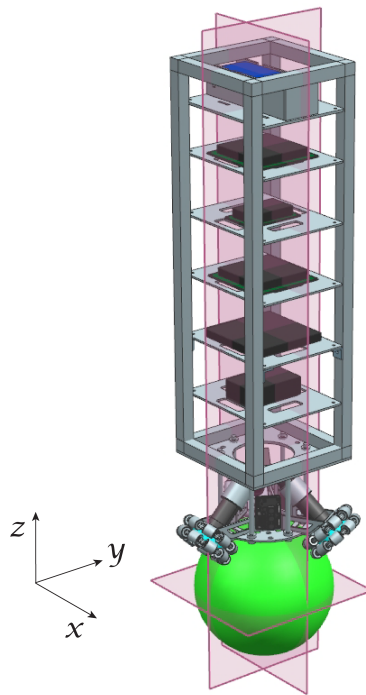


Figure 2.1: Modeling the 3D system with three 2D models.

centre of mass (COM) that is not exactly in the middle of the body. This modeling choice doesn't have influence on the model for the xy -plane.

- The ball is modeled as a disk that only rotates around the axis orthogonal to the plane. The COM of the body is attached to the rotation axis of the disk with a rigid rod. The virtual actuating wheel is also attached to this rod. The body can rotate around the rotation axis of the disk independent of the rotation of the disk.

The next modeling choice only holds for the horizontal plane:

- In the horizontal plane, the ball is modeled as a disk with radius r_S . The virtual actuating wheel is connected to the vertical axis of the disk with a rod and the body is attached to the rod, such that the COM of the body has the same x - and y -coordinates as the centre of the disk.

The 2D models are sketched in Fig. 2.2, one model for the yz -plane and one model for the xy -plane. The model for the xz -plane is identical to the model for the yz -plane, so only the model for the yz -plane will be treated here and the equations of motion for the yz -plane can be easily converted to the equations of motion for the xz -plane. A description of the parameters, used in Fig. 2.2, can be found in Table 2.1.

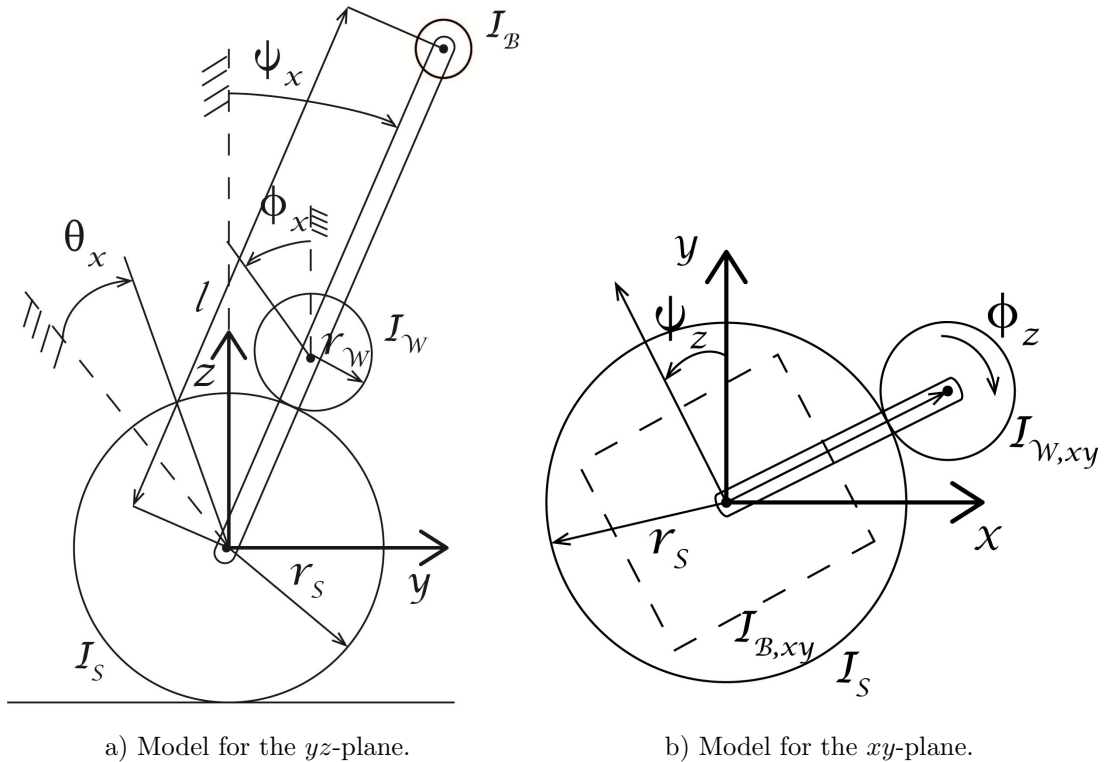


Figure 2.2: Sketches of the 2D models.

Table 2.1: Description of parameters used in sketches of 2D models.

Parameter	Description
r_S	Radius of the ball
r_W	Radius of virtual actuating wheel
I_S	Moment of inertia of the ball
I_W	Moment of inertia of the virtual actuating wheel in the yz -/ xz -plane
$I_{W,xy}$	Moment of inertia of the virtual actuating wheel in the xy -plane
I_B	Moment of inertia of the body of the robot in the yz -/ xz -plane
$I_{B,xy}$	Moment of inertia of the body of the robot in the xy -plane
l	Distance between COM of the ball and COM of the body of the robot

2.3 Coordinates

To derive the equations of motion, coordinates need to be defined. The coordinates are defined like they are shown for the yz -plane and the xy -plane in Fig. 2.2, where θ_x and θ_y indicate the orientation of the ball, ψ_x , ψ_y and ψ_z indicate the orientation of the body and ϕ_x , ϕ_y and ϕ_z indicate the orientation of the virtual actuating wheels. From θ_x and θ_y the translation of the ball along the x - and y -axis can be easily derived.

The vertical planes have both two DOFs, namely the rotation/translation of the ball and the rotation of the body. The horizontal plane has only one DOF, namely the rotation of the body, because it is assumed that the ball does not rotate around the vertical axis, see Sec. 2.1. Therefore the minimal coordinates for the three different planes are defined as

$$\mathbf{q}_{yz} = \begin{bmatrix} \theta_x \\ \psi_x \end{bmatrix}, \mathbf{q}_{xz} = \begin{bmatrix} \theta_y \\ \psi_y \end{bmatrix}, \mathbf{q}_{xy} = [\psi_z]. \quad (2.1)$$

2.3.1 Cartesian coordinates yz -plane

Now the minimal coordinates are defined, the positions, expressed in cartesian coordinates, of the ball (denoted by y_S and z_S), the body (denoted by y_B and z_B) and the virtual actuating wheel (denoted by y_W and z_W) can be written as functions of these minimal coordinates. The coordinate frame is chosen such that its origin is on the same height as the centre of the ball, see Fig. 2.2.

$$\begin{aligned} \begin{bmatrix} y_S \\ z_S \end{bmatrix} &= \begin{bmatrix} r_S \theta_x \\ 0 \end{bmatrix} \\ \begin{bmatrix} y_B \\ z_B \end{bmatrix} &= \begin{bmatrix} r_S \theta_x + l \sin(\psi_x) \\ l \cos(\psi_x) \end{bmatrix} \\ \begin{bmatrix} y_W \\ z_W \end{bmatrix} &= \begin{bmatrix} r_S \theta_x + (r_S + r_W) \sin(\psi_x) \\ (r_S + r_W) \cos(\psi_x) \end{bmatrix}. \end{aligned} \quad (2.2)$$

Those coordinates are needed for calculating the equations of motion for the yz -plane. A description of the used parameters can be found in Table 2.1.

2.3.2 Cartesian coordinates xy -plane

In the xy -plane, the coordinate frame is chosen such that its origin is located in the centre of the ball. The only minimal coordinate is ψ_z , so only motion that depends on ψ_z is taken into account in the 2D model for the xy -plane. That means that only the motion of the virtual actuating wheel and the body is of importance for the equations of motion. The body only rotates and doesn't move in the xy -plane, which implies that only the position of the virtual actuating wheel needs to be determined. The position of the virtual actuating wheel is expressed in cartesian coordinates and denoted by $x_{W,xy}$ and $y_{W,xy}$. Writing $x_{W,xy}$ and $y_{W,xy}$ as functions of the minimal coordinate ψ_z yields

$$\begin{bmatrix} x_{W,xy} \\ z_{W,xy} \end{bmatrix} = \begin{bmatrix} (r_S + r_W) \cos(\psi_z) \\ (r_S + r_W) \sin(\psi_z) \end{bmatrix}. \quad (2.3)$$

Those coordinates are needed for calculating the equations of motion for the xy -plane.

2.4 Equations of motion for YZ/XZ-plane

In this section the equations of motion for the yz -plane will be derived. In the same way, equations of motion for the xz -plane can be derived.

2.4.1 Lagrangian method

The equations of motion are derived using the Lagrangian method. This method consists of the following steps:

1. Express the kinetic energy (denoted with T) and potential energy (denoted with V) of all rigid bodies as functions of the minimal coordinates.
2. Express all external (or non-potential) torques (denoted as τ_{ext}) as functions of the minimal coordinates.
3. Define the Lagrangian L as $L(\mathbf{q}, \dot{\mathbf{q}}) = T - V$ (where \mathbf{q} is the vector of minimal coordinates).
4. Calculate the Euler-Lagrange equations $\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_{ext,i}$ with $i = 1, \dots, n$ where n is the length of the vector \mathbf{q} .

This will result in the equations of motion.

2.4.2 Kinetic and potential energy of the system

Energy ball

The kinetic energy of the ball is defined as the sum of the translational and rotational kinetic energy:

$$T_{S,yz} = \underbrace{\frac{1}{2} m_S v_{S,yz}^T \cdot v_{S,yz}}_{\text{Translation}} + \underbrace{\frac{1}{2} I_S \dot{\theta}_x^2}_{\text{Rotation}} \quad (2.4)$$

where m_S denotes the mass of the ball and where $v_{S,yz}$ denotes the velocity of the ball and $v_{S,yz}^T \cdot v_{S,yz}$ is defined as

$$\begin{aligned} v_{S,yz}^T \cdot v_{S,yz} &= |v_{S,yz}|^2 \\ &= \dot{y}_S^2 + \dot{z}_S^2 \\ &= r_S^2 \dot{\theta}_x^2. \end{aligned}$$

Substituting $v_{S,yz}^T \cdot v_{S,yz}$ in Eq. (2.4) yields the equation for the kinetic energy of the ball

$$T_{S,yz} = \frac{1}{2} m_S r_S^2 \dot{\theta}_x^2 + \frac{1}{2} I_S \dot{\theta}_x^2. \quad (2.5)$$

The potential energy of the ball is zero, because it is assumed that the ball only moves over horizontal surfaces:

$$V_{S,yz} = 0. \quad (2.6)$$

Energy body

Analogous, the kinetic energy of the body is defined as

$$T_{B,yz} = \frac{1}{2} m_B v_{B,yz}^T \cdot v_{B,yz} + \frac{1}{2} I_B \dot{\psi}_x^2$$

where m_B denotes the total mass of the body and where $v_{B,yz}^T \cdot v_{B,yz}$ is defined as

$$\begin{aligned} v_{B,yz}^T \cdot v_{B,yz} &= \dot{y}_B^2 + \dot{z}_B^2 \\ &= r_S^2 \dot{\theta}_x^2 + 2r_S l \dot{\theta}_x \dot{\psi}_x \cos \psi_x + l^2 \dot{\psi}_x^2. \end{aligned}$$

(The derivation of $v_{B,yz}^T \cdot v_{B,yz}$ can be found in Appendix B.1).

Analogous to the calculation of the kinetic energy of the ball and the virtual actuating wheel, this yields the equation for $T_{B,yz}$

$$\begin{aligned} T_{B,yz} &= \frac{1}{2} m_B (r_S^2 \dot{\theta}_x^2 + 2r_S l \dot{\theta}_x \dot{\psi}_x \cos \psi_x + l^2 \dot{\psi}_x^2) + \frac{1}{2} I_B \dot{\psi}_x^2 \\ &= \frac{1}{2} m_B (r_S^2 \dot{\theta}_x^2 + 2r_S l \dot{\theta}_x \dot{\psi}_x \cos \psi_x) + \frac{1}{2} \underbrace{(I_B + m_B l^2)}_{I'_B} \dot{\psi}_x^2. \end{aligned} \quad (2.7)$$

I_B is the moment of inertia of the body about the axis going through its COM. According to the Parallel Axis Theorem¹, I'_B denotes the moment of inertia of the body about the axis the body rotates, namely the axis through the centre of the wheel. I'_B will be estimated in Sec. 2.7.

The potential energy of the body of the robot is defined as

$$V_{B,yz} = M_B g l \cos(\psi_x). \quad (2.8)$$

¹The Parallel Axis Theorem states that if a body is rotating around a new axis that is parallel to the original axis that goes through the COM, with a distance d between the two axes, then the moment of inertia about the new axis is related to the moment of inertia about the original axis (denoted by I_{COM}) by $I = I_{COM} + md^2$, where m denotes the total mass of the body [12].

Energy virtual actuating wheel

Analogous to the kinetic energy of the ball, the kinetic energy of the virtual actuating wheel is defined as

$$T_{W,yz} = \frac{1}{2}m_W v_{W,yz}^T \cdot v_{W,yz} + \frac{1}{2}I_W \dot{\phi}_x^2$$

where m_W denotes the mass of the virtual actuating wheel and where $v_{W,yz}^T \cdot v_{W,yz}$ is defined as

$$\begin{aligned} v_{W,yz}^T \cdot v_{W,yz} &= \dot{y}_W^2 + \dot{z}_W^2 \\ &= r_S^2 \dot{\theta}_x^2 + 2r_S(r_S + r_W)\dot{\theta}_x\dot{\psi}_x \cos \psi_x + (r_S + r_W)^2 \dot{\psi}_x^2 \end{aligned}$$

and where $\dot{\phi}_x$ is defined as

$$\dot{\phi}_x = \frac{r_S}{r_W}(\dot{\theta}_x - \dot{\psi}_x). \quad (2.9)$$

(The derivation of $v_{W,yz}^T \cdot v_{W,yz}$ and $\dot{\phi}_x$ can be found in Appendix B.1).

This yields the equation for the kinetic energy of the virtual actuating wheel

$$\begin{aligned} T_{W,yz} &= \frac{1}{2}m_W \left(r_S^2 \dot{\theta}_x^2 + 2r_S(r_S + r_W)\dot{\theta}_x\dot{\psi}_x \cos \psi_x + (r_S + r_W)^2 \dot{\psi}_x^2 \right) \\ &\quad + \frac{1}{2}I_W \left(\frac{r_S}{r_W}(\dot{\theta}_x - \dot{\psi}_x) \right)^2. \end{aligned} \quad (2.10)$$

The potential energy of the virtual actuating wheel is defined as

$$V_{W,yz} = m_W g (r_S + r_W) \cos(\psi_x). \quad (2.11)$$

2.4.3 External torques

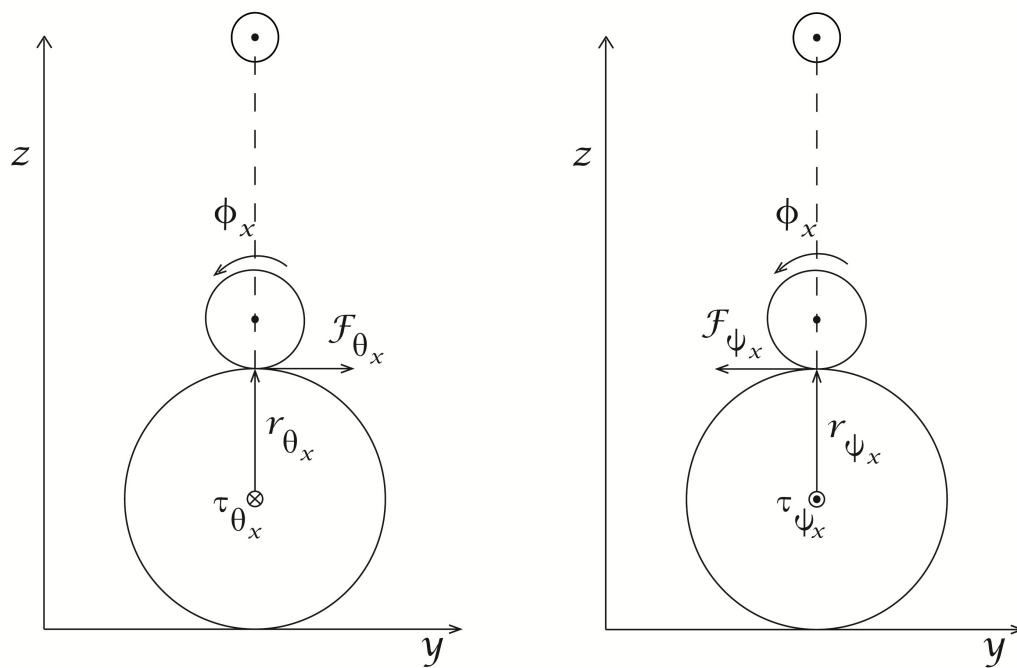
The only external torque of the system is the torque of the virtual motor. The torque of the virtual motor transfers a torque to the virtual actuating wheel, denoted by $\tau_{VW,yz}$. $\|\tau_{VW,yz}\|$ is the input of the model and because $\tau_{VW,yz}$ only has a component in the x -direction (denoted by τ_x), the input is $\|\tau_{VW,yz}\| = \tau_x$. A relation between the input torque and the torques that directly controls the minimal coordinates, which form the external torques of the model, will be derived in this section.

In Fig. 2.3 the external torques of the model are shown together with the forces that generate these torques and the levers in case $\theta_x = \psi_x = 0$.

Using the relation between $\dot{\phi}_x$ and the rates of the minimal coordinates $\dot{\theta}_x$ and $\dot{\psi}_x$, the geometric Jacobian (denoted by J), which maps joint velocities to ‘end-effector’ velocities, can be derived. This transpose of this Jacobian can be used to derive a relation between the input torque τ_x and the torques that directly control the minimal coordinates.

The relation between $\dot{\phi}_x$ and the joint velocities $\dot{\theta}_x$ and $\dot{\psi}_x$ is given by

$$\begin{aligned} \dot{\phi}_x &= \frac{r_S}{r_W}(\dot{\theta}_x - \dot{\psi}_x) \\ &= \underbrace{\begin{bmatrix} \frac{r_S}{r_W} & -\frac{r_S}{r_W} \end{bmatrix}}_J \begin{bmatrix} \dot{\theta}_x \\ \dot{\psi}_x \end{bmatrix} \end{aligned} \quad (2.12)$$



(a) Torque of the ball.

(b) Torque of the body.

Figure 2.3: Sketch of ball torque and body torque, generated by the virtual motor torque, the tangential forces and corresponding levers in yz -plane.

The relation between the input torque τ_x and the torques that directly control the minimal coordinates is thus given by

$$\begin{aligned}\boldsymbol{\tau}_{ext} &= \begin{bmatrix} \tau_{\theta_x} \\ \tau_{\psi_x} \end{bmatrix} \\ &= \mathbf{J}^T \tau_x \\ &= \begin{bmatrix} r_s \\ r_w \\ -\frac{r_s}{r_w} \end{bmatrix} \tau_x.\end{aligned}\tag{2.13}$$

2.4.4 Lagrangian and Euler-Lagrange equations

Define the Lagrangian as

$$L(\theta_x, \psi_x, \dot{\theta}_x, \dot{\psi}_x) = T_{S,yz} + T_{B,yz} + T_{W,yz} - V_{S,yz} - V_{B,yz} - V_{W,yz}.\tag{2.14}$$

Calculating the Euler-Lagrange equations

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_x} \right) - \frac{\partial L}{\partial \theta_x} = \tau_{\theta_x}$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}_x} \right) - \frac{\partial L}{\partial \psi_x} = \tau_{\psi_x}$$

and ordering them results into the following equations of motion:

$$\mathbf{M}(\mathbf{q}_{yz}) \ddot{\mathbf{q}}_{yz} + \mathbf{C}(\mathbf{q}_{yz}, \dot{\mathbf{q}}_{yz}) \dot{\mathbf{q}}_{yz} + \mathbf{G}(\mathbf{q}_{yz}) = \boldsymbol{\tau}_{ext}\tag{2.15}$$

where the first term represents the inertial forces due to accelerations, the second term represents the Coriolis and centrifugal forces, the third term represents the gravitational forces and the right-hand side represents the external torques.

The matrices $\mathbf{M}(\mathbf{q}_{yz})$, $\mathbf{C}(\mathbf{q}_{yz}, \dot{\mathbf{q}}_{yz})$ and $\mathbf{G}(\mathbf{q}_{yz})$ are defined as

$$\mathbf{M}(\mathbf{q}_{yz}) = \begin{bmatrix} I_S + r_S^2 m_{tot} + \frac{r_S^2}{r_W^2} I_W & r_S \lambda \cos(\psi_x) - \frac{r_S^2}{r_W^2} I_W \\ r_S \lambda \cos(\psi_x) - \frac{r_S^2}{r_W^2} I_W & r_{tot}^2 m_W + \frac{r_S^2}{r_W^2} I_W + I_B' \end{bmatrix}\tag{2.16}$$

$$\mathbf{C}(\mathbf{q}_{yz}, \dot{\mathbf{q}}_{yz}) = \begin{bmatrix} 0 & -r_S \lambda \dot{\psi}_x \sin(\psi_x) \\ 0 & 0 \end{bmatrix}\tag{2.17}$$

$$\mathbf{G}(\mathbf{q}_{yz}) = \begin{bmatrix} 0 \\ -\lambda g \sin(\psi_x) \end{bmatrix}\tag{2.18}$$

where

$$\begin{aligned}m_{tot} &= m_S + m_W + m_B \\ r_{tot} &= r_S + r_B \\ \lambda &= m_W(r_S + r_W) + m_B l.\end{aligned}$$

2.4.5 Linearization

In the previous paragraph the equations of motion are derived using the Lagrangian method. For controlling purposes, the equations of motion will be linearized in this paragraph.

The state vector \mathbf{x} (for the yz -plane) is defined as

$$\mathbf{x} = \begin{bmatrix} \mathbf{q}_{yz} \\ \dot{\mathbf{q}}_{yz} \end{bmatrix} = \begin{bmatrix} \theta_x \\ \psi_x \\ \dot{\theta}_x \\ \dot{\psi}_x \end{bmatrix} \quad (2.19)$$

whereas the input of the system is defined as $u = \tau_x$.

The system will be linearized around the unstable equilibrium point, where all state variables and the input are zero, i.e.

$$\bar{\mathbf{x}} = \begin{bmatrix} \bar{\theta}_x \\ \bar{\psi}_x \\ \bar{\dot{\theta}}_x \\ \bar{\dot{\psi}}_x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.20)$$

and

$$\bar{u} = 0. \quad (2.21)$$

The linearization results in the linear state space representation, defined as

$$\begin{aligned} \dot{\mathbf{x}} &= A \cdot \mathbf{x} + B \cdot u \\ \mathbf{y} &= C \cdot \mathbf{x} \end{aligned} \quad (2.22)$$

where the matrices A , B and C are defined as

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \left. \frac{\partial \ddot{\theta}_x}{\partial \theta_x} \right|_{(x,u)=(\bar{x},\bar{u})} & \left. \frac{\partial \ddot{\theta}_x}{\partial \psi_x} \right|_{(x,u)=(\bar{x},\bar{u})} & \left. \frac{\partial \ddot{\theta}_x}{\partial \dot{\theta}_x} \right|_{(x,u)=(\bar{x},\bar{u})} & \left. \frac{\partial \ddot{\theta}_x}{\partial \dot{\psi}_x} \right|_{(x,u)=(\bar{x},\bar{u})} \\ \left. \frac{\partial \ddot{\psi}_x}{\partial \theta_x} \right|_{(x,u)=(\bar{x},\bar{u})} & \left. \frac{\partial \ddot{\psi}_x}{\partial \psi_x} \right|_{(x,u)=(\bar{x},\bar{u})} & \left. \frac{\partial \ddot{\psi}_x}{\partial \dot{\theta}_x} \right|_{(x,u)=(\bar{x},\bar{u})} & \left. \frac{\partial \ddot{\psi}_x}{\partial \dot{\psi}_x} \right|_{(x,u)=(\bar{x},\bar{u})} \end{bmatrix} \quad (2.23)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \left. \frac{\partial \ddot{\theta}_x}{\partial u} \right|_{(x,u)=(\bar{x},\bar{u})} \\ \left. \frac{\partial \ddot{\psi}_x}{\partial u} \right|_{(x,u)=(\bar{x},\bar{u})} \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Implicitly differentiating the equations of motion w.r.t. the state variables followed by linearizing them, will result into the following equations:

$$\begin{aligned} \left(\frac{\partial M(\mathbf{q}_{yz})}{\partial x_i} \ddot{\mathbf{q}}_{yz} + M(\mathbf{q}_{yz}) \frac{\partial \ddot{\mathbf{q}}}{\partial x_i} + \frac{\partial C(\mathbf{q}_{yz}, \dot{\mathbf{q}}_{yz})}{\partial x_i} \dot{\mathbf{q}}_{yz} + \frac{\partial G(\mathbf{q}_{yz})}{\partial x_i} \right) \Big|_{(x,u)=(\bar{x},\bar{u})} &= 0 \\ \Leftrightarrow M(\bar{\mathbf{q}}_{yz}) \frac{\partial \ddot{\mathbf{q}}}{\partial x_i} + \frac{\partial G(\mathbf{q}_{yz})}{\partial x_i} \Big|_{(x,u)=(\bar{x},\bar{u})} &= 0 \end{aligned} \quad (2.24)$$

for all $i = 1, 2, 3, 4$, where x_i denotes the i^{th} element of the state vector \mathbf{x} .

Solving these equations for $\frac{\partial \ddot{\mathbf{q}}}{\partial x_i}$ yields the elements of matrix A.

Implicitly differentiating them w.r.t. the input u followed by linearizing them results into the following equations:

$$\begin{aligned} \left(M(\mathbf{q}_{yz}) \frac{\partial \ddot{\mathbf{q}}}{\partial u} \right) \Big|_{(x,u)=(\bar{x},\bar{u})} &= \left(\frac{\partial \boldsymbol{\tau}_{ext}}{\partial u} \right) \Big|_{(x,u)=(\bar{x},\bar{u})} \\ \Leftrightarrow M(\bar{\mathbf{q}}_{yz}) \frac{\partial \ddot{\mathbf{q}}}{\partial u} &= \left(\frac{\partial \boldsymbol{\tau}_{ext}}{\partial u} \right) \Big|_{(x,u)=(\bar{x},\bar{u})}. \end{aligned} \quad (2.25)$$

Solving these equations for $\frac{\partial \ddot{\mathbf{q}}}{\partial u}$ yields the elements of matrix B.

Substituting all parameters into the matrices A and B yields the state space representation

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -75.5784 & 0 & 0 \\ 0 & 34.1594 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 25.6412 \\ -6.4570 \end{bmatrix} u \quad (2.26)$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u. \quad (2.27)$$

Calculations with this state space model in MATLAB show that the controllability matrix and observability matrix of this system both have full rank. This means that the system is totally controllable and observable.

2.5 Equations of motion for XY-plane

2.5.1 Energies

In the xy -plane, the only motion of the BBR is the rotation around the z -axis. Due to the no slip assumption, the ball doesn't rotate around the z -axis, so the kinetic energy of the ball (denoted by $T_{S,xy}$) is equal to zero. The body only has rotational kinetic energy, which is defined as

$$T_{b,xy} = \frac{1}{2} I_{B,xy} \dot{\psi}_z^2 \quad (2.28)$$

Analogous to the energy calculations for the yz -plane, the kinetic energy of the virtual actuating wheel is defined as the sum of the rotational and translational kinetic energy:

$$T_{w,xy} = \frac{1}{2} I_{w,xy} \dot{\phi}_z^2 + \frac{1}{2} m_W v_{W,xy}^T \cdot v_{W,xy}$$

where $v_{W,xy}^T \cdot v_{W,xy}$ is defined as

$$\begin{aligned} v_{W,xy}^T \cdot v_{W,xy} &= \dot{x}_{W,xy}^2 + \dot{y}_{W,xy}^2 \\ &= (r_S + r_W)^2 \dot{\psi}_z^2 \end{aligned}$$

and where $\dot{\phi}_z$ is defined as

$$\dot{\phi}_z = -\frac{r_S}{r_W}\dot{\psi}_z. \quad (2.29)$$

(The derivation of $v_{W,xy}^T \cdot v_{W,xy}$ and $\dot{\phi}_z$ can be found in Appendix B.1).

This yields the equation for the kinetic energy of the virtual actuating wheel

$$\begin{aligned} T_{w,xy} &= \frac{1}{2}I_{w,xy}\left(-\frac{r_S}{r_W}\dot{\psi}_z\right)^2 + \frac{1}{2}m_W(r_S + r_W)^2\dot{\psi}_z^2 \\ &= \frac{1}{2}\left(I_{w,xy}\left(\frac{r_S}{r_W}\right)^2 + m_W(r_S + r_W)^2\right)\dot{\psi}_z^2. \end{aligned} \quad (2.30)$$

It is assumed that the BBR will always stay in contact with the floor, so there is no potential energy present in the xy -plane.

2.5.2 External torques

Analogous to the calculations for the external torques in the yz -plane, the only external torque is the torque of the virtual motor, which transfers a torque to the virtual actuating wheel, denoted by $\tau_{VW,xy}$. $\|\tau_{VW,xy}\|$ is the input of the model and because $\tau_{VW,xy}$ only has a component in the z -direction (denoted by τ_z), the input is $\|\tau_{VW,xy}\| = \tau_z$. A relation between the input torque and the torque that directly controls the minimal coordinate, which forms the external torque of the model, will be derived in this section.

The external torque of the model is shown in Fig. 2.4 together with the force that generates this torque and the lever in case $\psi_z = 0$.

Analogous to the calculations for the external torques in the yz -plane, the relation between $\dot{\phi}_z$ and $\dot{\psi}_z$ is given by the geometric Jacobian. The transpose of the geometric Jacobian relates the input torque τ_z to the torque that directly controls ψ_z .

The relation between $\dot{\phi}_z$ and $\dot{\psi}_z$ is given by

$$\dot{\phi}_z = \underbrace{-\frac{r_S}{r_W}}_J \dot{\psi}_z. \quad (2.31)$$

The relation between the input torque τ_z and the torque that directly controls ψ_z is thus given by

$$\begin{aligned} \tau_{ext,xy} &= J^T \tau_z \\ &= -\frac{r_s}{r_w} \tau_z. \end{aligned} \quad (2.32)$$

2.5.3 Lagrangian and Euler-Lagrange equation for the xy -plane

Define the Lagrangian as

$$L(\psi_x, \dot{\psi}_x) = T_{b,xy} + T_{w,xy}. \quad (2.33)$$

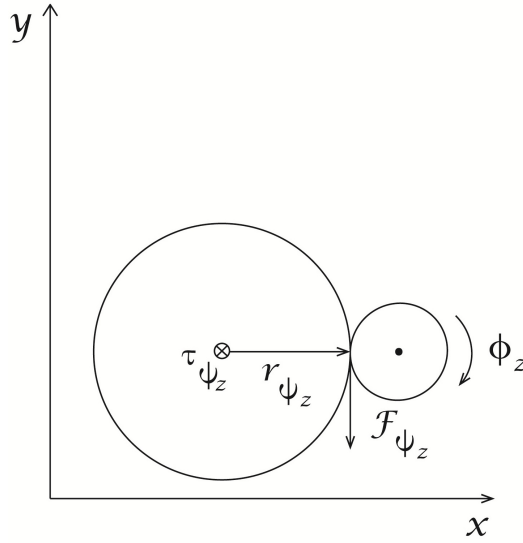


Figure 2.4: Sketch of body torque, generated by the virtual motor torque, the tangential force and corresponding lever in xy -plane.

Calculating the Euler-Lagrange equation

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}_x} \right) - \frac{\partial L}{\partial \psi_x} = \tau_{ext,xy}$$

results into the following equation of motion:

$$M(\mathbf{q}_{xy}) \ddot{\mathbf{q}}_{xy} = \tau_{ext,xy} \quad (2.34)$$

where the left-hand side represents the inertial forces in the xy -plane due to accelerations and the right-hand side represents the external torque.

The matrix $M(\mathbf{q}_{xy})$ is defined as

$$M(\mathbf{q}_{xy}) = I_{B,xy} + m_W (r_S + r_W)^2 + \frac{r_S^2}{r_W} I_{W,xy}. \quad (2.35)$$

Define the following state vector \mathbf{x} (for the xy -plane):

$$\mathbf{x} = \begin{bmatrix} \psi_z \\ \dot{\psi}_z \end{bmatrix}. \quad (2.36)$$

Define as input of the system $u = \tau_z$. Solving Eq. (2.34) for $\ddot{\psi}_z$ results in a linear differential equation:

$$\ddot{\psi}_z = \frac{-r_S u}{I_{B,xy} r_W^2 + m_W r_W^2 (r_S + r_W)^2 + I_{W,xy} r_S^2}. \quad (2.37)$$

Then the linear state space representation is given by:

$$\begin{aligned}\dot{\mathbf{x}} &= A \cdot \mathbf{x} + B \cdot u \\ \mathbf{y} &= C \cdot \mathbf{x} + D \cdot u\end{aligned}\quad (2.38)$$

where the matrices A , B , C and D are defined as:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ -22.2295 \end{bmatrix}\quad (2.39)$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.\quad (2.40)$$

2.6 Torque conversion

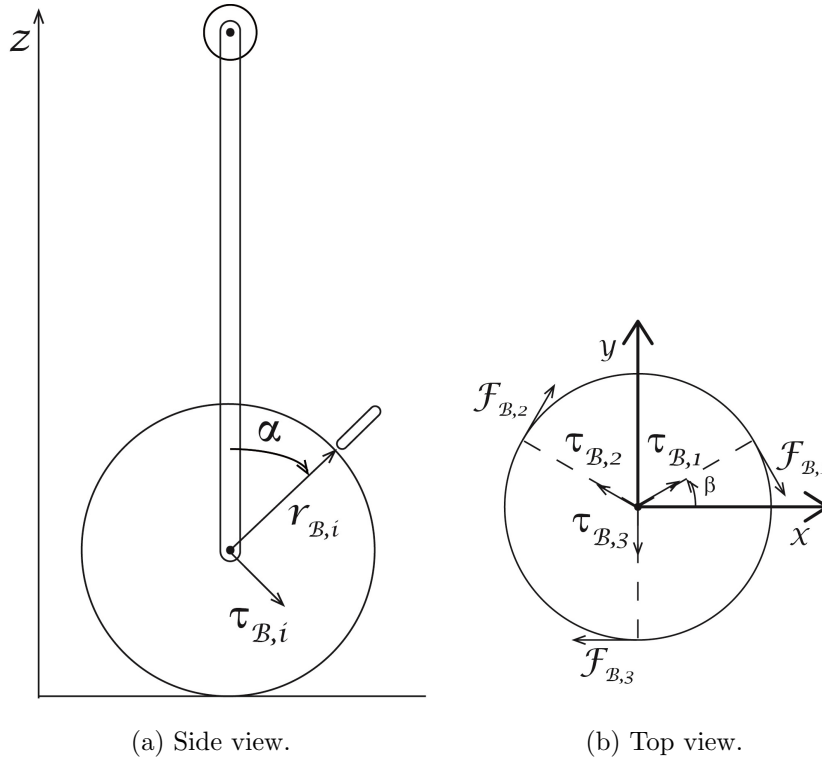


Figure 2.5: Sketch of body torques, generated by the real motor torques, the tangential forces and the corresponding levers.

In the previous paragraphs linear state space models were derived for the different planes. But these models still include virtual variables, namely the torques of the virtual actuating wheels. In this section a relation between the torques of the virtual actuating wheels and the torques of the real omni-wheels will be derived. This relation can be derived using the fact that the resulting torque of the body is conserved (as demonstrated in [2]):

$$\tau_{B,x} + \tau_{B,y} + \tau_{B,z} = \tau_{B,1} + \tau_{B,2} + \tau_{B,3}\quad (2.41)$$

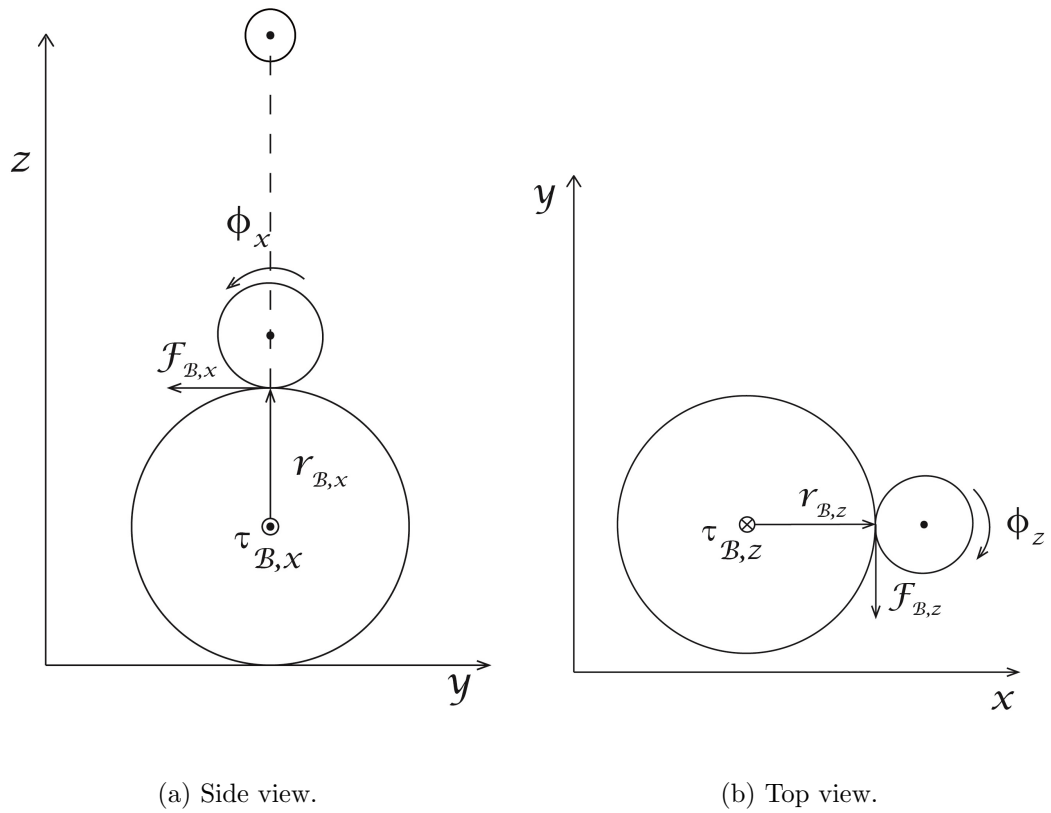


Figure 2.6: Sketch of body torques, generated by the virtual motor torques, the tangential forces and the corresponding levers.

where $\tau_{B,i}$ is defined as the torque vector of the body, generated by the torque of the omni-wheel i (whose magnitude is denoted by τ_i) with $i = 1,2,3$ and where $\tau_{B,j}$ is defined as the torque vector of the body, generated by the torque of the virtual actuating wheel in the direction of the j -axis (whose magnitude is denoted by τ_j) with $j = x,y,z$.

Define $F_{B,i}$ as the force, orthogonal to $\tau_{B,i}$, acting on the body, which is generated by the torque of omni-wheel i , and define $r_{B,i}$ as the lever corresponding to $F_{B,i}$.

Also define $F_{B,j}$ as the force, orthogonal to $\tau_{B,j}$, acting on the body, which is generated by the torque of the virtual actuating wheel, and define $r_{B,j}$ as the lever corresponding to $F_{B,j}$.

Furthermore define α as the angle that sets the vertical position of the omni-wheels and define β as the angle that sets the horizontal position of the first omni-wheel with respect to the x -axis. The position of the second and the third omni-wheel are then defined by respectively the angle $\beta + \frac{2\pi}{3}$ and the angle $\beta - \frac{2\pi}{3}$.

In Fig. 2.5 and Fig. 2.6 sketches of respectively the real and the virtual system are shown.

The torque vectors can be calculated as follows:

$$\tau_{S,i} = r_{S,i} \times F_{S,i}, \quad i = 1,2,3 \quad (2.42)$$

$$\tau_{S,j} = r_{S,j} \times F_{S,j}, \quad j = x,y,z. \quad (2.43)$$

Substituting these torques in Eq. (2.41) results in three equations that can be solved for the torques of the omni-wheels as functions of the torques of the virtual actuating wheels and vice versa.

2.6.1 Calculation of torques of the body $\tau_{B,i}$ generated by the real drive system

In the real system, the forces acting on the body can be derived using the fact that the forces generated by the omni-wheels, are the same forces. Using Fig. 2.5, the forces can be easily derived:

$$\begin{aligned} F_{B,1} &= \frac{\tau_1}{r_W} \begin{bmatrix} \sin(\beta) \\ -\cos(\beta) \\ 0 \end{bmatrix} \\ F_{B,2} &= \frac{\tau_2}{r_W} \begin{bmatrix} \sin(\beta + \frac{2\pi}{3}) \\ -\cos(\beta + \frac{2\pi}{3}) \\ 0 \end{bmatrix} \\ F_{B,3} &= \frac{\tau_3}{r_W} \begin{bmatrix} \sin(\beta - \frac{2\pi}{3}) \\ -\cos(\beta - \frac{2\pi}{3}) \\ 0 \end{bmatrix}. \end{aligned} \quad (2.44)$$

The levers are defined as

$$\begin{aligned}
 r_{B,1} &= r_S \begin{bmatrix} \sin(\alpha) \cos(\beta) \\ \sin(\alpha) \sin(\beta) \\ \cos(\alpha) \end{bmatrix} \\
 r_{B,2} &= r_S \begin{bmatrix} \sin(\alpha) \cos(\beta + \frac{2\pi}{3}) \\ \sin(\alpha) \sin(\beta + \frac{2\pi}{3}) \\ \cos(\alpha) \end{bmatrix} \\
 r_{B,3} &= r_S \begin{bmatrix} \sin(\alpha) \cos(\beta - \frac{2\pi}{3}) \\ \sin(\alpha) \sin(\beta - \frac{2\pi}{3}) \\ \cos(\alpha) \end{bmatrix}.
 \end{aligned} \tag{2.45}$$

Substituting these forces and levers into Eq. (2.42), yields the following torque vectors of the body, generated by the real drive system

$$\begin{aligned}
 \tau_{B,1} &= \begin{bmatrix} \frac{r_S \cos(\alpha) \tau_1 \cos(\beta)}{r_S \cos(\alpha) \tau_1 \sin(\beta)} \\ \frac{r_W}{\sin(\alpha) r_S \tau_1} \\ -\frac{r_W}{r_W} \end{bmatrix} \\
 \tau_{B,2} &= \begin{bmatrix} \frac{r_S \cos(\alpha) \tau_2 \cos(\beta + \frac{2\pi}{3})}{r_S \cos(\alpha) \tau_2 \sin(\beta + \frac{2\pi}{3})} \\ \frac{r_W}{\sin(\alpha) r_S \tau_2} \\ -\frac{r_W}{r_W} \end{bmatrix} \\
 \tau_{B,3} &= \begin{bmatrix} \frac{r_S \cos(\alpha) \tau_3 \cos(-\beta + \frac{2\pi}{3})}{r_S \cos(\alpha) \tau_3 \sin(-\beta + \frac{2\pi}{3})} \\ \frac{r_W}{\sin(\alpha) r_S \tau_3} \\ -\frac{r_W}{r_W} \end{bmatrix}.
 \end{aligned} \tag{2.46}$$

2.6.2 Calculations torques of the body $\tau_{B,j}$ generated by virtual drive system

In the virtual system, the forces acting on the body can be derived using the fact that the forces generated by the virtual actuating wheels are the same forces. Using Fig. 2.6, the forces can be easily derived:

$$\begin{aligned}
 F_{B,x} &= \frac{\tau_x}{r_W} \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \\
 F_{B,y} &= \frac{\tau_y}{r_W} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
 F_{B,z} &= \frac{\tau_z}{r_W} \begin{bmatrix} \sin(\beta) \\ -\cos(\beta) \\ 0 \end{bmatrix}.
 \end{aligned} \tag{2.47}$$

The levers are defined as

$$\begin{aligned} r_{B,x} &= r_S \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ r_{B,y} &= r_S \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ r_{B,z} &= r_S \begin{bmatrix} \cos(\beta) \\ \sin(\beta) \\ 0 \end{bmatrix}. \end{aligned} \quad (2.48)$$

Analogous with the calculations for the real system, this yields the following torque vectors of the body, generated by the virtual drive system

$$\begin{aligned} \tau_{B,x} &= \begin{bmatrix} \frac{r_S \tau_x}{r_W} \\ 0 \\ 0 \end{bmatrix} \\ \tau_{B,y} &= \begin{bmatrix} 0 \\ \frac{r_S \tau_y}{r_W} \\ 0 \end{bmatrix} \\ \tau_{B,z} &= \begin{bmatrix} 0 \\ 0 \\ -\frac{r_S \tau_z}{r_W} \end{bmatrix}. \end{aligned} \quad (2.49)$$

Substituting Eq. (2.46) and Eq. (2.49) into Eq. (2.41) and solving it for the real motor torques τ_1 , τ_2 and τ_3 as functions of the virtual motor torques τ_x , τ_y and τ_z in matrix form yields

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} \frac{2 \cos(\beta)}{3 \cos(\alpha)} & \frac{2 \sin(\beta)}{3 \cos(\alpha)} & \frac{1}{3 \sin(\alpha)} \\ -\frac{\cos(\beta) + \sqrt{3} \sin(\beta)}{3 \cos(\alpha)} & -\frac{\sin(\beta) + \sqrt{3} \cos(\beta)}{3 \cos(\alpha)} & \frac{1}{3 \sin(\alpha)} \\ -\frac{\cos(\beta) + \sqrt{3} \sin(\beta)}{3 \cos(\alpha)} & -\frac{\sin(\beta) + \sqrt{3} \cos(\beta)}{3 \cos(\alpha)} & \frac{1}{3 \sin(\alpha)} \end{bmatrix} \cdot \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}. \quad (2.50)$$

Solving Eq. (2.41) for the virtual motor torques τ_x , τ_y and τ_z as functions of the real motor torques τ_1 , τ_2 and τ_3 in matrix form yields

$$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} \cos(\alpha) \cos(\beta) & -\frac{\cos(\alpha) [\cos(\beta) + \sqrt{3} \sin(\beta)]}{2} & -\frac{\cos(\alpha) [\cos(\beta) - \sqrt{3} \sin(\beta)]}{2} \\ \cos(\alpha) \sin(\beta) & -\frac{\cos(\alpha) [\sin(\beta) - \sqrt{3} \cos(\beta)]}{2} & -\frac{\cos(\alpha) [\sin(\beta) + \sqrt{3} \cos(\beta)]}{2} \\ \sin(\alpha) & \sin(\alpha) & \sin(\alpha) \end{bmatrix} \cdot \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}. \quad (2.51)$$

2.7 Calculation of parameters

In the previous paragraphs linear state space models were derived for the different planes and relations between virtual variables and real variables were derived. Before simulations can be done with the 2D-models, several parameters need to be calculated, which will be done in this paragraph.

Firstly, the moments of inertia of the different parts of the virtual system will be estimated. The total virtual system consists of five rigid parts, namely the ball, the body of the robot and the

three virtual actuating wheels. Moreover, the moments of inertia of the body of the robot and the virtual actuating wheel have horizontal and vertical components. Except when explicitly specified, the moment of inertia is about the axis that goes through the COM. Secondly, the parameters m_W and l will be calculated. Finally, the choice of the value of α will be explained.

2.7.1 Estimation of moment of inertia of the ball

As mentioned in Sec. 1.2, the ball on which the robot will stand is a medicine ball. Therefore to model the ball, it will be approximated by a hollow sphere. The moment of inertia I of a hollow sphere with mass m and radius r is given by [11]

$$I = \frac{2mr^2}{3}. \quad (2.52)$$

The mass of the medicine ball is denoted by m_S its radius by r_S . So the moment of inertia of the ball, denoted by I_S , is defined by

$$I_S = \frac{2m_S r_S^2}{3}. \quad (2.53)$$

2.7.2 Estimation of moment of inertia of the body in yz -/ xz -plane

As mentioned in Sec. 2.2, the body will be approximated by a solid cuboid that consists of an upper and lower part with different density to approximate the non-uniform mass distribution. But the rotation axis of the body does not go through its COM. Therefore, estimating the moment of inertia of the body in the yz -plane will be done in the following three steps:

1. Estimate the moments of inertia of the upper and lower parts of the body about their COM.
2. Calculate the moments of inertia of the body parts about the axis the body rotates around, using the Parallel Axis Theorem, introduced in Sec. 2.4.2.
3. The total moment of inertia is the sum of the different moments of inertia in case that all parts rotate around the same axis. So the total moment of inertia is the sum of the moments of inertia of the body parts.

This method is illustrated in Fig. 2.7.

Step 1: Estimate I_{B1} and I_{B2}

The moment of inertia of the two parts of the body are estimated by the moment of inertia of a cuboid with square base. The moment of inertia of a solid cuboid with mass m , height h and width w (which is equal to the depth) about the x - and y -axis is given by [11]

$$I = \frac{m(w^2 + h^2)}{12}. \quad (2.54)$$

So the moments of inertia of the two parts of the body are estimated by

$$I_{B1} = \frac{m_{B1}(w_B^2 + h_1^2)}{12} \quad (2.55)$$

$$I_{B2} = \frac{m_{B2}(w_B^2 + h_2^2)}{12}. \quad (2.56)$$

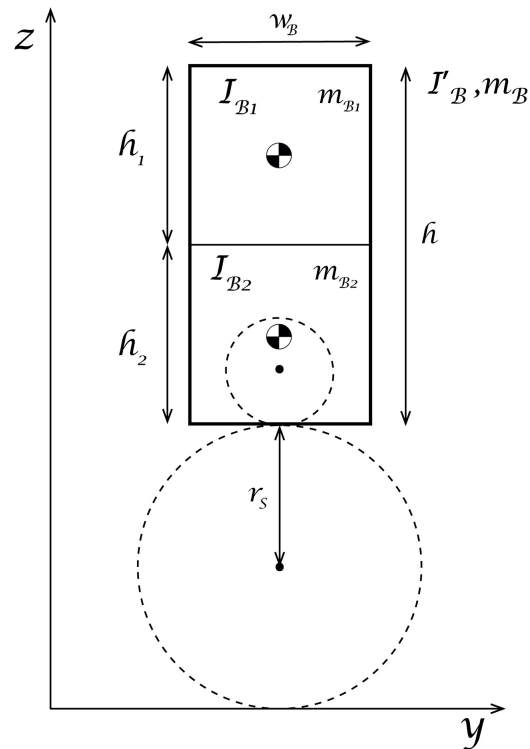


Figure 2.7: Illustration of the method used to estimate the moment of inertia of the body about the rotation axis.

Table 2.2: Description and values of parameters used for the estimation of the moment of inertia of the body.

Parameter	Description	Value
m_{B1}	Mass of the upper part of the body	6.2 kg
m_{B2}	Mass of the lower part of the body	0.935 kg
m_B	Total mass of the body	7.135 kg
h_1	Height of the upper part of the body	0.66 m
h_2	Height of the lower part of the body	0.14 m
w_B	Width of the body	0.2 m
r_S	Radius of the ball	0.115 m
I_W	Moment of inertia of the virtual actuating wheel in yz -/ xz -plane	$1.90 \cdot 10^{-3} \text{ kgm}^2$
I'_B	Moment of inertia of the body about rotation axis	2.40 kgm^2
I_{B1}	Moment of inertia of the upper body part about COM	$2.46 \cdot 10^{-1} \text{ kgm}^2$
I_{B2}	Moment of inertia of the lower body part about COM	$4.64 \cdot 10^{-3} \text{ kgm}^2$

The values of the different parameters can be found in Table 2.2.

Step 2: Calculate the moments of inertia about the axis the body rotates around

The moments of inertia about the axis the body rotates around are calculated using the Parallel Axis Theorem. The distances between the axis through the COM and the axis the body rotates around for each part are derived using Fig. 2.7

$$\begin{aligned} d_{B1} &= r_S + h - \frac{h_1}{2} \\ d_{B2} &= r_S + \frac{h_2}{2}. \end{aligned} \quad (2.57)$$

So the moments of inertia of the two body parts about the axis the body rotates around are given by

$$\begin{aligned} I'_{B1} &= I_{B1} + m_{B1}d_{B1}^2 \\ I'_{B2} &= I_{B2} + m_{B2}d_{B2}^2. \end{aligned} \quad (2.58)$$

Step 3: Calculate the moment of inertia of the whole body

Then the moment of inertia of the whole body in the vertical planes is given by

$$I'_B = I'_{B1} + I'_{B2}. \quad (2.59)$$

2.7.3 Estimation of moment of inertia of the body in xy -plane

The moment of inertia of a solid cuboid with mass m , height h and width w about the z -axis is given by [11]

$$I = \frac{mw^2}{6}. \quad (2.60)$$

So the moment of inertia of the body about the z -axis is estimated by

$$I_{B,xy} = \frac{m_B w_B^2}{6}. \quad (2.61)$$

2.7.4 Estimation moments of inertia of virtual actuating wheel

The omni-wheels and the motors are modeled together as a virtual actuating wheel, so based on the moments of inertia of the omni-wheels and the motors, the moments of inertia of the virtual actuating wheels have to be estimated. This can be done by comparing the rotational energy around an axis in the virtual system to the rotational energy around the same axis in the real system (as demonstrated in [2]).

The rotational energy of rotations around the x -, y -, and z -axis, in the real system, will be calculated separately. Then, by equating it to the corresponding rotational energy of the virtual system, the desired moments of inertia can be derived. For simplicity, β will be taken zero.

In order to calculate the rotational energies of the real motors and the omni-wheels, the angular rates of the omni-wheels around each axis have to be calculated. These angular rates are derived using Fig. 2.8, where $\omega_{Wi,j}$ denotes the angular rate of omni-wheel i around axis j ($i = 1,2,3$ and $j = x,y,z$).

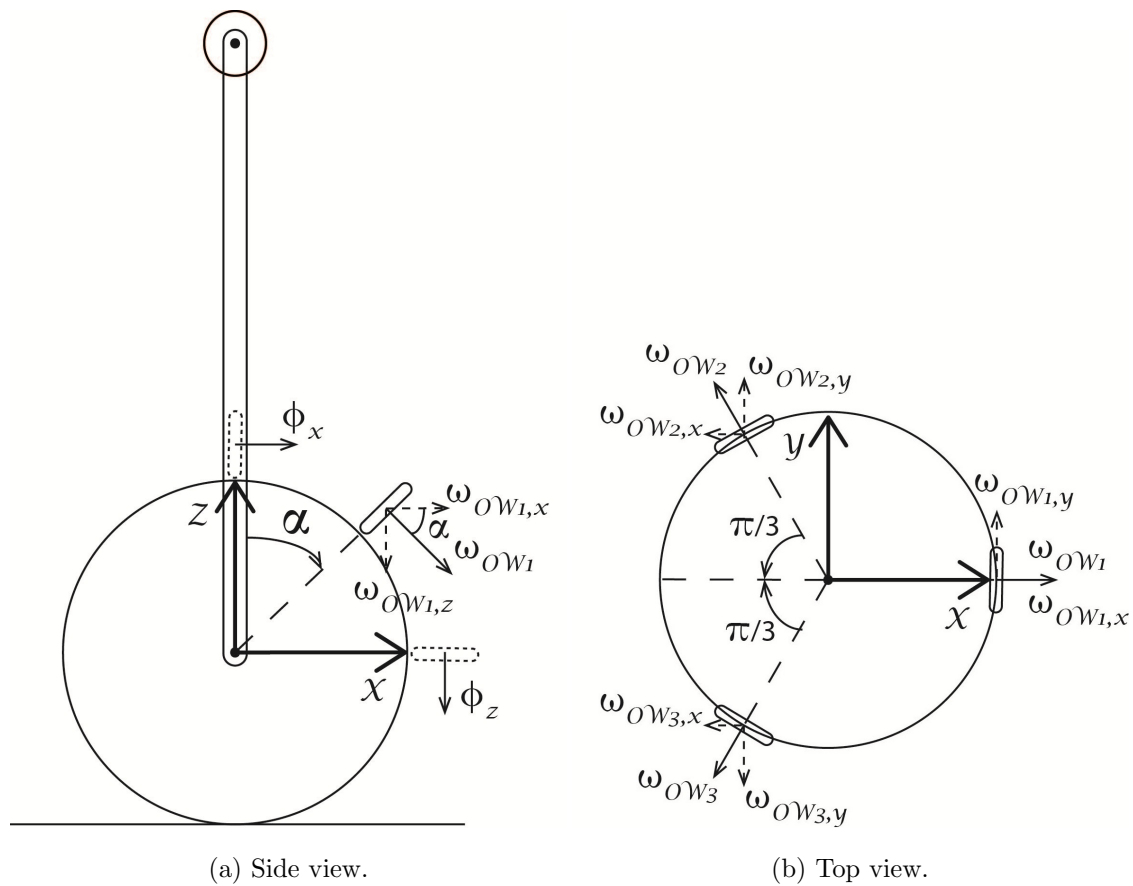


Figure 2.8: Sketch of the decomposed angular rates of the omni-wheels.

The angular rates of the omni-wheels around the x -axis are formulated as

$$\begin{aligned}\omega_{W1,x} &= \dot{\phi}_x \cos(\alpha) \\ \omega_{W2,x} = \omega_{W3,x} &= \cos\left(\frac{\pi}{3}\right)(-\dot{\phi}_x) \cos(\alpha) \\ &= -\frac{1}{2}\dot{\phi}_x \cos(\alpha).\end{aligned}\tag{2.62}$$

The angular rates of the omni-wheels around the y -axis are formulated as

$$\begin{aligned}\omega_{W1,y} &= 0 \\ \omega_{W2,y} &= \sin\left(\frac{\pi}{3}\right)\dot{\phi}_y \cos(\alpha) \\ &= \frac{1}{2}\sqrt{3}\dot{\phi}_y \cos(\alpha) \\ \omega_{W3,y} &= \sin\left(\frac{\pi}{3}\right)(-\dot{\phi}_y) \cos(\alpha) \\ &= -\frac{1}{2}\sqrt{3}\dot{\phi}_y \cos(\alpha).\end{aligned}\tag{2.63}$$

The angular rates of the omni-wheels are around the z -axis formulated as

$$\begin{aligned}\omega_{W1,z} = \omega_{W2,z} = \omega_{W3,z} &= \cos\left(\frac{\pi}{2} - \alpha\right)\dot{\phi}_z \\ &= \sin(\alpha)\dot{\phi}_z.\end{aligned}\tag{2.64}$$

Before the moments of inertia of the virtual actuating wheels can be calculated, some notation has to be introduced. The moment of inertia of the rotor of the real motor is denoted by I_M . Furthermore, the reduction of the gear box is denoted by k , so the angular velocity of the omni-wheel is reduced with a factor k compared to the angular velocity of the motor. The moment of inertia of the gear rotor is not taken into account, because it is negligible compared to the moments of inertia of the motor and the omni-wheel.

The moment of inertia of a omni-wheel, denoted by I_{OW} , is estimated by approaching the omni-wheel as a solid cylinder. The moment of inertia of a solid cylinder with mass m , height h and radius r , about the rotation axis, is given by [11]

$$I = \frac{mr^2}{2}.\tag{2.65}$$

The mass of the omni-wheel (denoted by m_{OW}) consists of three parts, namely the mass of the omni-wheel itself (denoted by m_{omni}), the mass of the clamping bush (denoted by m_{cb}) and the mass of the flange (denoted by m_f). The radius of the omni-wheel is denoted by r_{OW} .

So the moment of inertia of a omni-wheel I_{OW} will be estimated by

$$\begin{aligned}I_{OW} &= \frac{m_{OW}r_{OW}^2}{2} \\ &= \frac{(m_{omni} + m_{cb} + m_f)r_{OW}^2}{2}.\end{aligned}\tag{2.66}$$

The moments of inertia of the virtual actuating wheel ($I_{w,x}$, $I_{w,y}$ and $I_{w,z}$) can now be calculated by equating the rotational energy of the virtual actuating wheel to the corresponding energy of the omni-wheels and real motors. This is done in Appendix B.1.

The vertical component of the moment of inertia of the virtual actuating wheel (denoted by I_W) is defined as

$$\begin{aligned} I_W &= I_{W,x} = I_{W,y} \\ &= \frac{3}{2} \cos(\alpha)^2 (I_{OW} + k^2 I_M). \end{aligned} \quad (2.67)$$

The horizontal component of the moment of inertia of the virtual actuating wheel (denoted by $I_{W,xy}$) is defined as

$$\begin{aligned} I_{W,xy} &= I_{W,z} \\ &= 3 \sin(\alpha)^2 (I_{OW} + k^2 I_M). \end{aligned} \quad (2.68)$$

2.7.5 Calculation of m_W

In the virtual drive system, the motors (including gearheads) and omni-wheels are modeled as virtual actuating wheels. Therefore, the mass of the virtual actuating wheel m_W can be estimated by the sum of the masses of the different parts of the virtual actuating wheel. So m_W is defined as

$$m_W = m_{gh} + m_M + m_{OW} \quad (2.69)$$

where m_{gh} denotes the mass of the gearhead and m_M denotes the mass of the motor.

2.7.6 Calculation of l

The parameter l denotes the length between the COM of the ball and the COM of the body as is shown in Fig. 2.2a). It is defined as

$$l = \frac{m_{B1} d_{B1} + m_{B2} d_{B2}}{m_{B1} + m_{B2}} \quad (2.70)$$

where d_{B1} and d_{B2} are defined in Eq. (2.57). Substituting the expressions for d_{B1} and d_{B2} yields the following expression for l :

$$l = \frac{m_{B1} (r_S + h - \frac{h_1}{2}) + m_{B2} (r_S + \frac{h_2}{2})}{m_{B1} + m_{B2}}. \quad (2.71)$$

2.7.7 Choice of α

The angle α determines the vertical position of the omni-wheels on the ball as is shown in Fig. 2.5a). It is a significant parameter, because it determines the ratio between the rotation of the omni-wheel and the rotation of the ball and it determines the support of the body.

When α is chosen close to 90° , rotation around the x - and y -axis is almost impossible, while rotation around the z -axis can be easily obtained. When α is chosen close to 0° , the body quickly falls down, because the supporting triangle is so narrow and rotation of the body around the z -axis becomes difficult. This reasoning is confirmed by Eq. (2.46).

Based on this reasoning, an angle of 45° is chosen for the BBR to have sufficient support for the body and also make rotation around the x - and y -axis easily possible.

2.8 Simulation of 2D model

In the previous paragraphs linear state space models were derived for the different planes and relations between virtual variables and real variables were derived. Also several parameters were calculated. Therefore, it is now possible to simulate the 2D models.

The aim of this paragraph is to analyze the nonlinear 2D model of the vertical planes and furthermore to see how accurate the linearized model approximates the nonlinear model. Also the 2D model for the horizontal plane will be analyzed.

2.8.1 Simulations

Firstly, the free response (i.e. nonzero initial conditions and no external torques) of the nonlinear model for the yz -plane is simulated for 10 seconds with an initial pitch angle of 0.1° . For the first 1.2 seconds, also the free response of the linear model is shown. The results are shown in Fig. 2.9.

Fig. 2.9 shows that the robot will fall down and start to oscillate around the centre of the ball. As the floor is not defined, the robot will go through the plane $z = 0$ and the amplitudes of the oscillations do not decrease, because no energy loss or dynamic friction is modeled.

Fig. 2.9 also shows that the response of the linear model is almost indistinguishable from the nonlinear model with regard to the pitch angle ψ_x , within a range of 20° of the linearization point, which is confirmed by the fact that the error percentage of the linear model is only 3.16% , when $\psi_x = 20^\circ$.

However, the response of the linear model with regard to θ_x , the angle rotated by the ball, has already an error of 27.62% when $\theta_x = -90^\circ$ (which is equal to a displacement of about 20 cm).

Secondly, the forced response (i.e. zero initial conditions and nonzero external torques) of the linear and nonlinear model for the yz -plane are simulated for 1.2 seconds with a constant external torque of 0.01 Nm. The results are shown in Fig. 2.10.

Fig. 2.10 shows that, also in this case, the response of the linear model is almost indistinguishable from the nonlinear model with regard to the pitch angle ψ_x , within a range of 20° of the linearization point. This is confirmed by the fact that the error percentage of the linear model is only 3.13% , when $\psi_x = 20^\circ$.

Also in this case, the response of the linear model with regard to θ_x , the angle rotated by the ball, has already an error of 23.06% when $\theta_x = -90^\circ$ (which is equal to a displacement of about 20 cm). Therefore, the linearized model seems to be unreliable with regard to the position of the ball, but it should be noted that the position of the ball depends on the pitch and roll angles. Moreover, the relation between the position of the ball and the pitch and roll angles is linear, which can be derived from Eq. (B.3). So for values of the pitch and roll angles for which the linear model is a reliable approximation of the nonlinear model, the linear model should also be reliable with regard to the position of the ball. This is confirmed by Fig. 2.9 and Fig. 2.10, where the linear model for the y -position of the ball becomes unreliable when the pitch angle ψ_x is more than 20° .

Thirdly, the forced response of the model for the xy -plane is simulated for 1 second with a constant external torque of 0.3 Nm and the results are shown in Fig. 2.11.

As expected, a constant external torque results in an angle that increases quadratically and

a angular velocity that increases linearly, because the torque is proportional to the acceleration.

2.8.2 Shortcomings 2D model

The 2D models describe the main dynamical behaviour of the BBR. However, it is assumed that the vertical planes and the horizontal plane are independent in order to simplify the modeling. But in reality, these planes are not fully independent. For example, the 3D system contains a centrifugal force, because of rotation around the vertical axis, which is not modeled in any of the 2D models.

Moreover, the 2D models contain several virtual parts in order to approximate the 3D system. Therefore, a full 3D model will be developed in the next chapter, which will form a more reliable basis for the design of the controller.

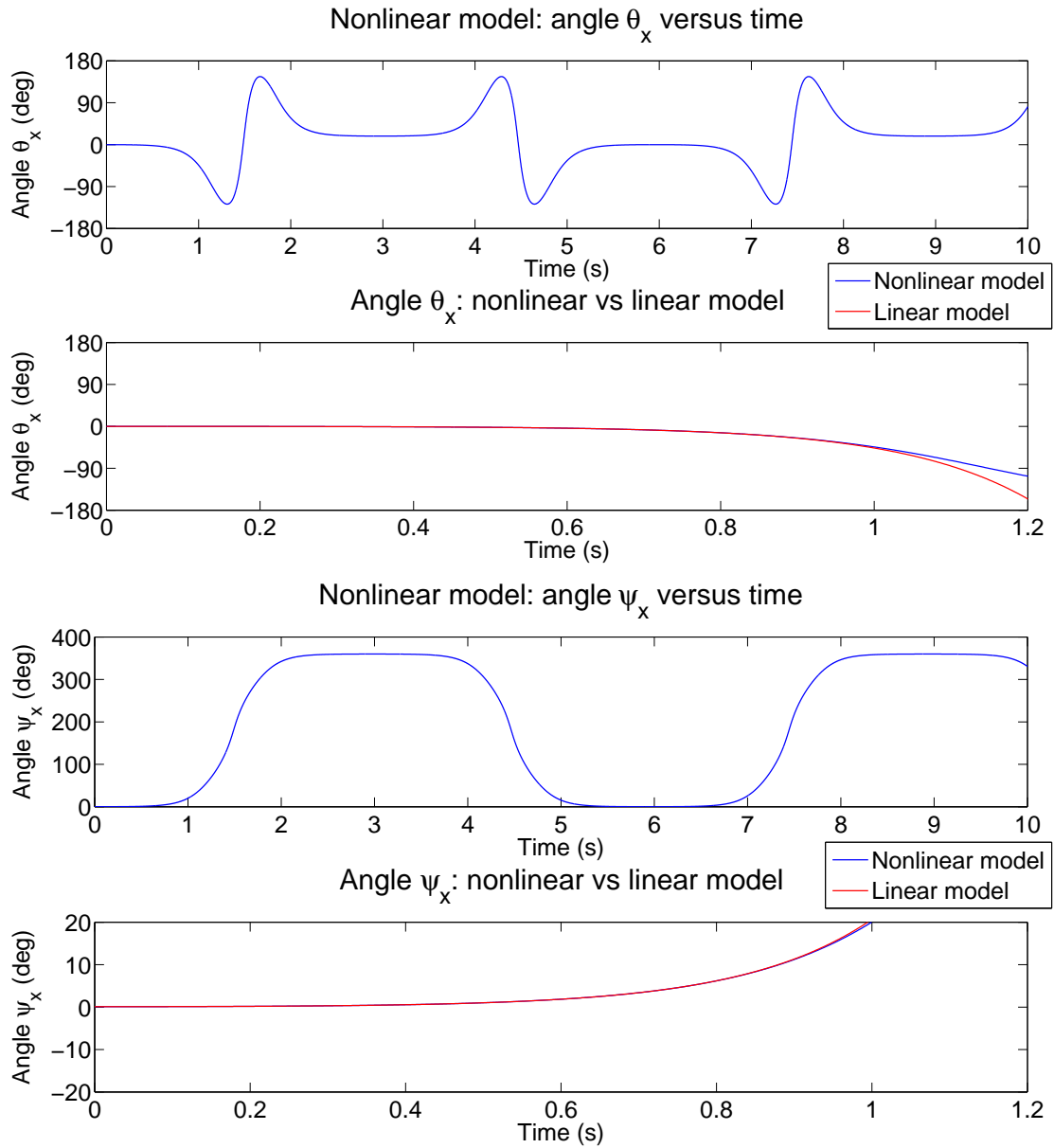


Figure 2.9: Free response of the linear and nonlinear 2D model for the yz -plane with an initial pitch angle of 0.1° .

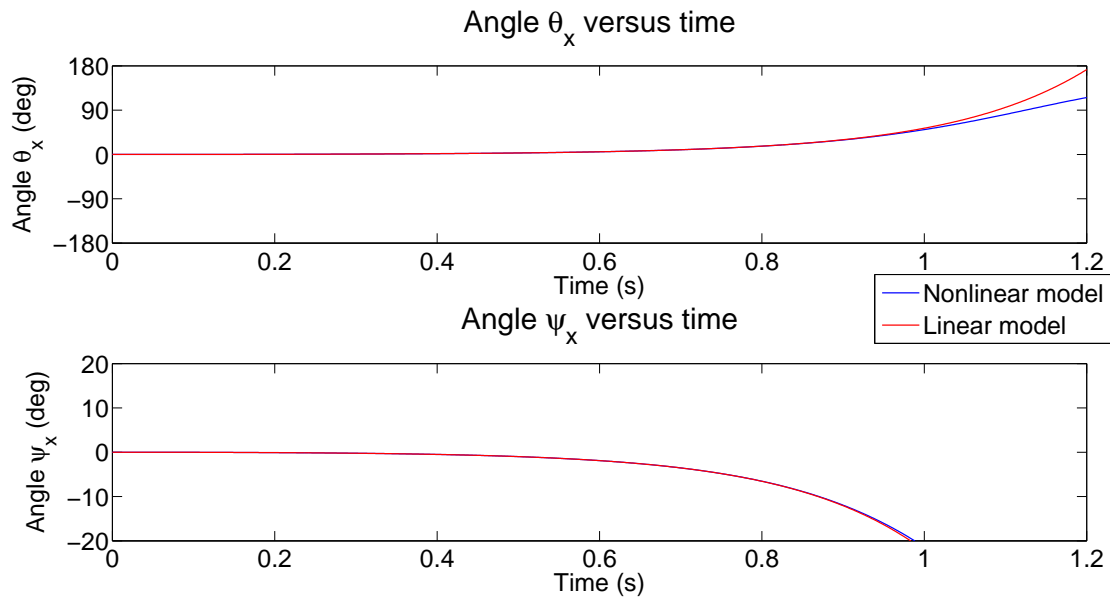


Figure 2.10: Forced response of the linear and nonlinear 2D model for the yz -plane with a constant external torque of 0.01 Nm.

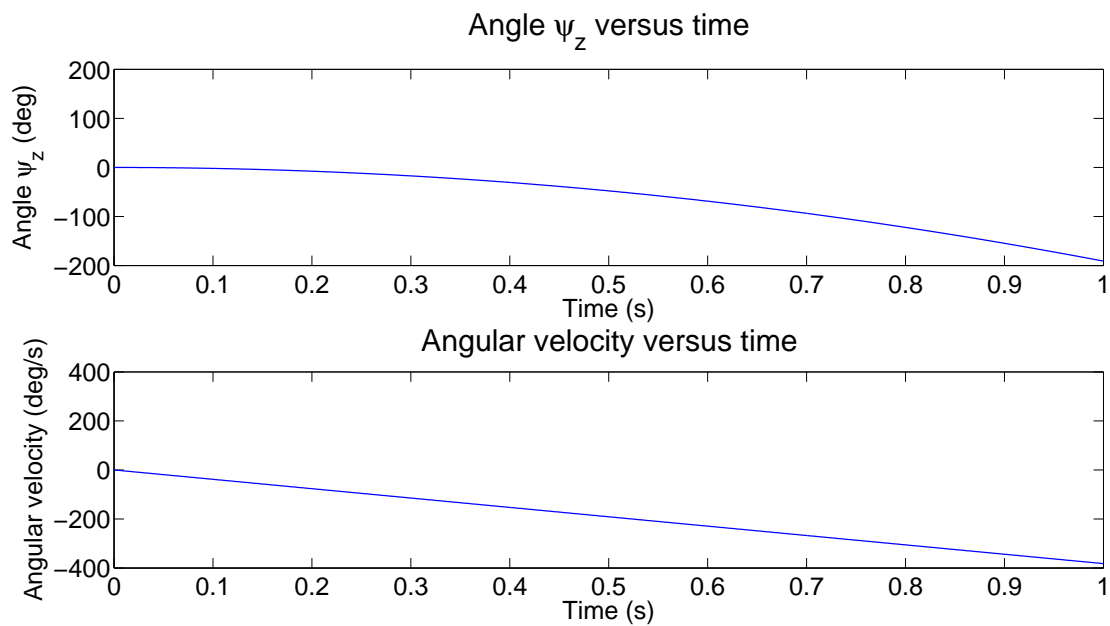


Figure 2.11: Forced response of the 2D model for the xy -plane with a constant external torque of 0.3 Nm

3. 3D MODEL

Modeling the BBR as three 2D models has several shortcomings as mentioned in Sec. 2.8.2. Therefore a full 3D model will be made in this chapter.

In the first paragraph assumptions will be made. In the second paragraph a description of the model will be given. After that coordinates will be defined in the third paragraph. In the fourth paragraph the main variables of the system will be derived. Some parameters will be derived in the fifth paragraph. Finally, in the sixth paragraph the equations of motion will be derived.

3.1 Assumptions

The 3D model is based on the same physical system as the 2D model is based on, i.e. the assumptions made for the 3D model are the same as the assumptions made for the 2D model, except the assumption of the independent vertical planes. In short, the assumptions were:

- Rigid bodies/floor
- No slip
- No rolling or kinetic friction
- Horizontal floor
- No time delay
- Simplified omni-wheels

These assumptions are described in more detail in Sec. 2.1.

3.2 Description of 3D-model

The 3D model consists of two rigid bodies, namely the ball and the body of the robot. The body of the robot includes the drive system with the omni-wheels attached to it, see Fig. 3.1 for a sketch.

The ball is modeled as a hollow sphere, the body of the robot is modeled as a solid cuboid and the omni-wheels are modeled as solid disks. Analogous to the 2D model for the yz -plane, the non-uniform mass distribution of the body is modeled by dividing the body into an upper and lower part that both have a uniform mass distribution. This makes it possible to set the height of the COM of the body.

3.3 Coordinates

To derive the equations of motion, coordinates need to be defined. The coordinates are defined based on the DOF of the system. The system has five DOFs, namely two DOFs for the translation

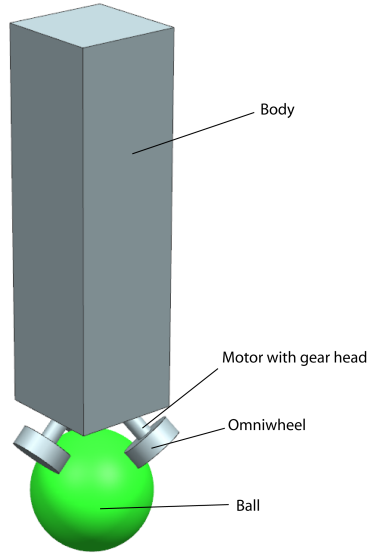


Figure 3.1: Sketch of the 3D model.

of the ball and three DOFs for the rotation of the body. For the ball it is only necessary to know its translation. Therefore, define as minimal coordinate vector

$$\mathbf{q} = [x_S \quad y_S \quad \psi_x \quad \psi_y \quad \psi_z]^T \quad (3.1)$$

where analogous to the 2D model, ψ_x , ψ_y and ψ_z indicate the orientation of the body. And x_S and y_S indicate the translation of the ball along respectively the x - and y -axis.

Coordinate frames are used to define the position of the ball and the orientation of the body. The coordinate frames that define the orientation of the body are defined using Tait-Bryan angles (a variation on Euler angles). Define the following six coordinate frames:

- **Inertial frame I** The inertial frame of reference is denoted by I .
- **Coordinate frame 1** Translating the inertial frame with x_S along its x -axis yields coordinate frame 1.
- **Coordinate frame 2** Translating coordinate frame 1 with y_S along its y -axis yields coordinate frame 2. Coordinate frame 2 has its origin located in the centre of the ball.
- **Coordinate frame 3** Rotating coordinate frame 2 around its z -axis with an angle ψ_z counterclockwise yields coordinate frame 3.
- **Coordinate frame 4** Rotating coordinate frame 3 around its y -axis with an angle ψ_y counterclockwise yields coordinate frame 4.
- **Coordinate frame 5** Rotating coordinate frame 4 around its x -axis with an angle ψ_x clockwise together with a constant translation of l m along its z -axis yields coordinate frame 5, which has its origin located in the COM of the body.

All these coordinate frames are sketched in Fig. 3.2, where Ψ_i denotes coordinate frame i .

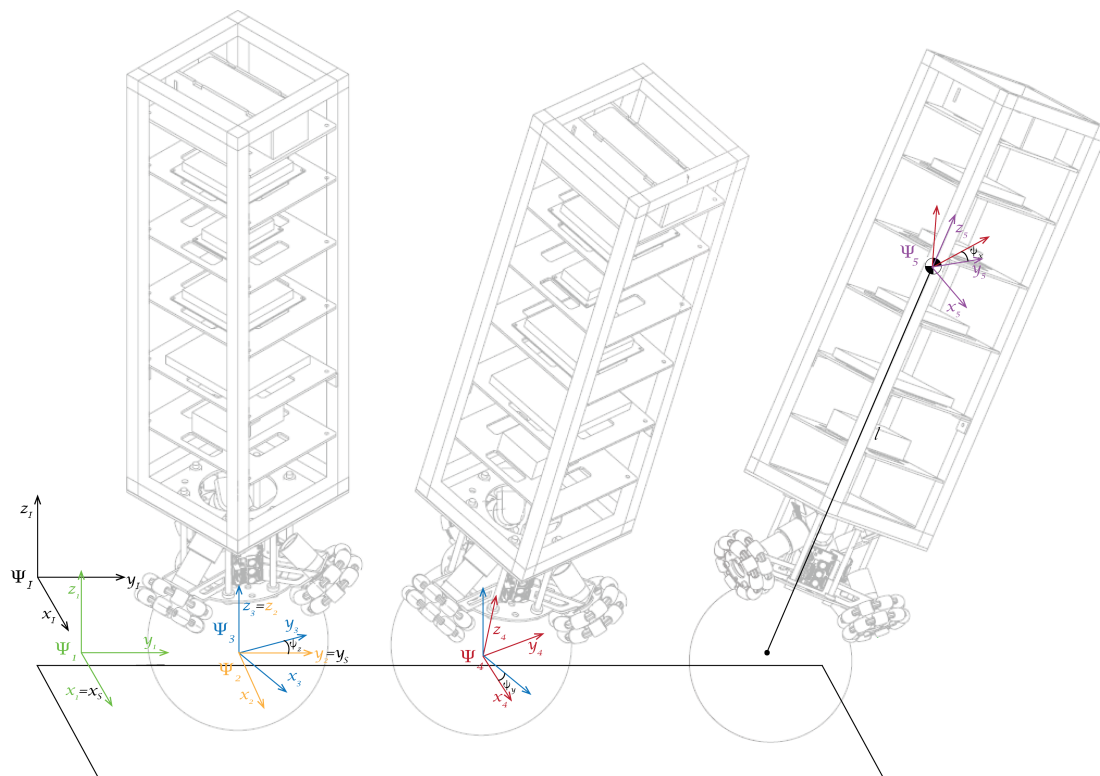


Figure 3.2: Sketch of the different coordinate frames.

3.3.1 Transformations between coordinate frames

The transformations from one coordinate frame to another can be expressed using homogeneous matrices. Homogeneous matrices describe the rotation and translation between different coordinate frames. A homogeneous matrix from Ψ_i to Ψ_j is defined as

$$H_i^j = \begin{bmatrix} R_i^j & o_i^j \\ 0_3 & 1 \end{bmatrix} \quad (3.2)$$

where R_i^j denotes the rotation matrix from Ψ_i to Ψ_j , o_i^j denotes the origin of Ψ_i expressed in Ψ_j and where $0_3 = [0, 0, 0]$.

The homogeneous matrices that describe the transformations between consecutive frames are defined as:

$$\begin{aligned} H_I^1 &= \begin{bmatrix} 1 & 0 & 0 & -x_S \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ H_1^2 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -y_S \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ H_2^3 &= \begin{bmatrix} \cos(\psi_z) & \sin(\psi_z) & 0 & 0 \\ -\sin(\psi_z) & \cos(\psi_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ H_3^4 &= \begin{bmatrix} \cos(\psi_y) & 0 & \sin(\psi_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\psi_y) & 0 & \cos(\psi_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ H_4^5 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\psi_x) & -\sin(\psi_x) & 0 \\ 0 & \sin(\psi_x) & \cos(\psi_x) & -l \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (3.3)$$

With these matrices given, all the other homogeneous matrices, that describe the transformations between different coordinate frames, can be calculated.

3.4 Variables

Now the coordinates and the coordinate frames are defined, the variables, that are needed for the calculation of the equations of motion, can be derived. But before these variables can be derived, some notation needs to be introduced.

3.4.1 Notation

Due to different notations in literature, it is important to define the used notation in this chapter. The notation and description of the different variables used in this chapter can be found in Table

3.1.

The theory and notation used in this chapter is based on [9].

Table 3.1: Description and notation of variables and parameters used in the 3D model.

Notation	Description
H_i^j	Homogeneous matrix from Ψ_i to Ψ_j
R_i^j	Rotation matrix from frame Ψ_i to Ψ_j
σ_i^j	Position of origin of Ψ_i expressed in Ψ_j
$\mathbf{r}_k^{i,j}$	Position of Ψ_k w.r.t. Ψ_j , expressed in Ψ_i
$\mathbf{v}_k^{i,j}$	Linear velocity of Ψ_k w.r.t. Ψ_j , expressed in Ψ_i
$\boldsymbol{\omega}_k^{i,j}$	Angular velocity of Ψ_k w.r.t. Ψ_j , expressed in Ψ_i
$\mathbf{T}_k^{i,j}$	Twist of Ψ_k w.r.t. Ψ_j , expressed in Ψ_i
$Ad_{H_i^j}$	Adjoint matrix of H_i^j that maps twists from Ψ_i to Ψ_j
I_O^i	Inertia tensor of an object O about its COM, chosen in Ψ_i
I_O	Moment of inertia of an object O about its COM
I_O'	Moment of inertia of an object O about its rotation axis

3.4.2 Twists

A twist $\mathbf{T}_k^{i,j} \in \mathbb{R}^{6 \times 1}$ is defined as

$$\mathbf{T}_k^{i,j} = \begin{bmatrix} \boldsymbol{\omega}_k^{i,j} \\ \mathbf{v}_k^{i,j} \end{bmatrix} \quad (3.4)$$

where $\boldsymbol{\omega}_k^{i,j}$ is the angular velocity of Ψ_k w.r.t. Ψ_j , expressed in Ψ_i and where $\mathbf{v}_k^{i,j}$ is the linear velocity of Ψ_k w.r.t. Ψ_j , expressed in Ψ_i .

A coordinate change of a twist $\mathbf{T}_k^{i,j}$ to coordinate frame n is defined as

$$\mathbf{T}_k^{n,j} = Ad_{H_i^n} \cdot \mathbf{T}_k^{i,j} \quad (3.5)$$

where $Ad_{H_i^n} \in \mathbb{R}^{6 \times 6}$ is the adjoint matrix of H_i^n , which is defined as

$$Ad_{H_i^n} = \begin{bmatrix} R_i^n & 0_{3 \times 3} \\ \tilde{\sigma}_i^n \cdot R_i^n & R_i^n \end{bmatrix}. \quad (3.6)$$

where $0_{3 \times 3}$ denotes a 3-by-3 matrix of zeros and where $\tilde{\sigma}_i^n$ is the tilde form of σ_i^n . The tilde form of σ_i^n is defined as

$$\sigma_i^n = \begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix} \Rightarrow \tilde{\sigma}_i^n = \begin{bmatrix} 0 & -o_z & o_y \\ o_z & 0 & -o_x \\ -o_y & o_x & 0 \end{bmatrix} \quad (3.7)$$

The twists of the different bodies w.r.t. the inertial frame need to be derived in order to be able to calculate the total energy of the system.

Ball velocities

The linear velocity of the ball w.r.t. Ψ_I , expressed in Ψ_I is defined as the linear velocity of Ψ_2

w.r.t. Ψ_I , expressed in Ψ_I :

$$\mathbf{v}_2^{I,I} = \begin{bmatrix} \dot{x}_S \\ \dot{y}_S \\ 0 \end{bmatrix}. \quad (3.8)$$

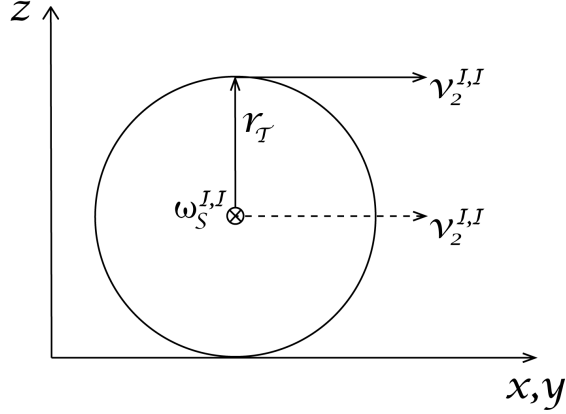


Figure 3.3: Twist of the ball.

The angular velocity $\omega_2^{I,I}$ (which is not the angular velocity of the ball) is equal to zero, because Ψ_2 doesn't rotate w.r.t. Ψ_I .

So the twist of Ψ_2 w.r.t. Ψ_I , expressed in Ψ_I is defined as

$$\mathbf{T}_2^{I,I} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{x}_S \\ \dot{y}_S \\ 0 \end{bmatrix}. \quad (3.9)$$

Now also the twist of Ψ_2 w.r.t. Ψ_I expressed in Ψ_2 (denoted by $\mathbf{T}_2^{2,I}$) can be calculated. It is defined by

$$\mathbf{T}_2^{2,I} = Ad_{H_7^2} \cdot \mathbf{T}_2^{I,I}. \quad (3.10)$$

For later calculations also the linear velocity of the ball w.r.t. the inertial frame, expressed in frame 5 (denoted by $\mathbf{v}_2^{5,I}$) is needed. Using the fact that

$$\mathbf{T}_2^{5,I} = Ad_{H_7^5} \cdot \mathbf{T}_2^{I,I} \quad (3.11)$$

an expression for $\mathbf{v}_2^{5,I}$ can be easily derived:

$$\mathbf{v}_2^{5,I} = \delta_I^5 \cdot R_I^5 \cdot \omega_2^{I,I} + R_I^5 \cdot \mathbf{v}_2^{I,I}. \quad (3.12)$$

For calculating the angular velocity of ball, define r_T as the vector from the centre of the ball to the point on top of the ball (denoted by T), see Fig. 3.3:

$$\mathbf{r}_T = \begin{bmatrix} 0 \\ 0 \\ r_S \end{bmatrix}. \quad (3.13)$$

The angular velocity of the ball cannot be defined as the angular velocity of the coordinate frames, so the angular velocity of the ball w.r.t. Ψ_I , expressed in Ψ_2 will be denoted by $\omega_S^{2,I}$ and it is defined as

$$\begin{aligned} \omega_S^{2,I} &= \frac{\mathbf{r}_T \times \mathbf{v}_2^{2,I}}{\|\mathbf{r}_T\|^2} \\ &= \begin{bmatrix} \frac{\dot{y}_S}{r_S} \\ -\frac{\dot{x}_S}{r_S} \\ 0 \end{bmatrix}. \end{aligned} \quad (3.14)$$

Body velocities

The twist of the body w.r.t. Ψ_I , expressed in Ψ_I is defined as

$$\mathbf{T}_5^{I,I} = J \cdot \dot{\mathbf{q}}_j \quad (3.15)$$

where J is the Jacobian that maps the joint velocities \dot{q}_j to the end-effector (i.e. the part that is connected to the end of a robot ‘arm’) velocities. The end-effector is in this case the body of the BBR.

The vector of joint velocities $\dot{\mathbf{q}}_j$ is defined as

$$\dot{\mathbf{q}}_j = [\dot{x}_S \quad \dot{y}_S \quad \dot{\psi}_z \quad \dot{\psi}_y \quad \dot{\psi}_x]^T. \quad (3.16)$$

The Jacobian is defined as

$$\begin{aligned} J &= [\mathbf{T}_1^{I,I} \quad \mathbf{T}_2^{I,1} \quad \mathbf{T}_3^{I,2} \quad \mathbf{T}_4^{I,3} \quad \mathbf{T}_5^{I,4}] \\ &= \begin{bmatrix} \mathbf{T}_1^{I,I} & Ad_{H_1^I} \cdot \mathbf{T}_2^{1,1} & Ad_{H_2^I} \cdot \mathbf{T}_3^{2,2} & Ad_{H_3^I} \cdot \mathbf{T}_4^{3,3} & Ad_{H_4^I} \cdot \mathbf{T}_5^{4,4} \end{bmatrix} \end{aligned} \quad (3.17)$$

where the twists can be derived using Fig. 3.2. They are defined as

$$\begin{aligned} \mathbf{T}_1^{I,I} &= [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0]^T \\ \mathbf{T}_2^{1,1} &= [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0]^T \\ \mathbf{T}_3^{2,2} &= [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^T \\ \mathbf{T}_4^{3,3} &= \left[\omega_4^{3,3} \quad \left(\omega_4^{3,3} \times R_4^3 \cdot [0 \quad 0 \quad l]^T \right) \right] \\ &= [0 \quad 1 \quad 0 \quad l \quad 0 \quad 0]^T \\ \mathbf{T}_5^{4,4} &= \left[\omega_5^{4,4} \quad \left(\omega_5^{4,4} \times R_5^4 \cdot [0 \quad 0 \quad l]^T \right) \right] \\ &= [-1 \quad 0 \quad 0 \quad 0 \quad l \quad 0]^T \end{aligned} \quad (3.18)$$

where $\omega_4^{3,3} = [0 \quad 1 \quad 0]^T$ and $\omega_5^{4,4} = [-1 \quad 0 \quad 0]^T$.

Now also the twist of the body of the robot w.r.t. Ψ_I expressed in Ψ_5 (denoted by $\mathbf{T}_5^{5,I}$) can be calculated. It is defined by

$$\mathbf{T}_5^{5,I} = Ad_{H_I^5} \cdot \mathbf{T}_5^{I,I}. \quad (3.19)$$

Angular rate of the omni-wheels

The omni-wheels are attached to the body of the robot, so only the rotational motion of the omni-wheels has to be calculated. And because the omni-wheels are fixed to the body of the robot, only the angular rate is relevant for the model.

Define $\dot{\phi}_i$ as the angular rate of omni-wheel i about the motor axis for $i = 1,2,3$, where a positive angular rate means counterclockwise rotation of the omni-wheels.

The angular rate of the omni-wheels can be expressed in the minimal coordinates using the fact that due to the no slip assumption, the circumferential speed of the ball in the direction of the omni-wheel is the same as the circumferential speed of the omni-wheel. To calculate the circumferential speed of the ball, first the angular velocity of the ball w.r.t. the body needs to be calculated.

The angular velocity of the ball w.r.t the body expressed in frame 5 is defined as

$$\begin{aligned} \boldsymbol{\omega}_S^{5,5} &= \boldsymbol{\omega}_S^{5,I} - \boldsymbol{\omega}_5^{5,I} \\ &= R_2^5 \cdot \boldsymbol{\omega}_S^{2,I} - \boldsymbol{\omega}_5^{5,I}. \end{aligned} \quad (3.20)$$

To calculate the circumferential speed of the ball, first define the vectors from the centre of the ball to the contact points with the omni-wheels (see Fig. 3.4a) as

$$\begin{aligned} \mathbf{r}_{W1} &= r_S \begin{bmatrix} \sin(\alpha) \\ 0 \\ \cos(\alpha) \end{bmatrix} \\ \mathbf{r}_{W2} &= r_S \begin{bmatrix} -\frac{1}{2} \sin(\alpha) \\ \frac{1}{2} \sqrt{3} \sin(\alpha) \\ \cos(\alpha) \end{bmatrix} \\ \mathbf{r}_{W3} &= r_S \begin{bmatrix} -\frac{1}{2} \sin(\alpha) \\ -\frac{1}{2} \sqrt{3} \sin(\alpha) \\ \cos(\alpha) \end{bmatrix} \end{aligned} \quad (3.21)$$

which are already derived in Sec. 2.6. The angle β , that determines the horizontal position of the omni-wheels (see Fig. 2.5), is chosen to zero.

Furthermore, define the unit vectors, that give the direction of the circumferential speed of the omni-wheels in the point of contact with the ball, in case of a positive angular rate of the omni-wheels, (see Fig. 3.4b) as

$$\begin{aligned} \mathbf{u}_{W1} &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ \mathbf{u}_{W2} &= \begin{bmatrix} -\frac{1}{2} \sqrt{3} \\ -\frac{1}{2} \\ 0 \end{bmatrix} \\ \mathbf{u}_{W3} &= \begin{bmatrix} \frac{1}{2} \sqrt{3} \\ -\frac{1}{2} \\ 0 \end{bmatrix}. \end{aligned} \quad (3.22)$$

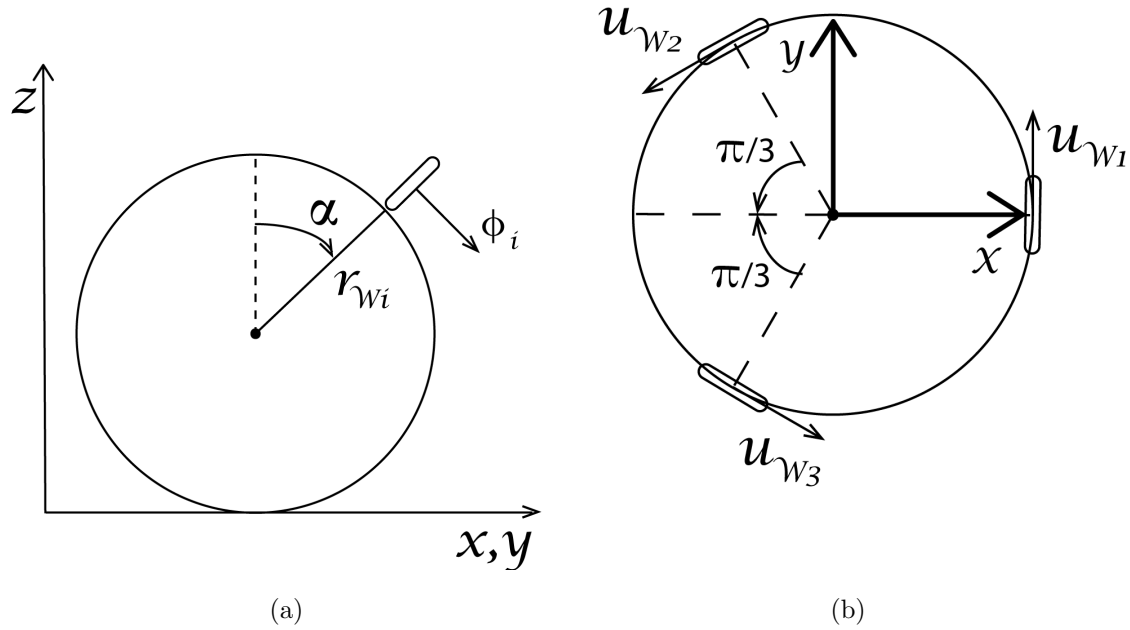


Figure 3.4: Sketch of the position of the omni-wheels.

Now the circumferential speed of the ball in the direction of the omni-wheel can be equated to the circumferential speed of the omni-wheel:

$$(\boldsymbol{\omega}_S^{5,5} \times \mathbf{r}_{W_i}) \cdot \mathbf{u}_{W_i} = \dot{\phi}_i r_W, \quad i = 1, 2, 3. \quad (3.23)$$

So the angular rates of the omni-wheels expressed in the minimal coordinates are given by

$$\dot{\phi}_i = \frac{1}{r_W} (\boldsymbol{\omega}_S^{5,5} \times \mathbf{r}_{W_i}) \cdot \mathbf{u}_{W_i}, \quad i = 1, 2, 3. \quad (3.24)$$

3.5 Parameters

In the previous paragraphs coordinates and coordinate frames were defined and variables were derived. Before the equations of motion can be derived, some parameters need to be calculated. Most of the parameters used in the 3D model are calculated and defined in Chap. 2 and the values can be found in Table A.1, but some additional parameters have to be calculated for the 3D model. Those additional parameters are the inertia tensors (i.e. the matrix that contains the moments of inertia and the inertia products about the three coordinate axes) of the ball and the body and the distance between the centre of the ball and the COM of the body (denoted by l).

3.5.1 Calculation of inertia tensor of the ball

The inertia tensor of the ball is defined as

$$I_S^2 = \begin{bmatrix} I_S & 0 & 0 \\ 0 & I_S & 0 \\ 0 & 0 & I_S \end{bmatrix} \quad (3.25)$$

and I_S is estimated in Sec. 2.7.1.

3.5.2 Calculation of inertia tensor of the body

In contrast with the estimation of the moments of inertia of the 2D model, the estimation of the moment of inertia of the body also takes into account the motors, gearheads and omni-wheels by adding the masses of the motors, gearheads and omni-wheels to the mass of the lower part of the body. This simplifies the calculation for the energies of the omni-wheels, because now only the rotational energy of the omni-wheels needs to be calculated and the translational energy of the omni-wheels is not taken into account with the calculation of the translational energy of the body.

The inertia tensor of the body is defined as

$$I_B^5 = \begin{bmatrix} I_{B,x} & 0 & 0 \\ 0 & I_{B,y} & 0 \\ 0 & 0 & I_{B,z} \end{bmatrix} \quad (3.26)$$

where $I_{B,j}$ is the moment of inertia of the body about the j -axis of the coordinate frame centered in its COM, for $j = x, y, z$. And because the inertia tensors are chosen in the coordinate frame centered in the COM, the inertia products become zero.

But $I_{B,j}$ cannot be calculated directly because of the non-uniform mass distribution of the body. The moments of inertia of the two parts of the body can only be added if they are the moments of inertia about the axis they both rotate around. So first the moments of inertia of the two parts of the body about the axis through their COM will be calculated. After that the moments of inertia of the two parts of the body about the rotation axis will be calculated and added. Finally, $I_{B,j}$ can be calculated using the Parallel Axis Theorem. For more details about these steps, see Sec. 2.7.2 and Fig. 2.7.

Step 1: Estimate $I_{B1,j}$ and $I_{B2,j}$

The moments of inertia of the two parts of the body are estimated by the moment of inertia of a cuboid with square base. The moment of inertia of a solid cuboid with mass m , height h and width w about the x - or y -axis is given by [11]

$$I = \frac{m(w_B^2 + h^2)}{12}. \quad (3.27)$$

The moment of inertia of a solid cuboid with mass m , height h and width w about the z -axis is given by [11]

$$I = \frac{mw_B^2}{6}. \quad (3.28)$$

In contrast with the calculations for the 2D model, the mass of the lower part of the body now also includes the mass of the motors, gearheads and omni-wheels:

$$m_{B2,tot} = m_{B2} + 3(m_M + m_{gh} + m_{OW}) \quad (3.29)$$

So the moments of inertia of the two parts of the body about their COM for the axis j are estimated by

$$I_{B1,j} = \begin{cases} \frac{m_{B1}(w_B^2 + h_1^2)}{12}, & \text{if } j = x \\ \frac{m_{B1}(w_B^2 + h_1^2)}{12}, & \text{if } j = y \\ \frac{m_{B1}w_B^2}{6}, & \text{if } j = z \end{cases} \quad (3.30)$$

$$I_{B2,j} = \begin{cases} \frac{m_{B2,tot}(w_B^2 + h_2^2)}{12}, & \text{if } j = x \\ \frac{m_{B2,tot}(w_B^2 + h_2^2)}{12}, & \text{if } j = y \\ \frac{m_{B2,tot}w_B^2}{6}, & \text{if } j = z \end{cases}$$

Step 2: Calculate the moments of inertia about the axis the body rotates around

The moments of inertia about the x - and y -axis were calculated about their COM, although they do not rotate around their COM. The moments of inertia about the axis the body rotates around are calculated using the Parallel Axis Theorem. The distances between the axis through the COM and the axis the body rotates around for each part are derived in Sec. 2.7.2 and given by Eq. (2.57).

So the moments of inertia of the two body parts about the axis the body rotates around are given by

$$I'_{B1,j} = \begin{cases} I_{B1,j} + m_{B1}d_{B1}^2, & \text{if } j = x \\ I_{B1,j} + m_{B1}d_{B1}^2, & \text{if } j = y \\ I_{B1,j}, & \text{if } j = z \end{cases} \quad (3.31)$$

$$I'_{B2,j} = \begin{cases} I_{B2,j} + m_{B2,tot}d_{B2}^2, & \text{if } j = x \\ I_{B2,j} + m_{B2,tot}d_{B2}^2, & \text{if } j = y \\ I_{B2,j}, & \text{if } j = z \end{cases}$$

Step 3: Calculate the moment of inertia of the whole body about each rotation axis

Now the moment of inertia of the whole body about each rotation axis j is given by

$$I'_{B,j} = I'_{B1,j} + I'_{B2,j}, \quad j = x, y, z. \quad (3.32)$$

Finally, the moments of inertia of the body about its COM can be calculated using the Parallel Axis Theorem again, which states that

$$I'_{B,j} = \begin{cases} I_{B,j} + m_B l^2, & \text{if } j = x \\ I_{B,j} + m_B l^2, & \text{if } j = y \\ I_{B,j}, & \text{if } j = z \end{cases} \quad (3.33)$$

Therefore the moments of inertia of the body about its COM are defined as

$$I_{B,j} = \begin{cases} I'_{B,j} - m_B l^2, & \text{if } j = x \\ I'_{B,j} - m_B l^2, & \text{if } j = y \\ I'_{B,j}, & \text{if } j = z \end{cases} \quad (3.34)$$

3.5.3 Calculation of height COM

The parameter l , which is the height of the COM of the body of the robot, is defined as

$$l = \frac{m_{B1}d_{B1} + m_{B2,tot}d_{B2}}{m_{B1} + m_{B2,tot}}. \quad (3.35)$$

(Later this parameter has been measured in experiments and the estimated value has been replaced by the measured value.)

The values of all the parameters used in this paragraph can be found in Table A.1.

3.6 Equations of motion

In the previous paragraphs coordinates and coordinate frames were defined, variables were derived and parameters were calculated. In this section the equations of motion for the 3D system will be derived. Again the Lagrangian method is used as defined in Sec. 2.4.1.

3.6.1 Kinetic and potential energy of the system

Energy ball

The kinetic energy of the ball is defined as:

$$\begin{aligned} T_S &= \frac{1}{2} \mathbf{T}_2^{2,I^T} \cdot \begin{bmatrix} I_S^2 & 0_{3 \times 3} \\ 0_{3 \times 3} & m_S \cdot Id_3 \end{bmatrix} \cdot \mathbf{T}_2^{2,I} + \frac{1}{2} \boldsymbol{\omega}_S^{2,I^T} \cdot I_S^2 \cdot \boldsymbol{\omega}_S^{2,I} \\ &= \frac{1}{2} \boldsymbol{\omega}_2^{2,I^T} \cdot I_S^2 \cdot \boldsymbol{\omega}_2^{2,I} + \frac{1}{2} m_S \mathbf{v}_2^{2,I^T} \cdot \mathbf{v}_2^{2,I} + \frac{1}{2} \boldsymbol{\omega}_S^{2,I^T} \cdot I_S^2 \cdot \boldsymbol{\omega}_S^{2,I} \\ &= \frac{1}{2} m_S \mathbf{v}_2^{2,I^T} \cdot \mathbf{v}_2^{2,I} + \frac{1}{2} \left(R_I^2 \cdot \boldsymbol{\omega}_S^{I,I} \right)^T \cdot I_S^2 \cdot \left(R_I^2 \cdot \boldsymbol{\omega}_S^{I,I} \right) \end{aligned} \quad (3.36)$$

where Id_3 denotes the 3-by-3 identity matrix and $\boldsymbol{\omega}_2^{2,I} = [0 \ 0 \ 0]^T$.

The potential energy of the ball is zero, because it is assumed that the ball only moves over horizontal surfaces:

$$V_S = 0. \quad (3.37)$$

Energy body

Analogous, the kinetic energy of the body is defined as

$$\begin{aligned} T_B &= \frac{1}{2} \mathbf{T}_5^{5,I^T} \cdot \begin{bmatrix} I_B^5 & 0_{3 \times 3} \\ 0_{3 \times 3} & m_B \cdot Id_3 \end{bmatrix} \cdot \mathbf{T}_5^{5,I} \\ &= \frac{1}{2} \boldsymbol{\omega}_5^{5,I^T} \cdot I_B^5 \cdot \boldsymbol{\omega}_5^{5,I} + \frac{1}{2} m_B \mathbf{v}_5^{5,I^T} \cdot \mathbf{v}_5^{5,I} \end{aligned} \quad (3.38)$$

The potential energy of the body is defined as

$$V_B = m_B \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} R_5^I \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix}. \quad (3.39)$$

Energy omni-wheels

The kinetic energy of the omni-wheels only consists of rotational energy, which is defined as

$$T_{W_i} = \frac{1}{2}I_{OW}\dot{\phi}_i^2 + \frac{1}{2}I_M(k\dot{\phi}_i)^2, \quad i = 1,2,3. \quad (3.40)$$

3.6.2 External torques

The system is actuated by the omni-wheels and motors, so the external torques are the torques of the motors which transfer a torque to the omni-wheels. Define the magnitude of the torque of omni-wheel i (denoted by τ_i for $i = 1,2,3$) as the inputs of the 3D model. As shown in Chap. 2 for the 2D model, the external torques are given by the transpose of the geometric Jacobian that maps the rates of the minimal coordinates to the angular rates of the omni-wheels. Using the expressions for the angular rates of the omni-wheels, given in Eq. (3.24), this Jacobian can be derived:

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix} = J \cdot \begin{bmatrix} \dot{x}_S \\ \dot{y}_S \\ \dot{\psi}_x \\ \dot{\psi}_y \\ \dot{\psi}_z \end{bmatrix}. \quad (3.41)$$

So the external torques of the model (denoted by $\boldsymbol{\tau}_{ext}$) are given by

$$\boldsymbol{\tau}_{ext} = J^T \cdot \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}. \quad (3.42)$$

3.6.3 Lagrangian and Euler-Lagrange equations

Define the Lagrangian as

$$L(\mathbf{q}, \dot{\mathbf{q}}) = T_S + T_B + T_{W1} + T_{W2} + T_{W3} - V_B. \quad (3.43)$$

Calculate the Euler-Lagrange equations

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_{ext,i}, \quad i = 1, \dots, 5. \quad (3.44)$$

The same notation as for the 2D model is used to display the equations of motion for the 3D model:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) = \boldsymbol{\tau}_{ext}. \quad (3.45)$$

Due to their complexity the matrices $M(\mathbf{q})$, $C(\mathbf{q}, \dot{\mathbf{q}})$ and $G(\mathbf{q})$ are not displayed here.

3.6.4 Linearization

For balancing the BBR the equations of motion are only needed around the position the BBR stands upright. Therefore a linearized model, that is linearized around the position the BBR stands upright, is used to model the dynamics of the BBR. In Sec. 3.7 it will be verified whether

the linearized model is a valid approximation of the nonlinear model. First, the states and inputs need to be defined.

Define the state \mathbf{x} as

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} = [x_S \quad y_S \quad \psi_x \quad \psi_y \quad \psi_z \quad \dot{x}_S \quad \dot{y}_S \quad \dot{\psi}_x \quad \dot{\psi}_y \quad \dot{\psi}_z]^T. \quad (3.46)$$

and define the inputs \mathbf{u} as

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}. \quad (3.47)$$

The system will be linearized around the unstable equilibrium, where all state variables and inputs are zero:

$$\begin{aligned} \bar{\mathbf{x}} &= [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \\ \bar{\mathbf{u}} &= [0 \quad 0 \quad 0]^T. \end{aligned} \quad (3.48)$$

Analogous to the 2D model, the equations of motion for the 3D model will be implicitly linearized, which is explained in detail in Sec. 2.4.5. This results in the linear state space representation:

$$\begin{aligned} \dot{\mathbf{x}} &= A \cdot \mathbf{x} + B \cdot \mathbf{u} \\ \mathbf{y} &= C \cdot \mathbf{x} \end{aligned} \quad (3.49)$$

where the matrices A and B are defined as

$$\begin{aligned} A &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -13.78 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -13.78 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 25.58 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25.58 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ B &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -3.73 & 3.73 \\ 4.30 & -2.15 & -2.15 \\ -6.16 & 3.08 & 3.08 \\ 0 & 5.34 & -5.34 \\ -20.99 & -20.99 & -20.99 \end{bmatrix} \end{aligned} \quad (3.50)$$

and where C is the 10-by-10 identity matrix.

Calculations with this state space model in MATLAB show that the system is totally controllable and observable.

3.7 Simulation of 3D model

In the previous paragraphs a complete 3D dynamical model was derived and linearized around the position the BBR stands upright. In this paragraph, simulations with this 3D model will be done to analyze the nonlinear 3D model. Furthermore, the accuracy of the linearized model will be investigated.

3.7.1 Simulations

The same kind of simulations will be done for the 3D model as have been done for the 2D model in Sec. 2.8.1.

So firstly, the free response (i.e. nonzero initial conditions and no external torques) of the nonlinear model is simulated for 10 seconds with an initial pitch angle of 0.1° . For the first 2.25 seconds, also the free response of the linear model is shown. The results are shown in Fig. 3.5.

Fig. 3.5 shows that the robot will fall down and start to oscillate around the centre of the ball. As the floor is not defined, the robot will go through the plane $z = 0$ and the amplitudes of the oscillations do not decrease, because no energy loss or dynamic friction is modeled.

Analogous to the simulations of the 2D model, Fig. 3.5 shows that the response of the linear model is almost indistinguishable from the nonlinear model with regard to the pitch angle ψ_x , within a range of 20° of the linearization point and the error percentage of the linear model is only 4.12%, when $\psi_x = 20^\circ$.

Also, the response of the linear model with regard to y_S , the y -position of the ball, has already an error of 27.00% when $y_S = 35$ cm.

Secondly, the forced response (i.e. zero initial conditions and nonzero external torques) of the linear and nonlinear model are simulated for 2 seconds with a constant external torque of 0.01 Nm. The results are shown in Fig. 3.6.

Analogous to the simulations of the 2D model, Fig. 3.6 shows that, also in this case, the response of the linear model is almost indistinguishable from the nonlinear model with regard to the pitch angle ψ_x , within a range of 20° of the linearization point, which is confirmed by the fact that the error percentage of the linear model is only 3.56%, when $\psi_x = 20^\circ$.

Again, the response of the linear model with regard to y_S has already a high error of 44.04% when y_S is only 35 cm.

So, from the results shown in Fig. 3.5 and Fig. 3.6, it can be concluded that the linear model is reliable with regard to the orientation of the body of the BBR when the tilt angle is at most 20° . Analogous to the 2D model, the linear model seems to be unreliable with regard to the position of the ball. However, as also mentioned in Sec. 2.8.1, the position of the ball is linear dependent on the pitch and roll angles, so the linear model with regard to the position of the ball is only reliable if the pitch and roll angles are at most 20° , which is confirmed by Fig. 3.5 and Fig. 3.6.

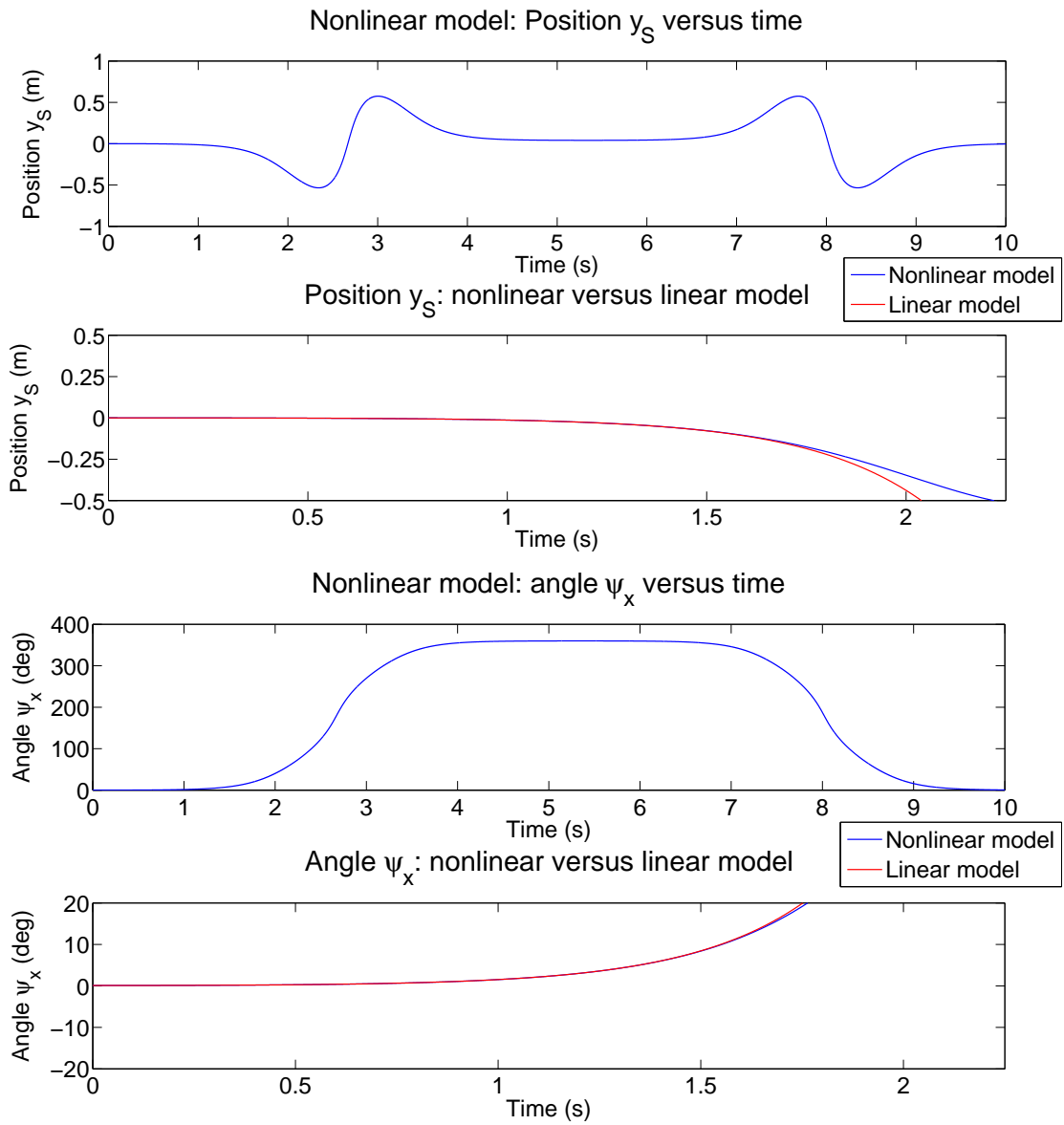


Figure 3.5: Free response of the linear and nonlinear 3D model with an initial pitch angle of 0.1° .

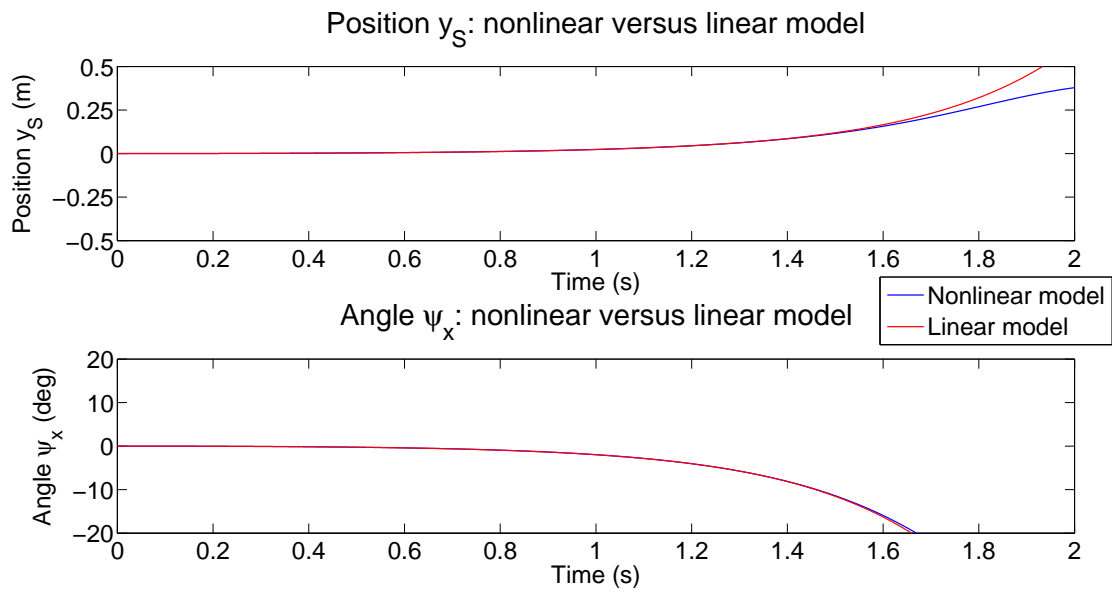


Figure 3.6: Forced response of the linear and nonlinear 3D model with a constant external torque of 0.01 Nm.

4. DESIGN OF A CONTROLLER

In this chapter, a controller will be designed, based on the derived dynamical 3D model. The controller will control the 3D orientation of the body and the position of the BBR, which includes both station keeping and tracking of given set points.

In the first paragraph, the results of a literature research on designed BBR controllers will be shown. The controller requirements will be formulated in the second paragraph together with a controller design approach. A static LQR controller will be designed in the third paragraph that controls the orientation of the body and the position of the BBR simultaneously. In the fourth paragraph, a SISO cascade controller will be designed with manual loopshaping, which consists of an inner loop for controlling the orientation of the body and an outer loop for position control. In the fifth paragraph, it will be explained how the position of the BBR is derived from the measurements. Finally, a short conclusion can be found in the sixth paragraph.

4.1 Literature review

Different approaches have been taken in the design of a balancing controller for a Ball-Balancing Robot. These approaches can be roughly divided into two groups. The first group consists of controllers whose design is based on a nonlinear model of the BBR, like is done in [5], [13] and [15]. The second group consists of controllers whose design is based on the linearized model of the BBR, as shown in [3], [7] and [19]. In that case, the 3D system is often modeled by independent planar models of the vertical planes, which are controlled by independent controllers. It is also possible to use a combination of linear controllers (based on a full 3D model), which results in a nonlinear controller, as demonstrated in [2].

The most common position controller approach consists of simultaneously controlling the orientation and the position of the BBR by using LQR control theory, as shown in [2], [5], [7] and [19]. Out of these approaches [2] and [5] also use feedforward (FF) control to improve tracking of set points. Also cascade control systems are used, where the balancing controller and the position controller are designed separately, see [13]. In this case, the control system consists of an inner loop for balancing control and an outer loop for position control.

In the next section, an overview is given of both the balancing and the position controllers found in literature.

4.1.1 Overview of controllers

University: Carnegie Mellon University (2006 and 2009).

Balancing controller: PID controller.

Position controller: PID controller + offline trajectory planning.

The BBR of Carnegie Mellon University, shown in Fig. 1.3a) was first controlled by a control system that consisted of an inner loop and an outer loop, described in [8] (published in 2006):

- *Inner loop:* The inner loop consisted of a PI controller that was used for control of the angular velocity of the ball. This inner loop automatically compensated for the various

frictional torques that had to be overcome to achieve velocity tracking, thus reducing the effect of the unmodeled static and dynamic friction.

- *Outer loop*: The outer loop consisted of a LQR controller that used full state feedback.

With these controllers, the system was able to keep position and track set points, although the performance was poor. In 2009, two other papers were published ([15], [14]), in which these controllers were replaced by a PID controller for balancing and a PID controller for station keeping and tracking set points, where the output of the position controller is used for the input of the balancing controller. Also an offline trajectory planning algorithm is proposed that plans trajectories, which are used as reference, for the body angle in order to move the ball to the desired position.

University: Tohoku Gakuin University (2009).

Balancing controller: PD controller.

Position controller: PD controller.

The BBR of Tohoku Gakuin University, shown in Fig. 1.3b), is controlled by two PD controllers, that control the pitch and roll angle and the position of the BBR simultaneously [7]. The position controller is used for both station keeping and tracking set points, so the problem of station keeping and tracking set points is reduced to one regulation problem.

The inputs of the plant are decoupled in x -, y - and z -direction, so the controllers do not control the omni-wheels directly, but control virtual wheels that lie in the same plane as the sensors for the pitch and roll angle. Accelerations are used as controller outputs, which are integrated to velocities and the velocities are used as plant inputs. The gains of the PD controllers are tuned experimentally. In the first experiments, a basketball was used, but although the robot could balance, the behaviour was unsteady. This seems to happen due to insufficient stiffness of the basketball. Therefore, the basketball was replaced by a more rigid ball, which showed a satisfying behaviour. Experiments also showed that the BBR was able to transport loads of at least 10 kg.

University: University of Adelaide (2009).

Balancing controller: LQR controller (PD).

Position controller: LQR controller (PID), only used for station keeping.

The BBR of University of Adelaide is built from LEGO and shown in Fig. 1.3c), is controlled by a LQR controller that uses full state feedback. In contrast with many other papers, the states consist of all the quantities that can be directly measured, i.e. the body angles and the motor shaft angles. In addition, they also introduced two extra states, which are the integral of the motor shaft angles around the x - and y -axis. This results in a LQR controller which is in fact a PID controller for the motor shaft angles and a PD controller for the pitch and roll angles [3]. The controller for the motor shaft angles allows controlling the position of the BBR, but it is only used for station keeping.

Experiments showed that the controller stabilized the system and provided satisfying disturbance attenuation. However, the performance of the robot was limited by nonlinearities that were present in the system, in particular nonlinearities present in the motor dynamics.

University: National Chung Hsing University (2012).

Balancing controller: LQR controller (PD).

Position controller: LQR controller (PD).

The BBR of National Chung Hsing University, shown in Fig. 1.3d), is also controlled by a LQR controller that used full state feedback [19], which is in fact a PD controller for the body angles and a PD controller for the position of the BBR. Like is done in [7], the problem of station keeping and tracking set points is reduced to one regulation problem.

This paper also shows the capabilities of LQR controller theory for a robot that is not built from LEGO.

University: ETH Zürich (2010).

Balancing controller: nonlinear controller based on LQR control theory.

Position controller: nonlinear controller based on LQR control theory + feedforward.

The BBR of ETH Zürich, called Rezero and shown in Fig. 1.3e) and f), is controlled by a nonlinear controller, that consists of a combination of different LQR controllers, based on different linearization points. Gain scheduling is used to combine the different linear controllers to a nonlinear controller. The idea of gain scheduling is to design a linear controller at different operating points and to interpolate the controller gains for the current states during operation [2]. Also feedforward is used.

Experiments show that the controller stabilizes the system and Rezero has been presented in public for several times, showing the full potential of moving on a single ball.

University: Bilkent University & Middle East Technical University (2012).

Balancing controller: Inverse dynamics controller + PD controller.

Position controller: Inverse dynamics controller + PD controller.

Inal et al. [5] developed a nonlinear 3D model for a BBR. An inverse dynamics controller is used in combination with a PD controller to control the body angles and the position.

Simulations show that these controllers are capable of sustaining dynamic behaviours such as circular trajectories in a robust and stable way. However, as their paper lacks results, it is unsure whether these controllers have ever been implemented and tested.

4.2 Control approach

Simulations in Sec. 3.7 showed that the linearized model is a reliable approximation of the nonlinear model with regard to the orientation angles of the body, provided that the tilt angle is at most 20° . Moreover, the literature review in the previous section showed that for many BBR's, both the orientation angles of the body and the position of the ball could be controlled with linear controllers. Based on these arguments, it is decided to design a linear controller for controlling both the orientation angles of the BBR and the position of the ball.

There are mainly three controller design techniques in classical and modern control theory for designing a linear controller, namely SISO loopshaping, LQR control theory and \mathcal{H}_∞ control theory.

SISO loopshaping is a controller design technique that manually shapes the transfer function from an input to an output. It gives good insight in the design of a controller, but it works only to MIMO systems that can be decoupled into independent SISO subsystems. Also, for large MIMO systems that can be decoupled, SISO loopshaping can be a time consuming technique, because all transfer functions between different inputs and outputs need to be controlled separately.

LQR control theory is a controller design technique that designs an optimal controller by minimizing a quadratic cost function subject to the system dynamics. It has the advantage that it can be applied to MIMO systems. A disadvantage of LQR control theory is that the control

parameters of LQR control theory are often not directly related to control requirements, which makes it more difficult to find the optimal controller that satisfies all the controller requirements. Finally, \mathcal{H}_∞ control theory is a controller design technique that designs a controller based on the minimization of the peak magnitude of a suitable closed loop system transfer function. It has the advantage that it can be applied to MIMO systems and it also has the advantage to include robustness constraints explicitly in the criterion.

Before a controller design approach will be formulated, the controller requirements will to be formulated.

4.2.1 Controller requirements

In order to have a well-designed control system, the controller should be designed such that:

1. It makes the system internally stable (i.e. all the closed loop poles have negative real parts) with a gain margin of at least 6 dB and a phase margin of at least 30° .
2. It attenuates disturbances up to 2 Hz for at least 80% (which is equal to an attenuation with at least -14 dB).
3. The closed loop system attenuates frequencies above 100 Hz for at least 80% (because the IMU has a sampling frequency of 200 Hz and the Nyquist-Shannon sampling theorem states that perfect reconstruction of a signal is possible when the sampling rate is greater than twice the maximum frequency of the signal).
4. The settling time² of the inverse step response³ of the pitch and roll angle is between 2 and 4 seconds in case it is station keeping at $(x_S, y_S)=(0, 0)$, to prevent that the system will behave too slow or too aggressive (so this controller requirement only holds for the controller that controls the pitch and roll angle).
5. Torque saturation is minimized.

4.2.2 Controller design approach

There are many possibilities in the design of a controller for the BBR, which is clearly visible in the literature overview, shown in Sec. 4.1. Different approaches are taken, based on e.g. the dynamics and size of the BBR or the purpose of the BBR.

The purpose of this BBR is to use it as a demonstrator on fairs as soon as possible. Therefore, the approach taken in this report is to design first a simple, static controller, which is easily implementable and has a short design time to make the BBR fair ready as soon as possible. For this controller, simplicity is more important than performance. Secondly, a more advanced dynamic controller will be designed, which will have a longer design time and might be more difficult to implement, but has more possibilities to address specific behaviour of the BBR and focuses more on performance. These controllers will be implemented and based on their performance it will be decided if it is necessary to design another controller, for example a nonlinear controller as is done in [2].

²The time after which the output remains within a range of 0.5° from 0°

³An inverse step response is a step response with an input offset of 10° and step amplitude of -10° at $t = 0$. It is used, because it is a response to a more natural disturbance than in case of a normal step response.

The controllers can both be designed with all three linear controller design techniques mentioned earlier in this section. In the design of the simple static controller, LQR control theory is chosen as controller design technique, because with LQR control theory a static controller can be calculated in a relatively short time. Although it is more difficult to find the optimal controller with LQR control theory (because the control parameters are not directly related to most of the controller requirements) it is relatively easy to find a suboptimal controller. LQR control theory is preferred above SISO loopshaping, because using SISO loopshaping as controller design technique results in a cascade controller, which makes the controller more complex. Moreover, designing a controller with SISO loopshaping takes more time, which is not in accordance with the design approach of the first controller to design a simple controller in a short time. LQR control theory is preferred above \mathcal{H}_∞ control theory, because LQR control theory is more suitable for the design of a static controller in a relatively short time.

In the design of the more advanced dynamic controller, SISO loopshaping is chosen as controller design technique. SISO loopshaping is possible because of a static decoupling of the inputs of the system. SISO loopshaping is preferred above LQR control theory, because for the design of a more advanced dynamic controller, it is desirable that the control parameters are directly related to most of the controller requirements to be able to achieve maximal performance, which is not the case for LQR control theory. Moreover, LQR control theory is based on the minimization of the \mathcal{L}_2 -norm⁴ of the error and input signals, which means that the peak values of the errors are not specifically taken into account. Also, Parseval's theorem⁵ states that the \mathcal{L}_2 -norm of a signal in time domain is proportional to the \mathcal{L}_2 -norm of that signal in frequency domain. This means that also in frequency domain, peak values are not specifically taken into account, which is undesirable for the design of a more advanced dynamic controller. Furthermore, SISO loopshaping is preferred above \mathcal{H}_∞ control theory, because of its possibilities to manually shape the controller, although \mathcal{H}_∞ control theory also is a suitable controller design technique for the design of this controller.

4.3 Design of a LQR controller

In this section a LQR controller will be designed that controls the orientation of the body and the position of the BBR simultaneously.

4.3.1 LQR control theory

The LQR controller consists of a feedback gain matrix K , which will be implemented as $\mathbf{u} = -K \cdot \mathbf{e}$ in the state space representation of the system, given in Eq. (3.49), where \mathbf{e} is the error defined as $\mathbf{e} = \mathbf{x}_{ref} - \mathbf{x}$, with \mathbf{x}_{ref} denoting the reference states. This results in a closed loop system with the reference states as input, shown in Fig. 4.1, for which the state space representation is given by

$$\begin{aligned}\dot{\mathbf{x}} &= (A - B \cdot K) \cdot \mathbf{x} + B \cdot K \cdot \mathbf{x}_{ref} \\ \mathbf{y} &= C \cdot \mathbf{x}\end{aligned}\tag{4.1}$$

where C is again the 10-by-10 identity matrix.

⁴The \mathcal{L}_2 -norm of a time function $f(t)$ is defined as $\|f(t)\|_{\mathcal{L}_2} = (\int_{-\infty}^{\infty} \|f(t)\|_2^2 dt)^{\frac{1}{2}}$, where $\|\cdot\|_2$ is the Euclidean norm.

⁵Parseval's theorem states that for a time function $f(t)$ holds that $\int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |F(i\omega)|^2 d\omega$, where $F(i\omega)$ is the Fourier transform of $f(t)$.

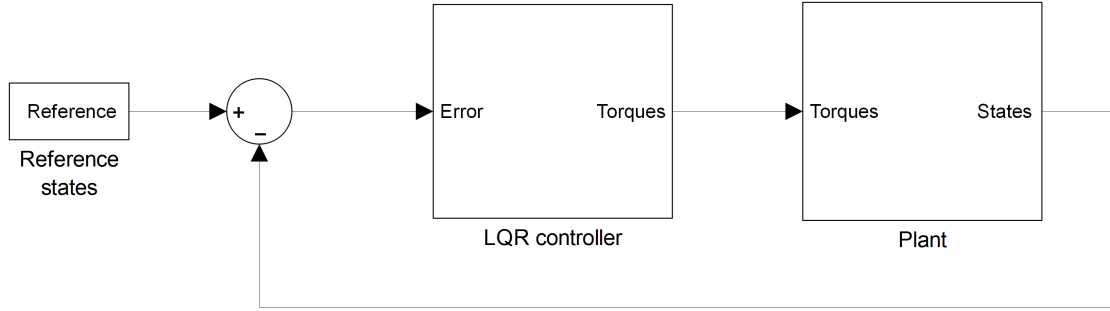


Figure 4.1: The closed loop system with a LQR controller.

The feedback gain matrix K is calculated such that $\mathbf{u} = -K \cdot \mathbf{e}$ minimizes the quadratic cost function

$$J(\mathbf{u}) = \int_0^{\infty} (\mathbf{e}^T \cdot Q \cdot \mathbf{e} + \mathbf{u}^T \cdot \rho \cdot R \cdot \mathbf{u}) dt \quad (4.2)$$

subject to the system dynamics

$$\dot{\mathbf{x}} = A \cdot \mathbf{x} + B \cdot \mathbf{u} \quad (4.3)$$

where Q is defined as $Q = N^T \cdot Q' \cdot N$, with N defining the plant outputs that need to be controlled and where Q' is the weighting matrix corresponding to these controlled outputs. Furthermore, R is the weighting matrix corresponding to the controller outputs and ρ is a positive constant.

In case $\rho \cdot R \gg Q$, the cost function is dominated by the control effort \mathbf{u} , so the controller focuses on minimizing the control action. In case $\rho \cdot R \ll Q$, the cost function is dominated by the errors \mathbf{e} , so the controller focuses on minimizing the errors of the states. So ρ is a scale factor that sets the ratio between optimizing the control effort and optimizing the states.

There is no such thing as a standard method of finding Q , R and ρ . Bryson's rule [4] is used for an initial value of the matrices Q' and R . With Bryson's rule, Q' and R are chosen diagonal with

$$\begin{aligned} Q'_{ii} &= \frac{1}{e_{i,max}^2} \\ R_{jj} &= \frac{1}{u_{j,max}^2} \end{aligned} \quad (4.4)$$

where $e_{i,max}$ is the maximum acceptable absolute value of the error with respect to the reference value of the controlled state i and $u_{j,max}$ is the maximum acceptable absolute value of input j . In fact, Bryson's rule normalizes the variables \mathbf{e} and \mathbf{u} to improve equal penalization of variables.

4.3.2 LQR control theory applied to the BBR

In order to apply the theory of LQR control to the BBR, the weighting matrices need to be determined together with control parameter ρ , such that the controller satisfies the controller requirements. Firstly, all states will be controlled, so N is a 10-by-10 identity matrix. Secondly, the maximum acceptable absolute values of the errors need to be estimated. In fact, of all the minimal coordinates (x_S , y_S , ψ_x , ψ_y and ψ_z), only the pitch and roll angle have a critical bound on the maximum acceptable absolute error and the other values need to be roughly estimated.

Therefore, the maximum error for the position of the ball is chosen to be 0.5 m and the maximum error for the pitch and roll angle is chosen to be 10° . There is no maximum error for the yaw angle required, so maximum error for the yaw angle is chosen to be the maximum possible error, which is 180° . The maximum errors for the derivatives of the minimal coordinates are taken thrice as much as the maximum errors of the minimal coordinates, which results in the following matrix Q' :

$$Q' = \begin{bmatrix} \frac{1}{0.5^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{0.5^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{180^2}{(10\pi)^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{180^2}{(10\pi)^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\pi^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{1.5^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{1.5^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180^2}{(30\pi)^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180^2}{(30\pi)^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3\pi^2} \end{bmatrix}. \quad (4.5)$$

The maximum torque of an omni-wheel is 3.75 Nm, which results in the following matrix R :

$$R = \begin{bmatrix} \frac{1}{3.75^2} & 0 & 0 \\ 0 & \frac{1}{3.75^2} & 0 \\ 0 & 0 & \frac{1}{3.75^2} \end{bmatrix}. \quad (4.6)$$

The feedback gain matrix K can now be calculated with MATLAB and optimized by changing ρ . Furthermore, a reference generator is implemented, which generates 4th order reference signals for the desired position of the ball to improve the position control.

The controlled system will be simulated for different values of ρ . The initial conditions for the states of Eq. (4.1) are chosen zero, except for the pitch angle ψ_x , which is chosen to be 10° to approximate the inverse step response, mentioned in controller requirement 4).

The results of the described simulation for the pitch angle is shown in Fig. 4.2 for $\rho = 0.01, 0.1, 1, 10, 100$.

Inspection of Fig. 4.2 shows that choosing ρ is a trade-off between aggressive motion resulting in a low settling time and smooth motion resulting in a higher settling time. Only for $\rho = 100$ and $\rho = 10$ the settling time does not satisfy controller requirement 4), but for all other values of ρ , the settling time is between 2 and 4 seconds. Therefore, to satisfy controller requirement 4) and also minimize the control effort, ρ is chosen as high as possible, provided that the settling time is between 2 and 4 seconds. Based on this reasoning, ρ is chosen to be 1, which results in the following gain matrix K :

$$K = \begin{bmatrix} 0 & -6.1237 & -40.0104 & 0 & -0.6892 & 0 & -10.7974 & -16.6234 & 0 & -0.2780 \\ 5.3033 & 3.0619 & 20.0052 & 34.6500 & -0.6892 & 9.3485 & 5.3974 & 8.3117 & 14.3963 & -0.2780 \\ -5.3033 & 3.0619 & 20.0052 & -34.6500 & -0.6892 & -9.3485 & 5.3974 & 8.3117 & -14.3963 & -0.2780 \end{bmatrix}. \quad (4.7)$$

In fact, this LQR controller is a sum of PD controllers for the minimal coordinates.

Whether the controller satisfies the other controller requirements will be analyzed in the frequency domain. To make the analysis in frequency domain easier, the plant inputs will be decoupled in

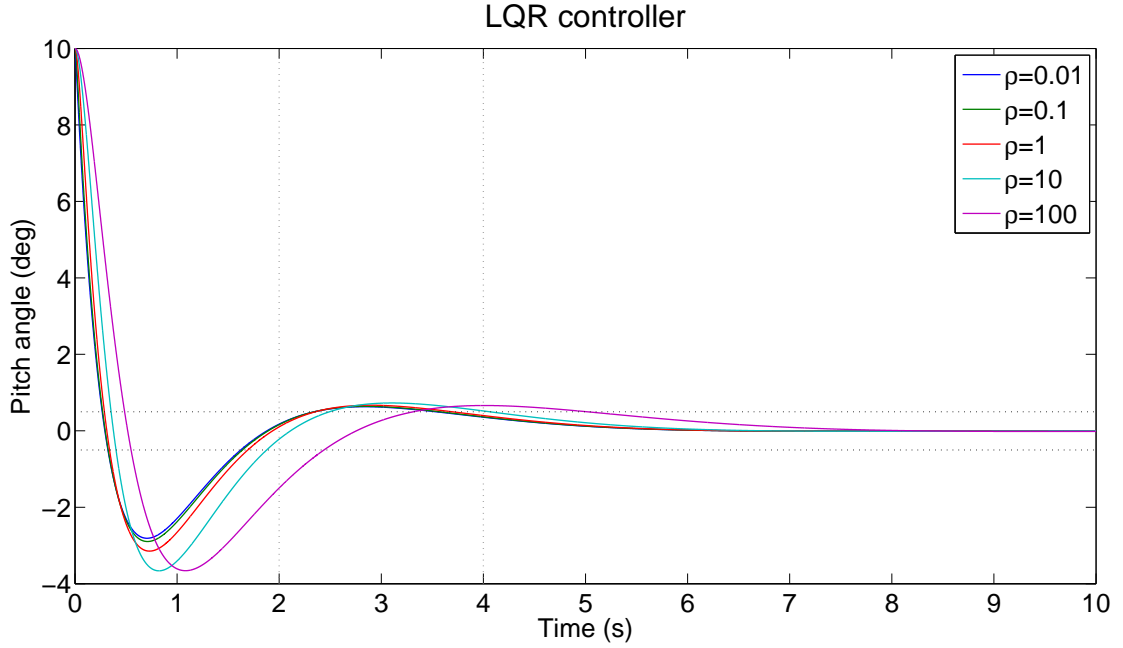


Figure 4.2: The inverse step response of the pitch angle simulated for different values of ρ , that determines the ratio between optimizing the control effort and optimizing the states.

x -, y - and z -direction and the LQR controller will be converted such that it controls the torques in x -, y - and z -direction. This makes it possible to convert the LQR controller to independent PD controllers for each of the minimal coordinates.

4.3.3 Decoupling of torques

The transformation matrix that decouples the torques in x -, y - and z -direction has already been derived in Sec. 2.6 for the 2D model and is given by Eq. (2.50). However, τ_z was defined clockwise for the 2D model, but counterclockwise for the 3D model, so the last column of the transformation matrix needs to be multiplied by -1 , because the controller design will be based on the 3D model. Substituting $\alpha = 45^\circ$ and $\beta = 0^\circ$, yields the following decoupling matrix M

$$M = \begin{bmatrix} \frac{2\sqrt{2}}{3} & 0 & -\frac{\sqrt{2}}{3} \\ -\frac{\sqrt{2}}{3} & \frac{\sqrt{6}}{3} & -\frac{\sqrt{2}}{3} \\ -\frac{\sqrt{2}}{3} & -\frac{\sqrt{6}}{3} & -\frac{\sqrt{2}}{3} \end{bmatrix} \quad (4.8)$$

such that

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = M \cdot \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}. \quad (4.9)$$

The state space representation of the system was given by Eq. (3.49) in Sec. 3.6. Post-multiplying

matrix B with M results in a state space model with the new inputs τ_x , τ_y and τ_z

$$\begin{aligned}\dot{\mathbf{x}} &= A \cdot \mathbf{x} + B \cdot M \cdot \tilde{\mathbf{u}} \\ \mathbf{y} &= C \cdot \mathbf{x}\end{aligned}\quad (4.10)$$

with $\tilde{\mathbf{u}} = [\tau_x \ \tau_y \ \tau_z]^T$ and where C is the 10-by-10 identity matrix.

4.3.4 Verification of controller requirements in frequency domain

The plant transfer matrix P can now be calculated by taking the Laplace Transform of the state space model, which is defined as

$$P = C \cdot (s \cdot Id - A)^{-1} \cdot B \cdot M \quad (4.11)$$

where Id denotes the 10-by-10 identity matrix.

Pre-multiplying the controller gain matrix K with M^{-1} results in a controller that controls the torques in x -, y - and z -direction. Combining the gains of the minimal coordinates and their derivatives into one controller for each of the minimal coordinates results in a PD controller for each of the minimal coordinates. With these PD controllers, together with the plant transfer functions given in Eq. (4.11), all the different transfer functions, needed for the verification of the controller requirements, can be calculated.

To check whether the controller satisfies the second controller requirement, the sensitivity functions of the minimal coordinates are plotted in Fig. 4.3.

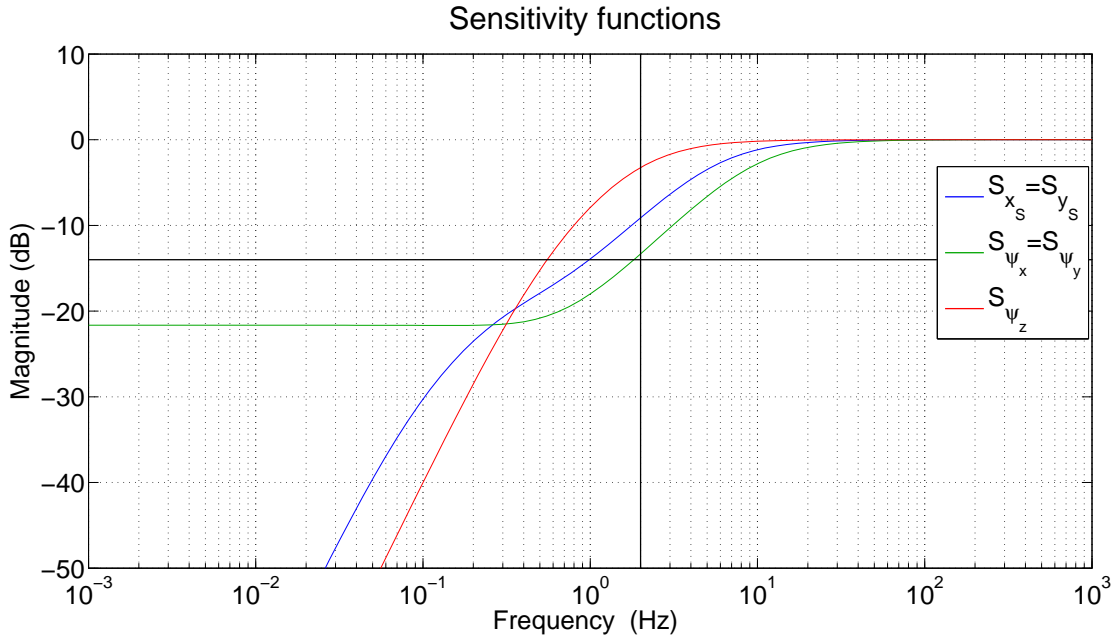


Figure 4.3: Bode magnitude plot of the sensitivity functions of the minimal coordinates (in case $\rho = 1$).

Fig. 4.3 shows that for none of the minimal coordinates, the controller satisfies controller requirement 2), because all the sensitivity functions cross the -14 dB line before 2 Hz. For controlling the yaw angle, this problem is solved by reducing the maximum error for the yaw angle from 180° to 60° . For controlling the other four minimal coordinates, this problem can be solved by increasing the bandwidth (as little as possible). This can be done by decreasing ρ from 1 to 0.25, which is proved in Fig. 4.4 (Note that the sensitivity functions S_{x_s} and S_{ψ_x} are identical to respectively S_{y_s} and S_{ψ_y}).

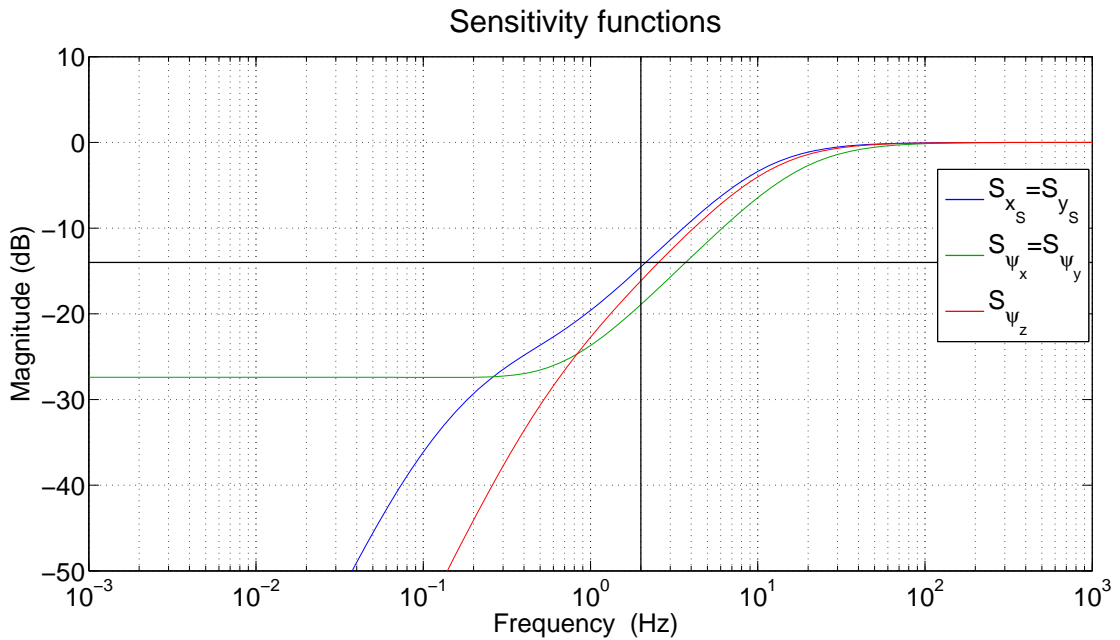


Figure 4.4: Bode magnitude plot of the sensitivity functions of the minimal coordinates (in case $\rho = 0.25$).

To check whether the controller also satisfies the third controller requirement, the closed loop transfer functions are plotted in Fig. 4.5. Fig. 4.5 shows that all of the closed loop transfer functions satisfy controller requirement 3), because they attenuate frequencies above 100 Hz with at least -14 dB.

To make sure that torque saturation is minimized with this controller, three scenarios will be considered.

The first scenario is that torque saturation will never happen. In that case the controller automatically satisfies controller requirement 5) and no additional actions need to be taken.

The second scenario is that torque saturation will happen sometimes. In that case it needs to be investigated how frequently torque saturation happens and based on that it can be decided if additional actions need to be taken.

The third scenario is that torque saturation will often happen and in that case additional actions definitely need to be taken to minimize the torque saturation.

To predict which scenario will be most probable for the BBR, two different situations will be simulated. In the first simulation a natural situation will be simulated to get an indication how large the average needed torque is. In the second simulation an exceptional situation will be

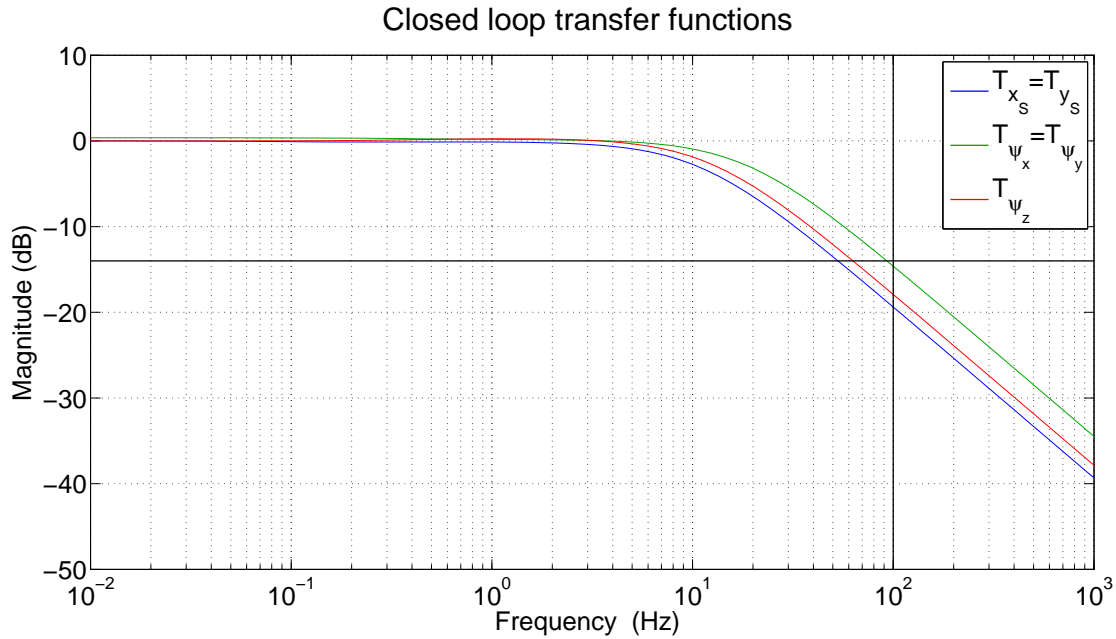


Figure 4.5: Bode magnitude plot of the closed loop transfer functions of the minimal coordinates (in case $\rho = 0.25$).

simulated to get an indication whether or not the torque will saturate in these kind of situations. In the first simulation, the closed loop system is simulated with the nonlinear model in Simulink for 5 seconds with a sinusoidal disturbance d for the roll angle, with a frequency of 0.5 Hz and an amplitude of 5° , which is for now assumed to be a natural disturbance signal for the BBR. The references are set to zero for all states, which means that the BBR is balancing and station keeping at the position $(x_S, y_S) = (0, 0)$. The disturbance signal together with the torque required to stabilize the system in this situation, is shown in Fig. 4.6.

The dotted gray lines indicate the nominal torque of 1.74 Nm and the dotted red lines indicate the maximum torque of 3.75 Nm.

In the second simulation, the closed loop system is simulated with the nonlinear model in Simulink for 5 seconds with a sinusoidal disturbance d for the roll angle, with a frequency of 0.5 Hz and an amplitude of 15° , which is for now assumed to represent an exceptional situation for the BBR. The references are set to zero for all states, which means that the BBR is balancing and station keeping at the position $(x_S, y_S) = (0, 0)$. The disturbance signal together with the torque required to stabilize the system in this situation, is shown in Fig. 4.7.

Fig. 4.6 shows that in case of a natural disturbance signal, the required torques stay far under the nominal torque. Furthermore, Fig. 4.7 shows that in case of an exceptional disturbance signal, the required torques stay far under the maximum torque. So the results of these simulations give a strong indication that the first scenario is most probable for the BBR. Therefore, no additional actions seem to be needed to improve the minimization of torque saturation and the controller satisfies controller requirement 5).

The last controller requirement that needs to be satisfied is controller requirement 1). Therefore,

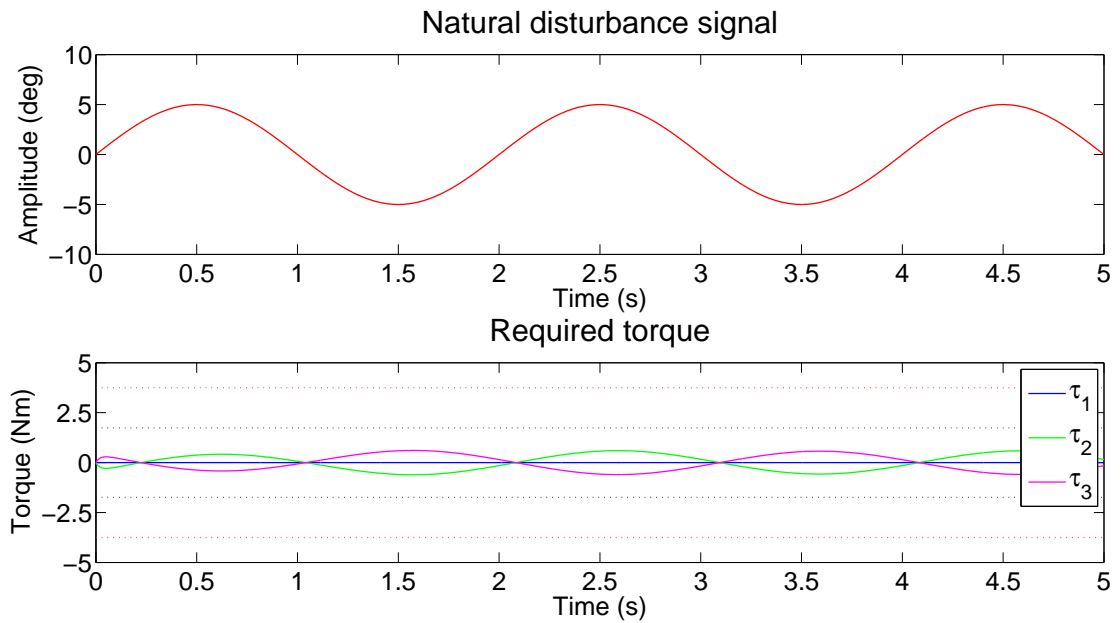


Figure 4.6: The required torque in case of a natural disturbance signal with a frequency of 0.5 Hz and an amplitude of 5° .

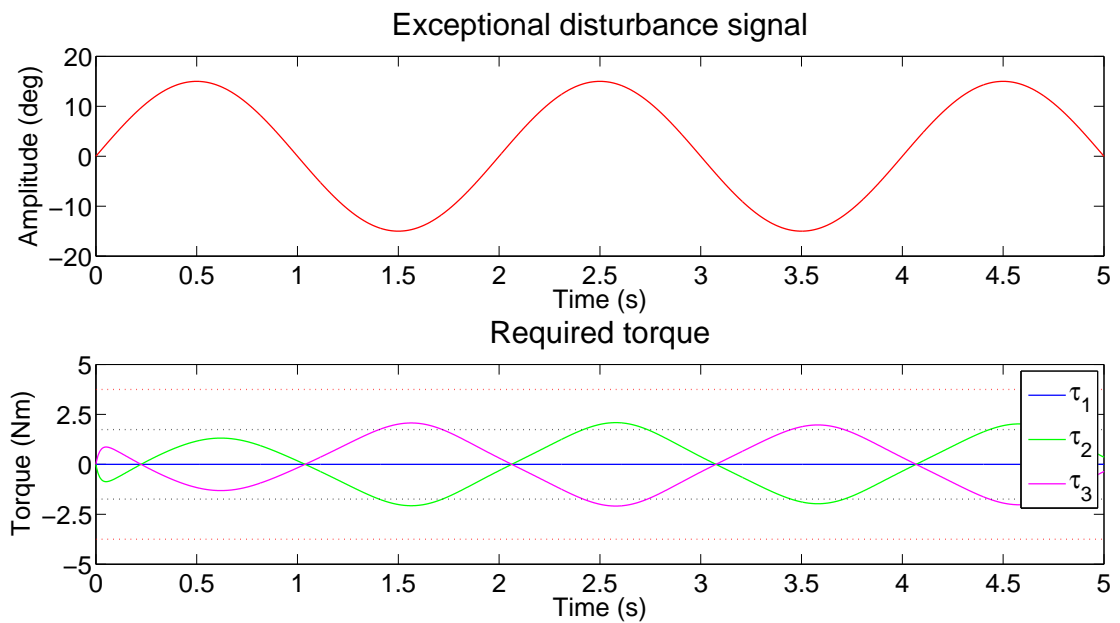


Figure 4.7: The required torque in case of a exceptional disturbance signal with a frequency of 0.5 Hz and an amplitude of 15° .

the closed loop poles are calculated and given by (in rad/s)

$$\begin{aligned}
 s &= -0.9284 \pm 0.6942i \\
 s &= -0.9284 \pm 0.6942i \\
 s &= -3.0005 \\
 s &= -3.0005 \\
 s &= -3.0023 \\
 s &= -45.9959 \\
 s &= -45.9959 \\
 s &= -77.4180
 \end{aligned} \tag{4.12}$$

which proves that the closed loop system is stable. Also, LQR controllers have an infinite gain margin and a phase margin of at least 60° (see [18] for the proof), so the controller also satisfies controller requirement 1).

The final gain matrix K is given by

$$K = \begin{bmatrix} 0 & -12.2474 & -74.7196 & 0 & -4.1350 & 0 & -21.2709 & -31.5757 & 0 & -1.4307 \\ 10.6066 & 6.1237 & 37.3598 & 64.7091 & -4.1350 & 18.4212 & 10.6355 & 15.7878 & 27.3453 & -1.4307 \\ -10.6066 & 6.1237 & 37.3598 & -64.7091 & -4.1350 & -18.4212 & 10.6355 & 15.7878 & -27.3453 & -1.4307 \end{bmatrix}. \tag{4.13}$$

Finally, the closed loop system will be simulated with a initial position of (0,0) and a desired position of (3,1) to illustrate the performance of the position controller. The results are shown in Fig. 4.8.

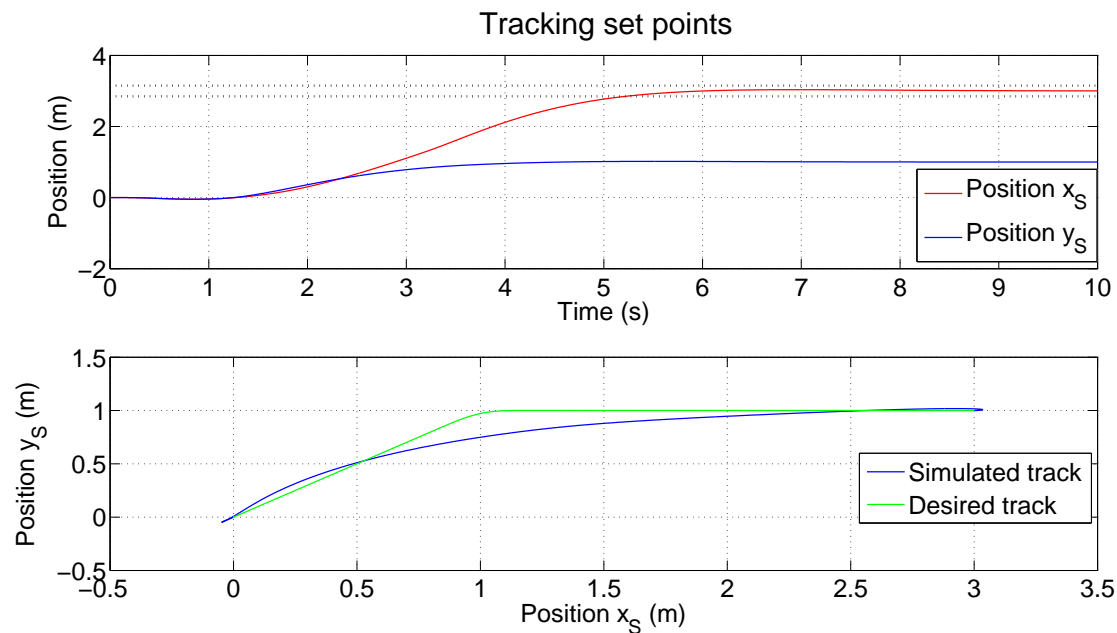


Figure 4.8: Simulation of position of the BBR from position (0,0) to (3,1) with the LQR controller.

Fig. 4.8 shows that the desired position (3,1) is reached in about 6 seconds, which means that the BBR has a linear speed of about 0.5 m/s.

4.4 Design of a SISO controller

In this section a controller, denoted as a SISO controller, will be designed with manual loopshaping of the transfer functions between the different inputs and outputs, that controls both the orientation of the body and the position of the ball.

In order to design a SISO controller, the inputs of the plants will be decoupled in the x -, y - and z -direction, like is done in Sec. 4.3.3. This results in a diagonal plant matrix, which makes it possible to control each of the minimal coordinates separately. The control system consists of an inner loop that controls the orientation angles of the body and an outer loop that controls the position of the ball.

The inner control loop consists of three SISO controllers for the pitch, roll and yaw angle. Firstly, a SISO controller will be designed for the pitch angle ψ_x with manual loopshaping using ShapeIt⁶. The controller for the roll angle ψ_y is then obtained by multiplying the controller for the pitch angle ψ_x by -1. Secondly, a SISO controller will be designed for the yaw angle.

After that the outer control loop will be designed, which consists of two SISO controllers for the x - and y -coordinates of the ball. These position controllers actuate the pitch and roll angle in order to move the BBR to the desired position.

A sketch of this control system can be found in Fig. 4.9.

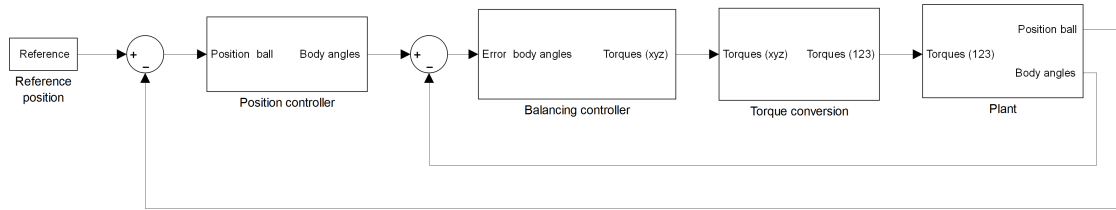


Figure 4.9: Sketch of the closed loop system with a SISO controller.

4.4.1 Calculation of SISO plant transfer functions

The inputs of the plant can be decoupled in the x -, y - and z -direction using the decoupling matrix M , like is explained in Sec. 4.3.3. In this case, C is defined as

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.14)$$

such that the pitch, roll and yaw angles are the only outputs.

The transfer matrix P , that maps the inputs to the outputs, can be found by taking the Laplace Transform of this state space model. It is defined as

$$P = C \cdot (s \cdot Id - A)^{-1} \cdot B \cdot M \quad (4.15)$$

⁶http://cstwiki.wtb.tue.nl/index.php?title=Home_of_ShapeIt

where Id denotes the 10-by-10 identity matrix. Calculating this transfer matrix P with MATLAB gives the following input/output-representation

$$\begin{bmatrix} \psi_x \\ \psi_y \\ \psi_z \end{bmatrix} = \begin{bmatrix} \frac{-3.544}{s^2-11.49} & 0 & 0 \\ 0 & \frac{3.544}{s^2-11.49} & 0 \\ 0 & 0 & \frac{26.5}{s^2} \end{bmatrix} \cdot \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}. \quad (4.16)$$

This proves that, because of the decoupling, the plant matrix is diagonal and the pitch, roll and yaw angle can be controlled independently.

4.4.2 Controller design for pitch and roll angle

The transfer function from τ_x to the pitch angle ψ_x (denoted by P_{ψ_x}) is thus given by

$$P_{\psi_x}(s) = \frac{-3.544}{(s + 3.3894)(s - 3.3894)} \quad (4.17)$$

which has no zeros and poles at $s = \pm 3.3894$ rad/s. A Bode plot of P_{ψ_x} is shown in Fig. 4.10.

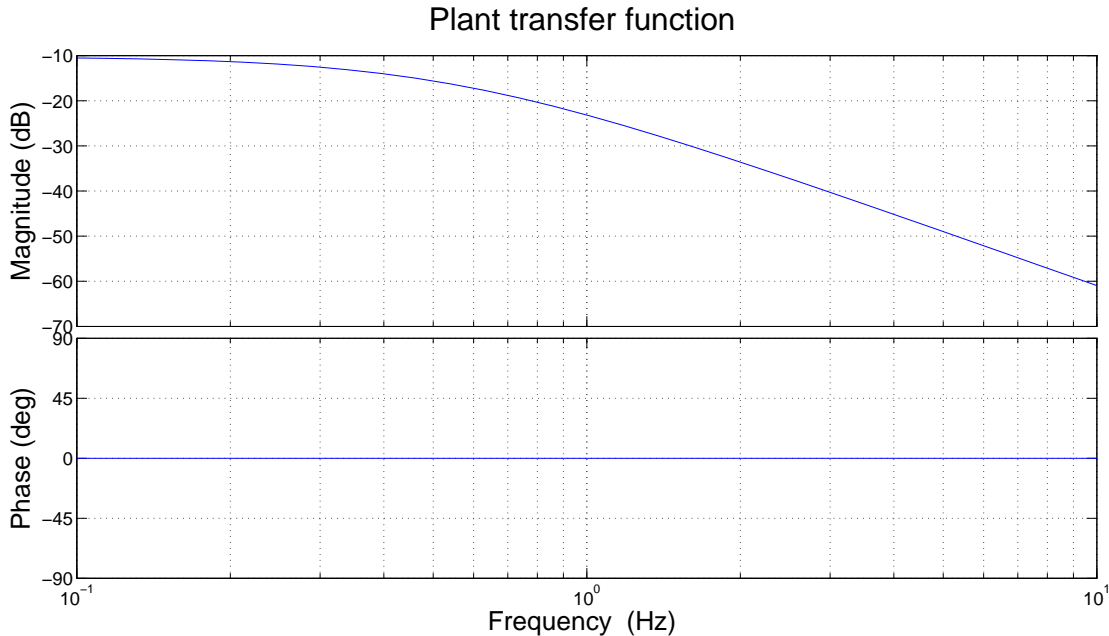


Figure 4.10: Bode plot of plant transfer function $P_{\psi_x}(s)$.

Now a controller will be designed for this plant. By using a pure gain controller, with gain $K_{\psi_x} = -100$, the Bode plot of the closed loop transfer function $T_{\psi_x} = P_{\psi_x} K_{\psi_x} \cdot (1 + P_{\psi_x} K_{\psi_x})^{-1}$ shows satisfactory tracking of reference signal by the output signal up to a frequency of about 1 Hz, as can be seen in Fig. 4.11. However, T_{ψ_x} also has a large peak at about 3 Hz and the phase margin is 0° . This problem can be solved by introducing a lead compensator of the form

$$K_{\psi_x}(s) = \frac{\frac{n}{2\pi\omega_B} s + 1}{\frac{1}{2\pi n\omega_B} s + 1} \quad (4.18)$$

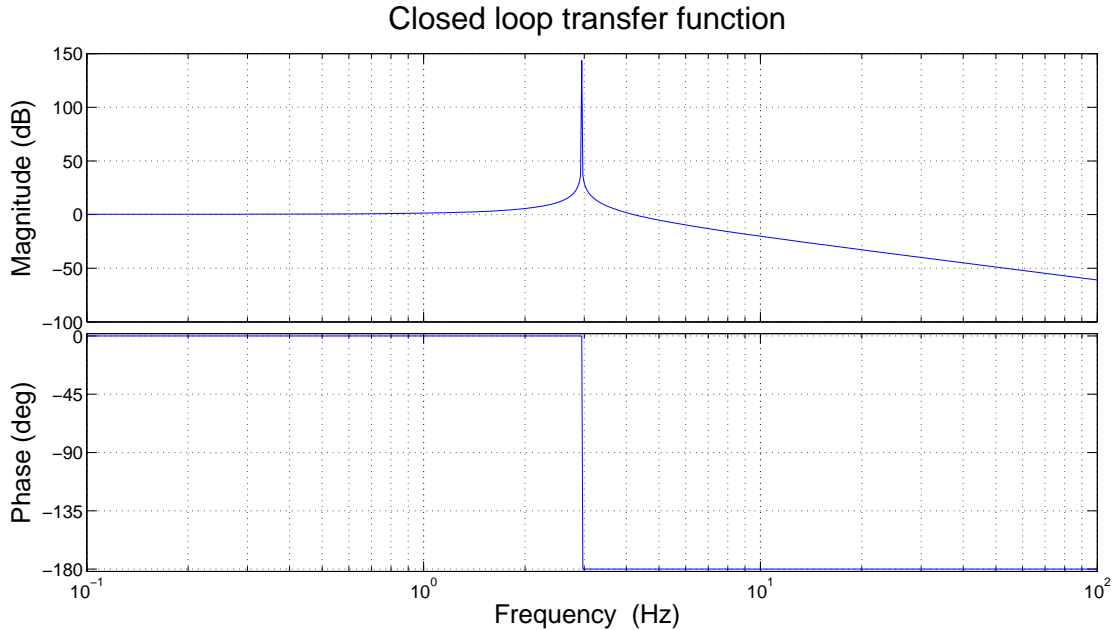


Figure 4.11: Bode plot of closed loop transfer function $T_{\psi_x}(s)$.

where n sets the frequencies where the zero and pole are placed and where ω_B denotes the bandwidth⁷. This form of lead compensator makes sure that the maximum phase lead occurs at the bandwidth ω_B . It is important to keep the bandwidth as low as possible, because of unmodeled vibrations, risk of amplification of measurement noise and large control signals. Based on the requirements that disturbances up to 2 Hz need to be rejected, a bandwidth of 3 Hz is used as a starting point.

Based on a bandwidth of 3 Hz, a lead compensator is designed for different n . The impulse response of T_{ψ_x} and a Bode plot of the sensitivity function $S_{\psi_x} = (1 + P_{\psi_x}K_{\psi_x})^{-1}$ are shown in Fig. 4.12 for $n = 2, 3, 4, 5, 6$.

To prevent the system from reacting too aggressive on disturbances, the first peak of the impulse response of T_{ψ_x} , shown in Fig. 4.12a) should be as low as possible. Also, the higher this peak is, the more control effort is needed, so this peak should be minimized to satisfy controller requirements 4) and 5). Inspection of Fig. 4.12a) shows that for $n = 3$ the peak is minimal and that it increases if n increases. So, based on this reasoning, $n = 3$ would be the best choice.

Inspection of Fig. 4.12b) shows that, in order to minimize the peak of S_{ψ_x} , n should be chosen as high as possible. However, minimizing the peak of S_{ψ_x} causes an increase of S_{ψ_x} at low frequencies, which is called the 'waterbed effect'. In order to satisfy controller requirement 2), n can be at most 5 to attenuate disturbances up to 2 Hz with at most -14 dB. So, based on this reasoning, $n = 5$ would be the best choice.

So choosing n is a trade-off between optimizing the settling time and minimizing the overshoot on the one hand and having a ideally shaped sensitivity function to prevent amplification of noise

⁷In this report, the bandwidth ω_B is defined as the frequency, expressed in Hz, where the open loop gain is unity.

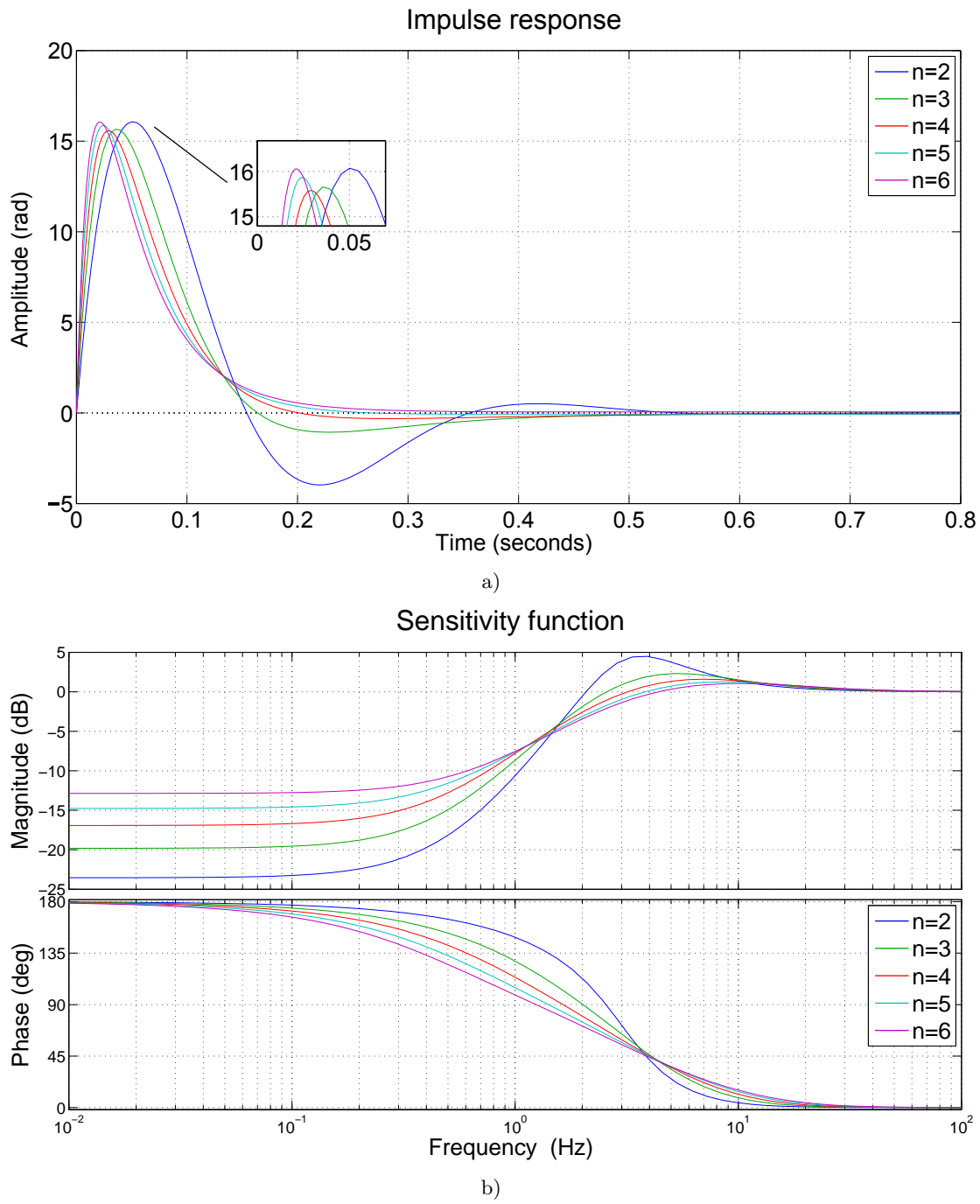


Figure 4.12: Simulation for different values of n , that sets the frequencies of the pole and the zero of the lead compensator. a) The impulse response of the closed loop transfer function $T_{\psi_x}(s)$. b) Bode plot of the sensitivity function $S_{\psi_x}(s)$.

on the other hand. For $n = 4$, the impulse response of T_{ψ_x} converges and the first peak is of about the same height as for $n = 3$, compared to other values of n . Also, for $n = 4$, peaking of S_{ψ_x} is acceptable. Therefore, n is chosen to be 4.

Furthermore, Fig. 4.12b) shows that the bandwidth needs to be increased to satisfy controller requirement 2), i.e. to achieve a disturbance attenuation of at least 80% up to disturbances of 2 Hz. Simulations show that the bandwidth needs to be increased to 9 Hz, which is proved by Fig. 4.13.

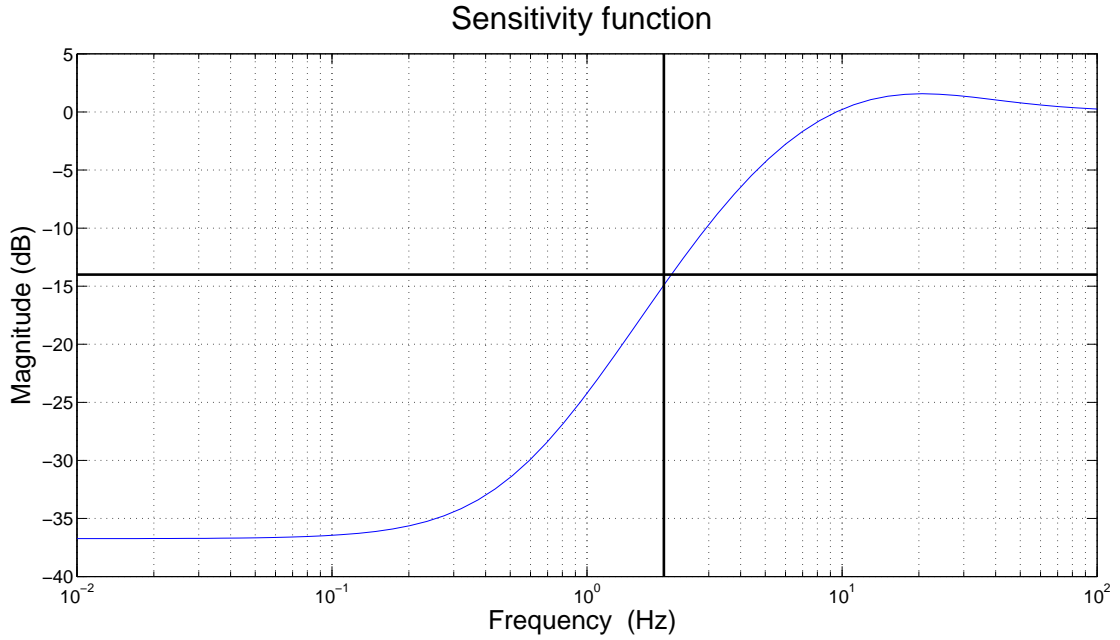


Figure 4.13: Bode magnitude plot of the sensitivity function $S_{\psi_x}(s)$.

This results in the following controller:

$$K_{\psi_x}(s) = -226 \cdot \frac{0.07074s + 1}{0.004421s + 1} \quad (4.19)$$

where the gain has been adjusted to -226 to achieve the required bandwidth of 9 Hz.

To check whether the controller satisfies also the third controller requirement, the closed loop transfer function T_{ψ_x} is plotted in Fig. 4.14.

Fig. 4.14 shows that the closed loop system attenuates frequencies above 100 Hz with at least -30 dB, which is equal to an attenuation for at least 95%, so the controller clearly satisfies controller requirement 3).

The next thing that will be verified, is the stability of the closed loop system to check whether the controller satisfies controller requirement 1). Therefore, the closed loop poles will be calculated. They are given by (in rad/s)

$$\begin{aligned} s &= -147.6878 \\ s &= -57.4668 \\ s &= -21.0401 \end{aligned} \quad (4.20)$$

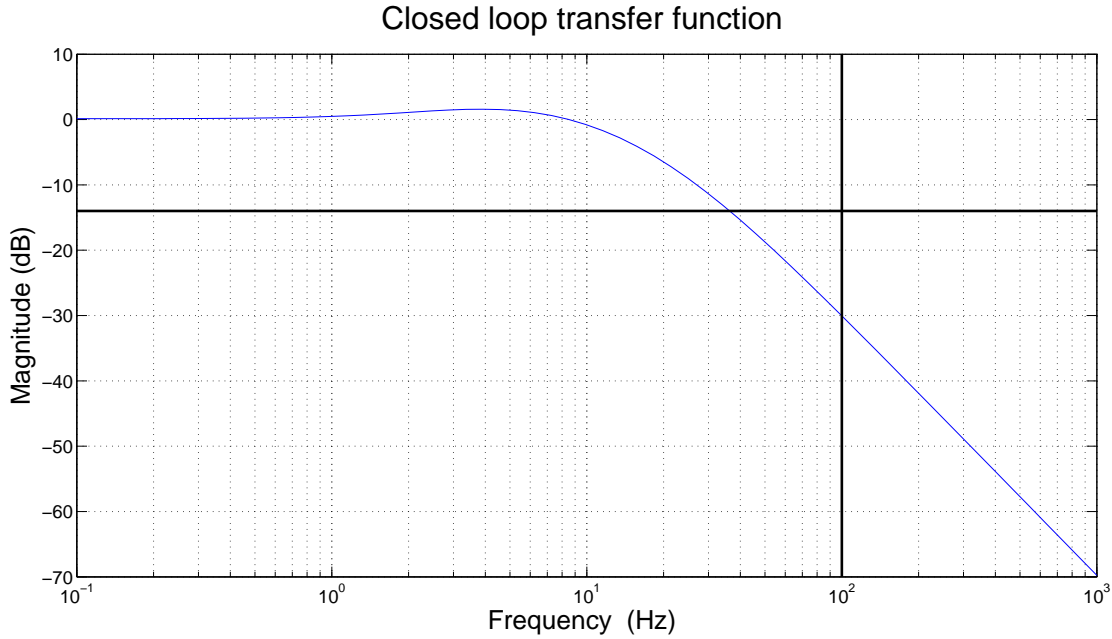


Figure 4.14: Bode magnitude plot of closed loop transfer function $T_{\psi_x}(s)$.

which proves that the closed loop system is stable. Furthermore, it has an infinite gain margin and a phase margin of 62° , so the controller also satisfies controller requirement 1). So until now, it has been proved that the controller for the pitch and roll angle satisfies controller requirements 1) up to 3). Controller requirements 4) and 5) will be verified later with the total controlled system.

4.4.3 Controller design for yaw angle

The transfer function from τ_z to the yaw angle ψ_z (denoted by P_{ψ_z}) is given by

$$P_{\psi_z}(s) = \frac{26.5}{s^2} \quad (4.21)$$

which has no zeros and two poles at $s = 0$ rad/s. A Bode plot of P_{ψ_z} is shown in Fig. 4.15.

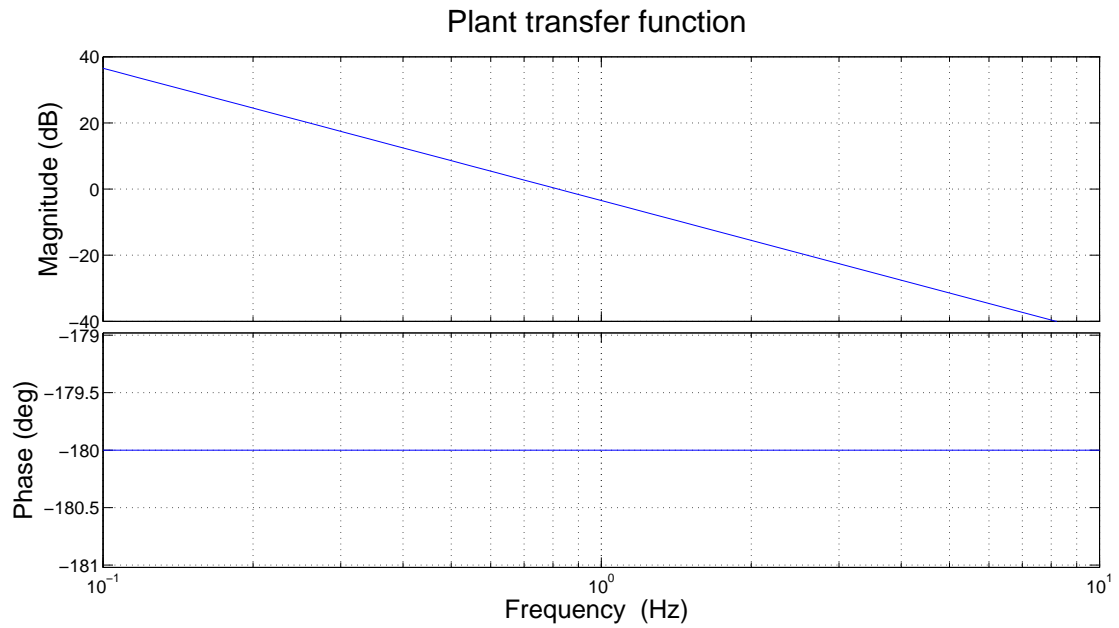
Inspection of Fig. 4.15 shows that phase margin is 0° . Analogous to the design of a controller for the pitch and roll angle, this is solved by introducing a lead compensator of the form

$$K_{\psi_z}(s) = \frac{\frac{4}{2\pi\omega_B}s + 1}{\frac{1}{2\pi 4\omega_B}s + 1}. \quad (4.22)$$

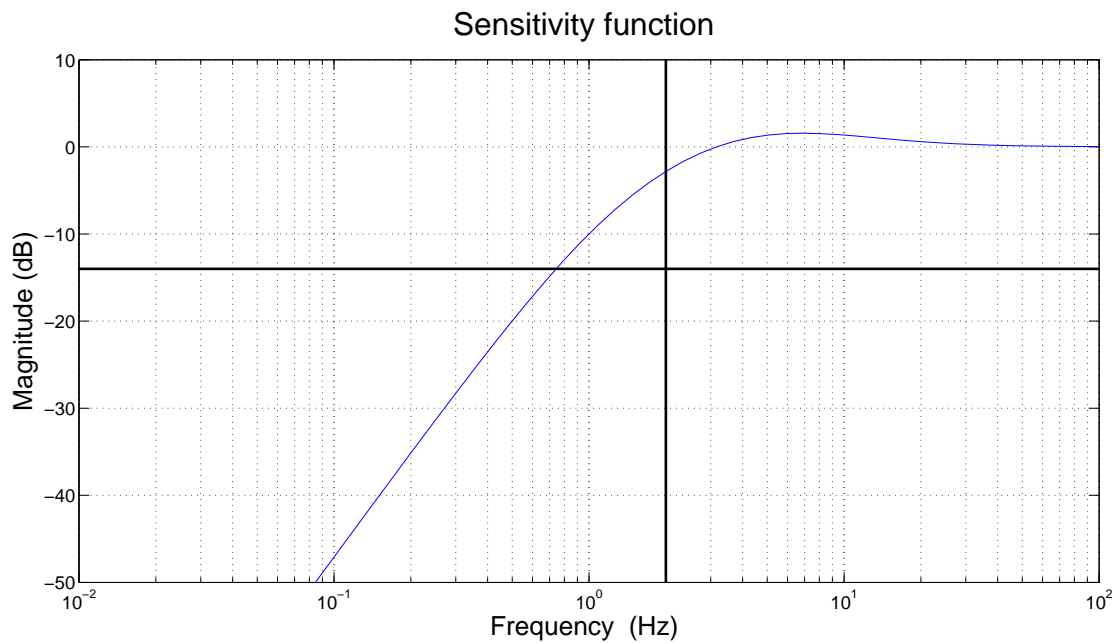
Also, a bandwidth of 3 Hz is used as a starting point, which results in the following controller:

$$K_{\psi_z}(s) = 3.35 \cdot \frac{0.2122s + 1}{0.01326s + 1}. \quad (4.23)$$

where the gain has been chosen such that the bandwidth is 3 Hz.

Figure 4.15: Bode plot of plant transfer function $P_{\psi_z}(s)$.

In Fig. 4.16 the sensitivity function S_{ψ_z} is plotted to check whether the controller satisfies controller requirement 2).

Figure 4.16: Bode magnitude plot of the sensitivity function $S_{\psi_z}(s)$ with a bandwidth of 3 Hz.

Inspection of Fig. 4.16 shows that frequencies up to 2 Hz are not attenuated for at least 80%. This problem is solved by increasing the bandwidth to 9 Hz, which is proved by Fig. 4.17. This

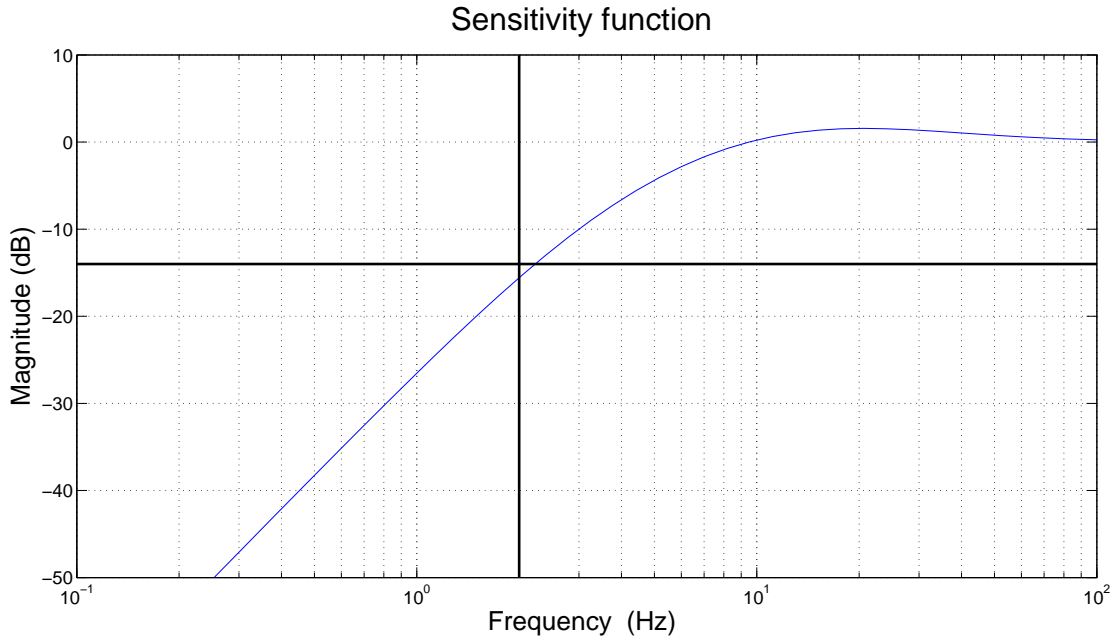


Figure 4.17: Bode magnitude plot of the sensitivity function $S_{\psi_z}(s)$ with a bandwidth of 9 Hz.

results in the following controller:

$$K_{\psi_z}(s) = 30.15 \cdot \frac{0.07074s + 1}{0.004421s + 1}. \quad (4.24)$$

where the gain has been chosen such that the bandwidth is 9 Hz.

To check whether the controller satisfies controller requirement 3), the closed loop transfer function T_{ψ_z} is plotted in Fig. 4.18.

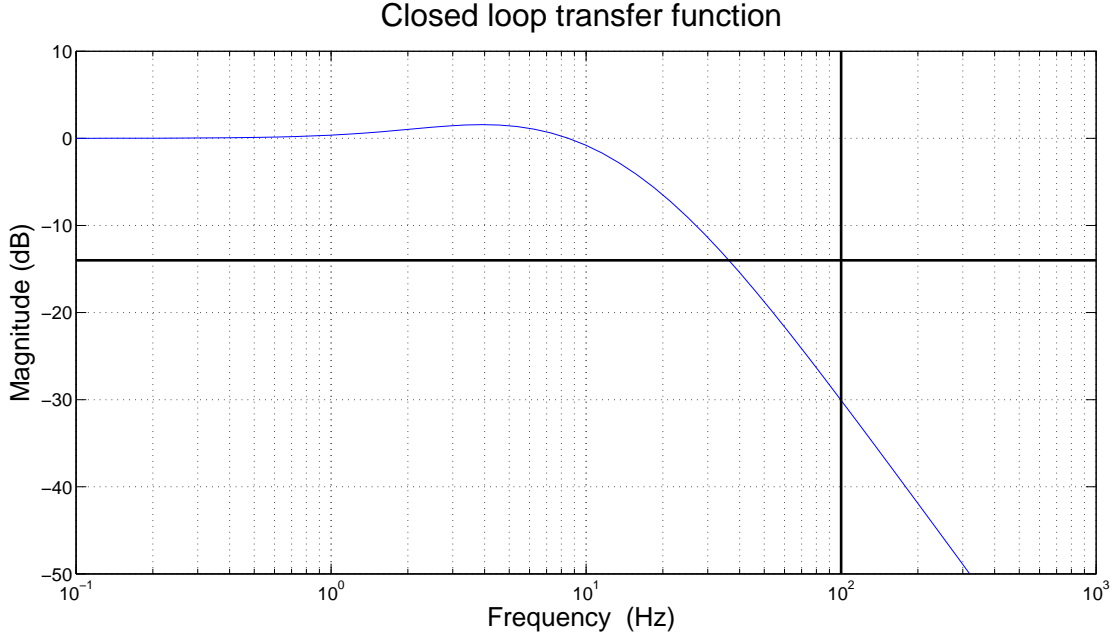
Fig. 4.18 shows that the closed loop system attenuates frequencies above 100 Hz for more than 80%, so the controller also satisfies controller requirement 3).

Finally, the stability of the closed loop system needs to be verified. Therefore, the closed loop poles are calculated and they are given by (in rad/s)

$$\begin{aligned} s &= -148.1320 \\ s &= -56.4505 \\ s &= -21.6122 \end{aligned} \quad (4.25)$$

which proves that the closed loop system is stable. Furthermore, it has an infinite gain margin and a phase margin of 62° , so the controller also satisfies controller requirement 1).

So controller requirements 1) up to 3) are satisfied now and controller requirement 4) and 5) will be verified later with the total controlled system.

Figure 4.18: Bode magnitude plot of closed loop transfer function $T_{\psi_x}(s)$.

4.4.4 Controller design for position of the ball

In this paragraph, controllers will be designed that control the x - and y -position of the ball by setting the reference of the pitch and roll angle. Therefore, the transfer function from the reference signals of the pitch and roll angle to the position of the ball need to be calculated.

The transfer function from r_{ψ_y} to the x -position of the ball x_S (denoted by P_{x_S}) is defined as the transfer function from r_{ψ_y} to ψ_y (denoted by $T_{\psi_y} = \frac{P_{\psi_y} K_{\psi_y}}{1 + P_{\psi_y} K_{\psi_y}}$) multiplied by the transfer function from ψ_y to x_S (denoted by $P_{\psi_y \mapsto x_S}$), which can be seen from the closed loop system of x_S in Fig. 4.19. P_{x_S} is given by

$$P_{x_S}(s) = \underbrace{\frac{12815s + 181169}{s^3 + 226.2s^2 + 12804s + 178570}}_{T_{\psi_y}} \cdot \underbrace{\frac{-0.8485s^2 + 4.029}{s^2}}_{P_{\psi_y \mapsto x_S}} \quad (4.26)$$

where $P_{\psi_y \mapsto x_S}$ is defined as

$$P_{\psi_y \mapsto x_S} = \frac{P_{\tau_y \mapsto x_S}}{P_{\tau_y \mapsto \psi_y}} \left(\frac{\text{Transfer function from } \tau_x \text{ to } x_S}{\text{Transfer function from } \tau_x \text{ to } \psi_y} \right). \quad (4.27)$$

The transfer function from r_{ψ_x} to the y -position of the ball y_S (denoted by P_{y_S}) is identical to P_{x_S} and therefore only P_{x_S} will be considered in this paragraph.

A Bode plot of P_{x_S} is shown in Fig. 4.20. It has zeros (in Hz) at

$$\begin{aligned} s &= \pm 0.34683 \\ s &= -2.25 \end{aligned} \quad (4.28)$$

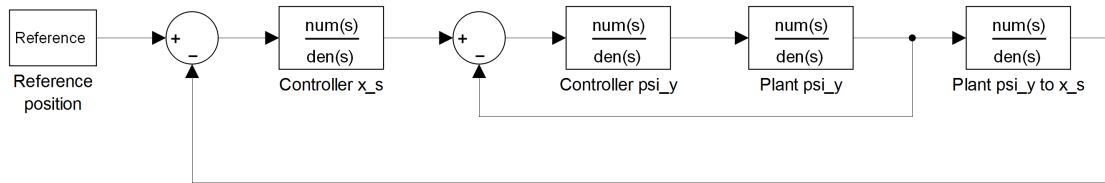


Figure 4.19: Sketch of the closed loop system for x_s .

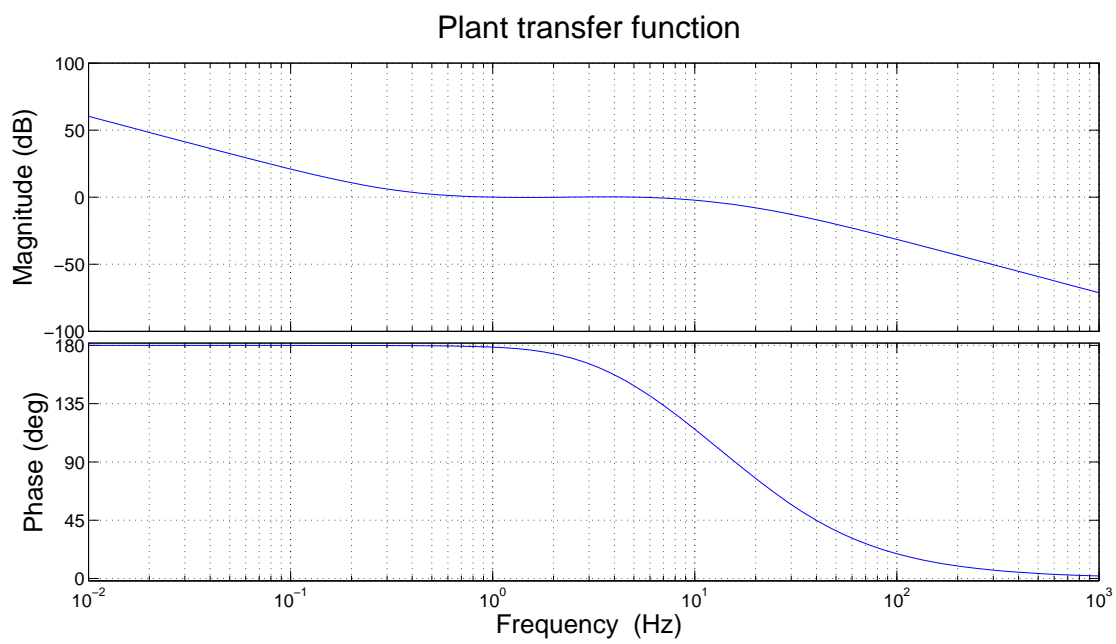


Figure 4.20: Bode plot of plant transfer function $P_{x_s}(s)$.

and poles (in Hz) at

$$\begin{aligned} s &= 0 \\ s &= 0 \\ s &= -3.3486 \\ s &= -9.1461 \\ s &= -23.505. \end{aligned} \tag{4.29}$$

Due to the right-half plane (RHP) zero, P_{x_S} is non-minimum phase. This constrains the performance of the system as the frequency over which effective disturbance attenuation is possible, without excessive peaking of the sensitivity function (denoted by S_{x_S}), is constrained from above by the magnitude of the smallest RHP zero [1], i.e. 0.34683 Hz. The RHP zero thus also constrains the maximum bandwidth.

To make sure that excessive peaking of S_{x_S} is prevented, an additional requirement is introduced by requiring a modulus margin of at least 0.5, because the modulus margin is defined as $\frac{1}{\max|S_{x_S}(s)|}$. This means that S_{x_S} must be at most 6 dB.

Inspection of Fig. 4.20 shows that P_{x_S} has a phase of between 90° and 180° in the crossover region, which makes the closed loop system unstable with a unity controller. Moreover, zooming in the Bode magnitude plot of P_{x_S} , see Fig. 4.21, shows that P_{x_S} crosses the 0 dB line multiple times, which causes peaking of the sensitivity function S_{x_S} and the closed loop transfer function T_{x_S} . Adjusting the gain does not solve the problem of multiple crossing of the 0 dB line, because increasing the gain results in a too high bandwidth and decreasing the gain results in excessive peaking of both S_{x_S} and T_{x_S} . Therefore, a pole is added by introducing a low-pass filter:

$$K_{lp,x_S}(s) = \frac{1}{\frac{1}{2\pi\omega_{lp}}s + 1} \tag{4.30}$$

where ω_{lp} is the frequency of the pole of the low-pass filter. The frequency of the pole is chosen to be the same frequency as the frequency of the RHP zero of P_{x_S} to minimize peaking of both S_{x_S} and T_{x_S} , which results in a slope of -1 in the crossover region. The Bode plot of the resulting open loop transfer function L_{x_S} is shown in Fig. 4.22.

However, adding this pole causes even more phase lag in the crossover region, as can be seen in Fig. 4.22. To add phase to the system, a lead compensator is introduced of the form

$$K_{lc,x_S}(s) = \frac{\frac{1}{2\pi\omega_z}s + 1}{\frac{1}{2\pi\omega_p}s + 1} \tag{4.31}$$

where ω_z is the frequency of the zero and where ω_p is the frequency of the pole, both expressed in Hz. To maximize the phase lead, the frequencies of the zero and the pole should not be too close. Moreover, Fig. 4.22 shows that the phase of L_{x_S} decreases as the frequency increases. Therefore to achieve a phase margin of at least 30° , the phase of L_{x_S} must be at least 120° at the frequency at which maximum phase lead will occur, so the maximum frequency at which maximum phase lead may occur is 0.59 Hz. Note that a double lead compensator is not preferred, because it again creates a resonance in the crossover region.

Finding the optimal values of ω_p and ω_z is not straightforward. Initially, ω_p is set to a frequency that is for sure higher than the maximum bandwidth, e.g. 2 Hz. Then, for different values of ω_z the Bode plot of L_{x_S} is shown in Fig. 4.23 together with the step response of the closed loop system. For each value of ω_z , the gain was increased until the requirement for the modulus margin was violated to increase the bandwidth as much as possible.

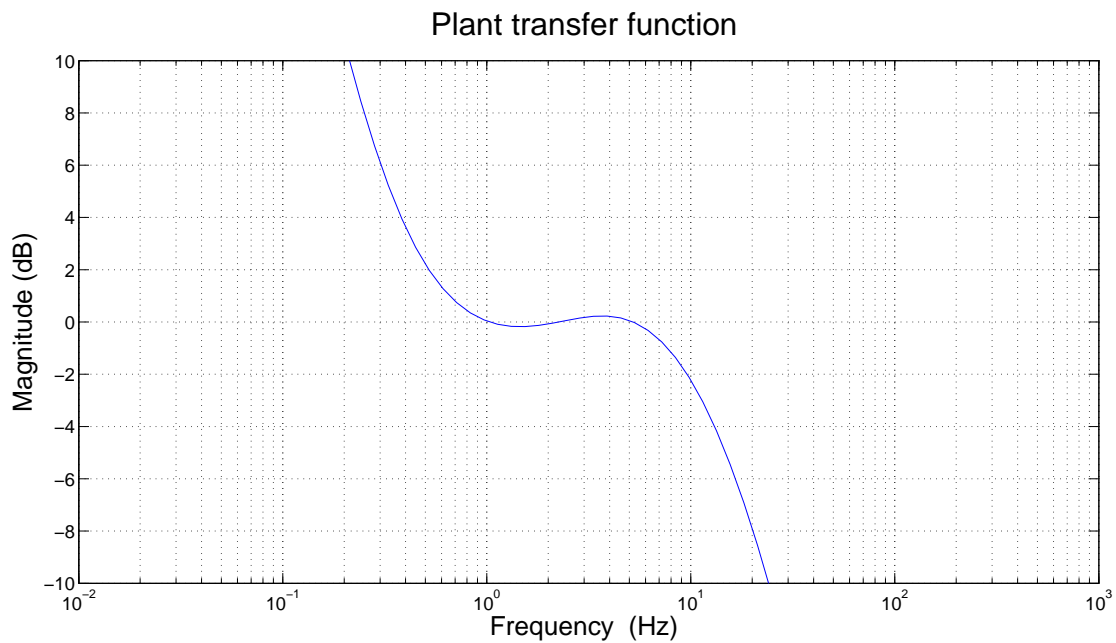


Figure 4.21: Bode magnitude plot of plant transfer function $P_{x_S}(s)$, zoomed in around the crossing of the 0 dB line.

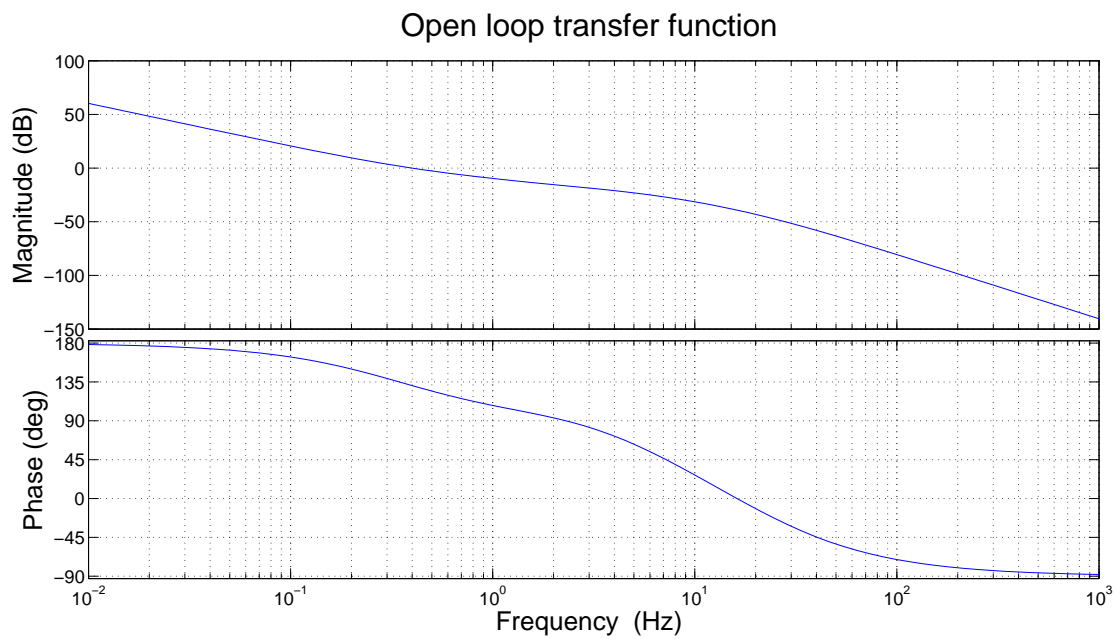


Figure 4.22: Bode plot of the open loop transfer function $L_{x_S}(s)$.

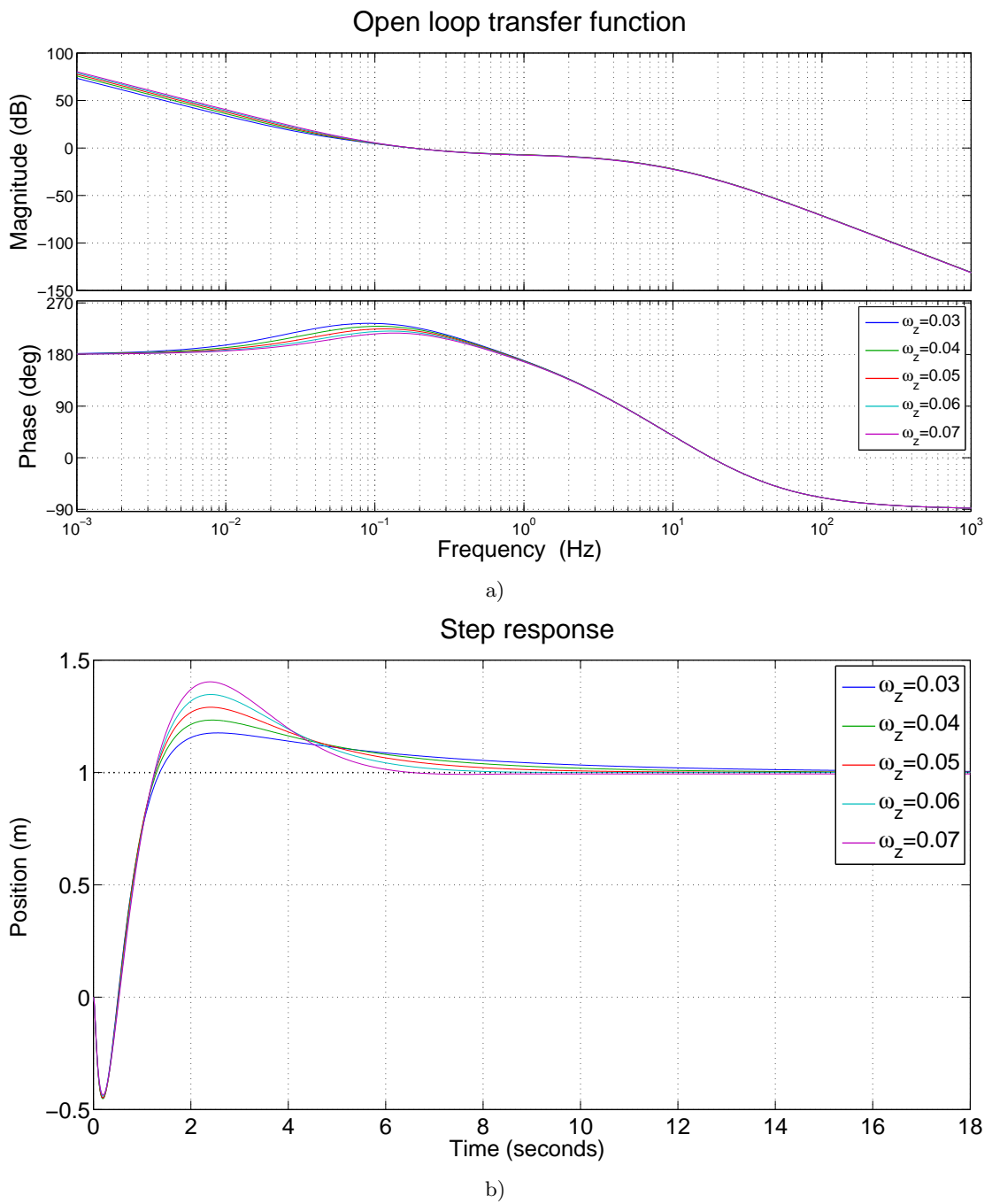


Figure 4.23: Simulations for different values of ω_z . a) Bode plot of the open loop transfer function $L_{x_S}(s)$. b) The step response of the closed loop transfer function $T_{x_S}(s)$.

Inspection of Fig. 4.23b) shows that for all values of ω_z the system is stabilized. Furthermore, choosing the frequency of the zero of the lead compensator is mainly a trade-off between a low settling time⁸ and a low overshoot, which makes it difficult to choose the optimal controller, because there are no critical requirements on the settling time and overshoot. Moreover, the use of feedforward control will be analyzed and feedforward control might influence the performance of the system. Therefore, three different controllers are proposed and after implementing feedforward control, a controller will be chosen, based on the performance of the total system. The following three controllers are proposed:

1. The first controller ($K1_{x_s}$) is designed such that the bandwidth is about 0.12 Hz, the overshoot is about 20% and the settling time is between 10 and 11 seconds.
2. The second controller ($K2_{x_s}$) is designed such that the bandwidth is about 0.15 Hz, the overshoot is about 30% and the settling time is between 7 and 8 seconds.
3. The third controller ($K3_{x_s}$) is designed such that the bandwidth is about 0.18 Hz, the overshoot is about 40% and the settling time is between 5 and 6 seconds.

This results in the following controllers:

$$\begin{aligned}
 K1_{x_s} &= 0.032 \cdot \frac{1}{\left(\frac{1}{2\pi \cdot 0.35}\right)s + 1} \cdot \frac{\left(\frac{1}{2\pi \cdot 0.03}\right)s + 1}{\left(\frac{1}{2\pi \cdot 2}\right)s + 1} \\
 K2_{x_s} &= 0.062 \cdot \frac{1}{\left(\frac{1}{2\pi \cdot 0.35}\right)s + 1} \cdot \frac{\left(\frac{1}{2\pi \cdot 0.05}\right)s + 1}{\left(\frac{1}{2\pi \cdot 2}\right)s + 1} \\
 K3_{x_s} &= 0.1 \cdot \frac{1}{\left(\frac{1}{2\pi \cdot 0.35}\right)s + 1} \cdot \frac{\left(\frac{1}{2\pi \cdot 0.07}\right)s + 1}{\left(\frac{1}{2\pi \cdot 2}\right)s + 1}
 \end{aligned} \tag{4.32}$$

The step response of these three controllers is shown in Fig. 4.24. In the next calculations and simulations on feedforward control, only the results with feedback controller $K1_{x_s}$ are shown.

Analogous to LQR control theory, a reference generator is implemented, which generates 4th order reference signals for the desired coordinates of the ball to reduce the undershoot of the step response, which can be seen in Fig. 4.24. The reference signal is generated with a maximum velocity of 1 m/s and a maximum acceleration of 4 m/s², which are in accordance with the motor capabilities of the BBR. For controller $K1_{x_s}$, the response of the system to the reference signal is compared with the response of the system to a step input and the results are shown in Fig. 4.25.

Fig. 4.25 shows that the implementation of a reference generator drastically reduces the undershoot of the response, while the settling time is hardly affected. To reduce the error e_{x_s} even more, feedforward control is implemented, by adding the reference acceleration signal to the reference signal of the body angle. The reference acceleration signal is chosen as feedforward control, because it matches the error profile of e_{x_s} . To make the feedforward transfer function proper, a second order low-pass filter is added of the form

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4.33}$$

where ω_n is called the natural frequency (in rad/s) and ζ is called the damping ratio. Here ω_n is chosen equal to the closed loop bandwidth and ζ is chosen $\frac{1}{2}\sqrt{2}$, which implies that the poles

⁸The time after which the output remains within a range of 2.5% of the reference value.

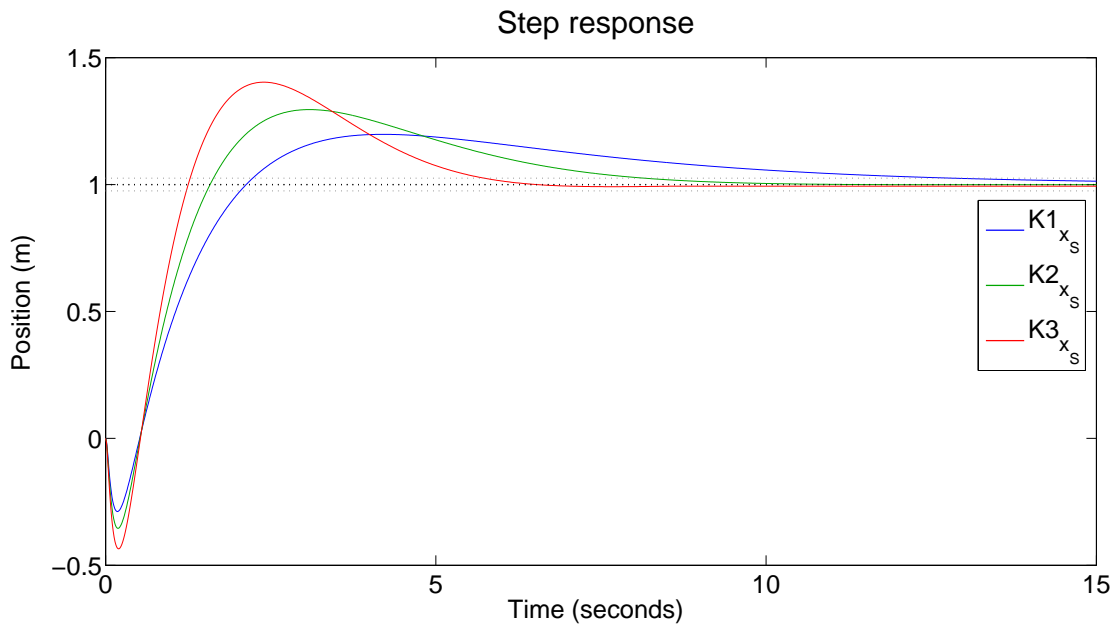


Figure 4.24: The step response of the closed loop transfer function $T_{x_s}(s)$ for three different feedback controllers.

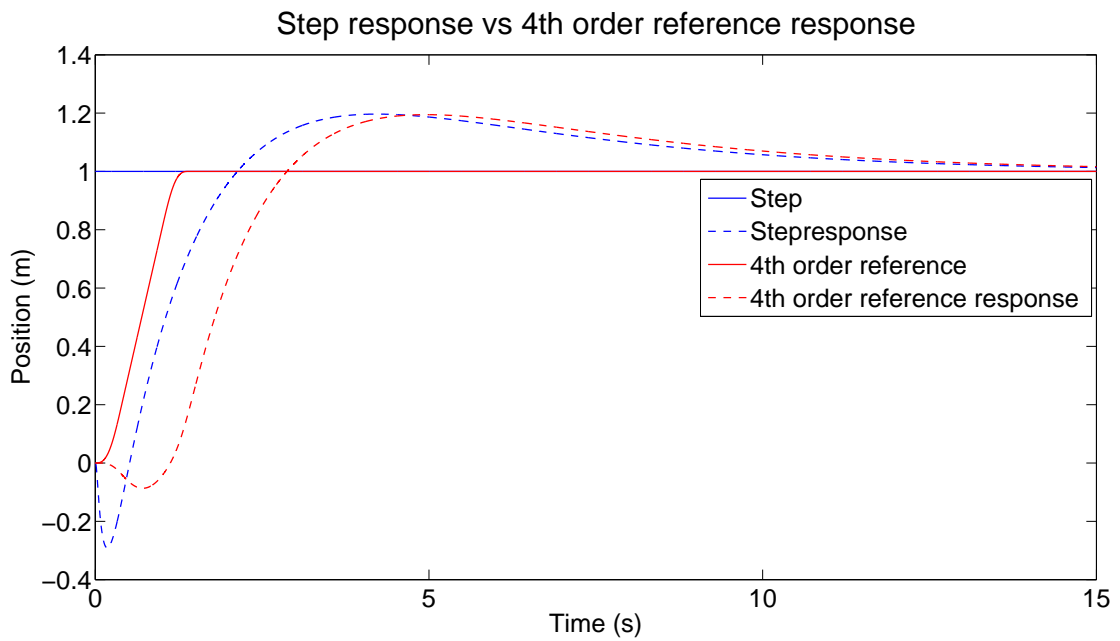


Figure 4.25: The step response versus the response to a 4th order reference signal of the closed loop transfer function with controller $K1_{x_s}$.

are critically damped.

Fig. 4.25 also shows that there is a large delay between the reference signal and the response, because the response of the system is limited by the non-minimum phase behaviour of the plant. To have a more realistic reference position signal, the same low-pass filter, as used for feedforward control, is added to the reference position signal. The resulting control system is shown in Fig. 4.26.

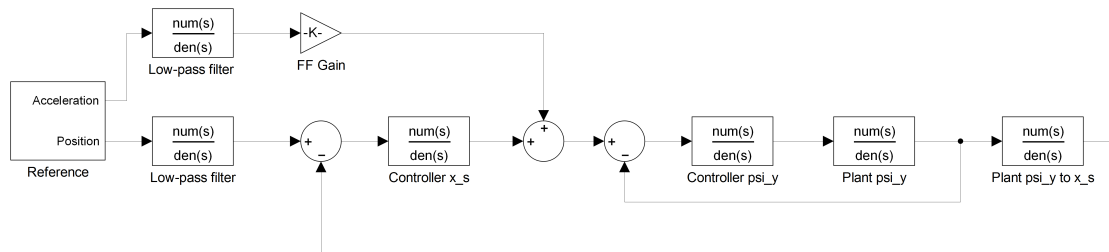


Figure 4.26: Sketch of the closed loop system including a 4th order reference signal and feedforward control.

In order to find the optimal value of the feedforward gain, the system is simulated for different values of the feedforward gain and the response is shown in Fig. 4.27.

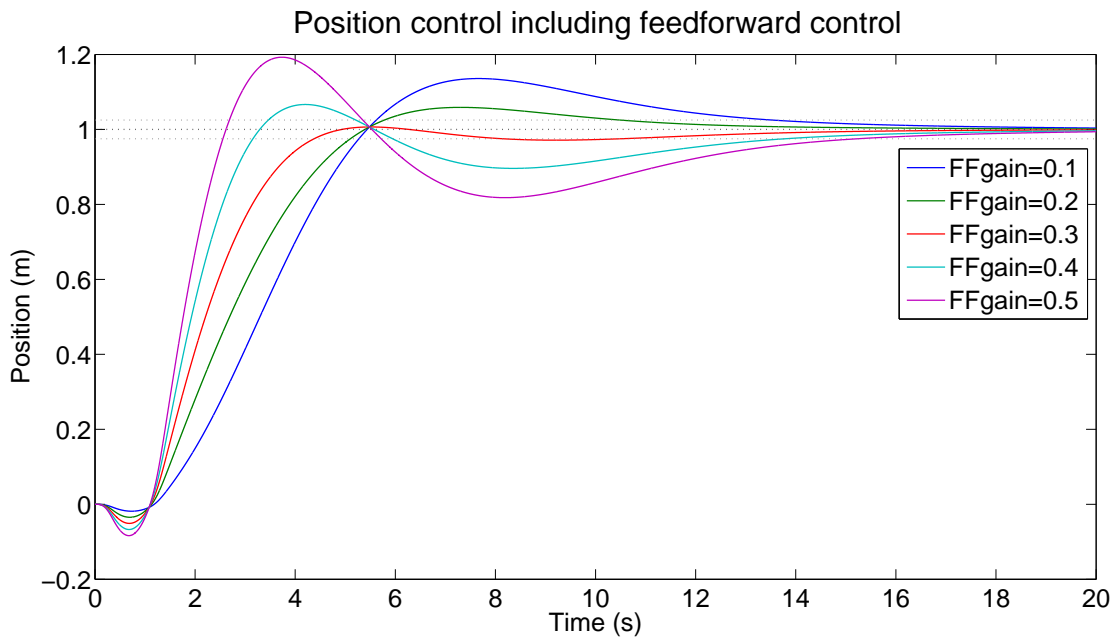


Figure 4.27: The x -position of the ball moving from x -position 0 to 1, including a 4th order reference signal and feedforward control, displayed for feedforward gains from 0.1 until 0.5.

Fig. 4.27 shows that the optimal feedforward gain is somewhere between 0.2 and 0.3, because then the overshoot and the settling time are both minimal. However, with a gain between 0.2 and 0.3 the optimal gain is often a trade-off between a low settling time and a low undershoot,

as shown in Fig. 4.28. Based on a low settling time in combination with a low error, the optimal gain is chosen to be 0.26.

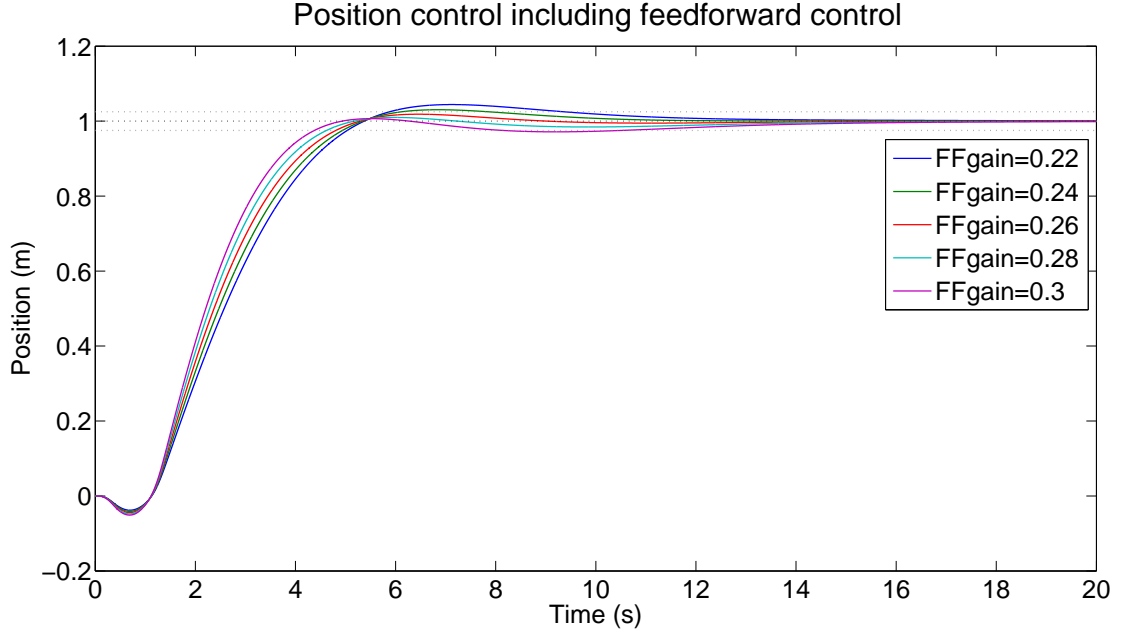


Figure 4.28: The x -position of the ball moving from x -position 0 to 1, including a 4th order reference signal and feedforward control, displayed for feedforward gains from 0.2 until 0.3.

Similarly, the optimal feedforward gains for controllers $K2_{x_S}$ and $K3_{x_S}$ were found to be 0.25 and 0.24 respectively, where the natural frequency ω_n of the low-pass filter was adjusted to the closed loop bandwidth corresponding to the controllers.

Now feedforward control is implemented for all three feedback controllers, their performance will be compared. Therefore, their response to a 4th order reference signal from position 0 to 1 is plotted in Fig. 4.29.

Fig. 4.29 shows that $K2_{x_S}$ and $K3_{x_S}$ clearly have a lower settling time than $K1_{x_S}$. Of $K2_{x_S}$ and $K3_{x_S}$, $K2_{x_S}$ has both the lowest undershoot and the lowest overshoot, so $K2_{x_S}$ is chosen to have the best performance. So the final feedback controller K_{x_S} is

$$K_{x_S} = K2_{x_S} = \frac{5.454s + 1.713}{s^2 + 14.77s + 27.63} \quad (4.34)$$

and the feedforward controller K_{FF} is

$$K_{FF} = 0.25s^2 \cdot \underbrace{\frac{0.8883}{s^2 + 1.333s + 0.8883}}_{\text{low-pass filter}} \quad (4.35)$$

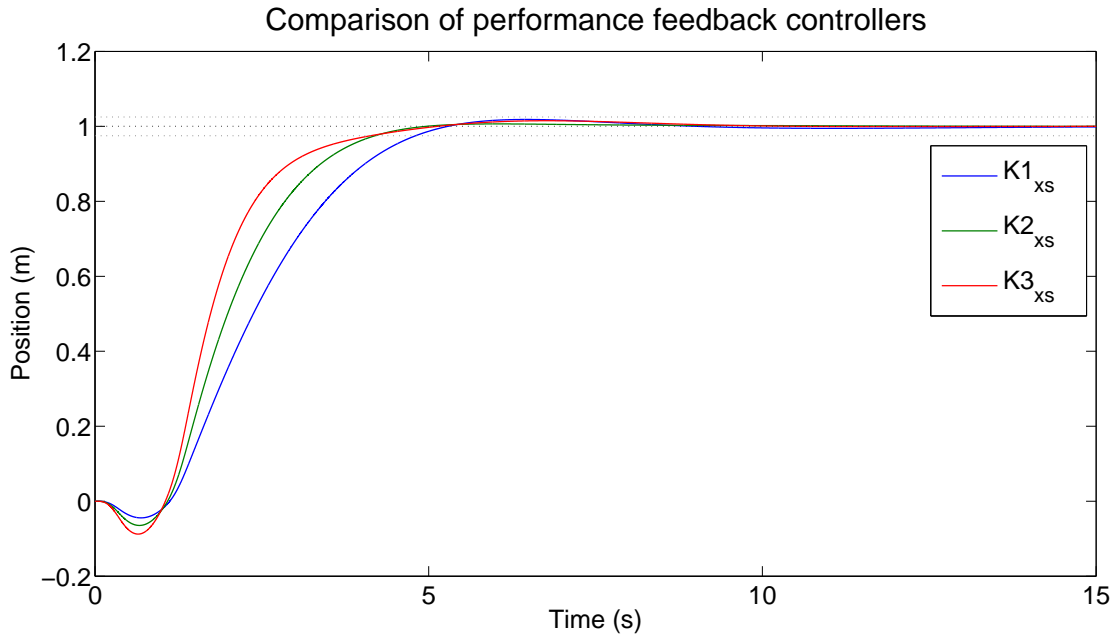


Figure 4.29: Comparison of the response of the system in moving from x -position 0 to 1, with feedback controllers $K1_{x_S}$, $K2_{x_S}$ and $K3_{x_S}$.

4.4.5 Verification of controller requirements

In this section it will be verified whether the position controller satisfies controller requirement 1) and whether the total controlled system verifies controller requirement 4) and 5).

Firstly, to check whether the outer control loop is also stable, the closed loop poles are calculated. They are given by (in rad/s)

$$\begin{aligned}
 s &= -0.61324 \pm 0.31637i \\
 s &= -7.4705 \\
 s &= -15.512 \\
 s &= -72.624 \\
 s &= -141.93
 \end{aligned}
 \tag{4.36}$$

which proves that the outer control loop is also stable. Furthermore, it has an infinite gain margin and a phase margin of 44° , so the position controller also satisfies controller requirement 1).

Secondly, it will be verified whether the total controlled system satisfies controller requirement 4). The system is simulated in Simulink, but setting the initial value of the pitch angle to 10° doesn't approximate the inverse step response good enough. Therefore the inverse step response is approximated with an inverse step function as reference signal for the pitch angle. The results of the described simulation are shown in Fig. 4.30.

Fig. 4.30 shows that the settling time is just above 2 seconds, so the designed controllers satisfy controller requirement 4).

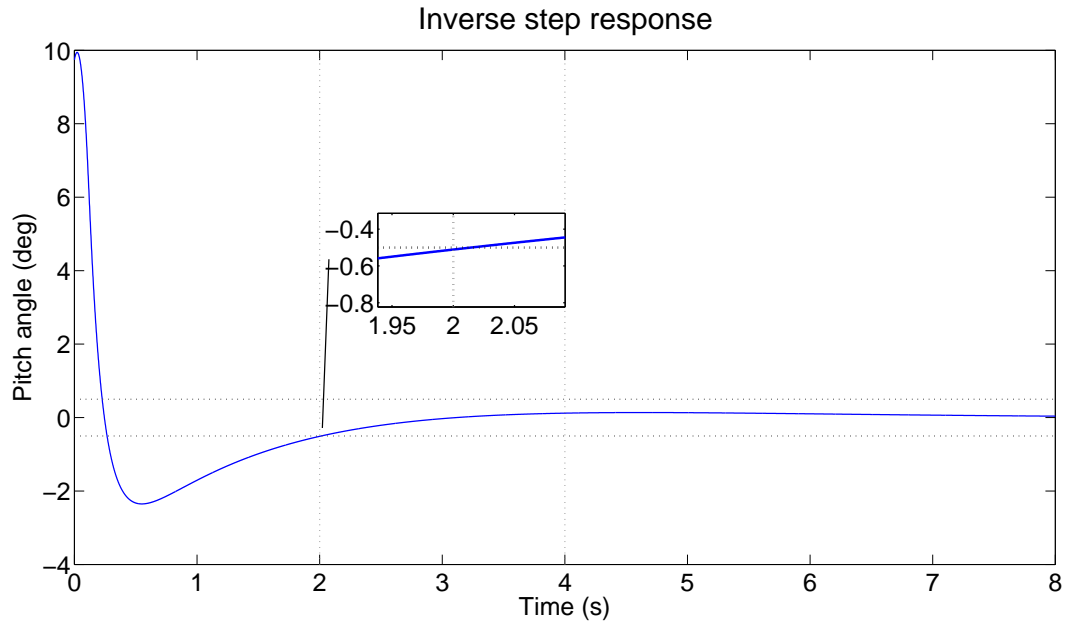


Figure 4.30: The inverse step response of the pitch angle.

Thirdly, it will be verified whether the total controlled system satisfies controller requirement 5). Analogous to the design of a LQR controller, controller requirement 5) is verified by considering the three scenarios, described in Sec. 4.3.4. Again, to predict which scenario will be most probable for the BBR, two different situations will be simulated. They are also described in Sec. 4.3.4, but for sake of clarity, they will be repeated here. In the first simulation, the closed loop system is simulated with the nonlinear model in Simulink for 5 seconds with a sinusoidal disturbance d for the roll angle, with a frequency of 0.5 Hz and an amplitude of 5° , which is for now assumed to be a natural disturbance signal for the BBR. The references are set to zero for all states, which means that the BBR is balancing and station keeping at the position $(x_S, y_S) = (0, 0)$. The disturbance signal together with the torque required to stabilize the system in this situation, is shown in Fig. 4.31.

The dotted gray lines indicate the nominal torque of 1.74 Nm and the dotted red lines indicate the maximum torque of 3.75 Nm.

In the second simulation, the closed loop system is simulated with the nonlinear model in Simulink for 5 seconds with a sinusoidal disturbance d for the roll angle, with a frequency of 0.5 Hz and an amplitude of 15° , which is for now assumed to represent an exceptional situation for the BBR. The references are set to zero for all states, which means that the BBR is balancing and station keeping at the position $(x_S, y_S) = (0, 0)$. The disturbance signal together with the torque required to stabilize the system in this situation, is shown in Fig. 4.32. The dotted gray lines again indicate the nominal torque of 1.74 Nm.

Fig. 4.31 shows that in case of a natural disturbance signal, the required torques stay under the nominal torque, except for the first 0.03 seconds where it has a small peak of about 2.6 Nm. Furthermore, Fig. 4.32 shows that in case of an exceptional disturbance signal, the required torques also stay under the maximum torque, except for the first 0.07 seconds where two of the three torques saturate. So the results of these simulations give an indication that the second

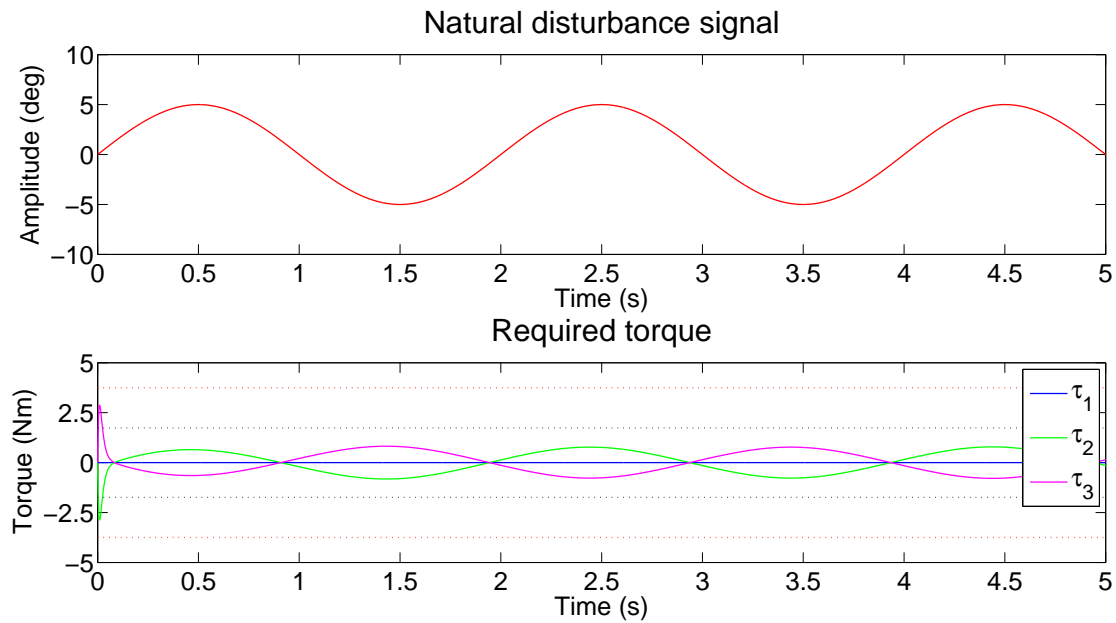


Figure 4.31: The required torque in case of a natural disturbance signal with a frequency of 0.5 Hz and an amplitude of 5° .

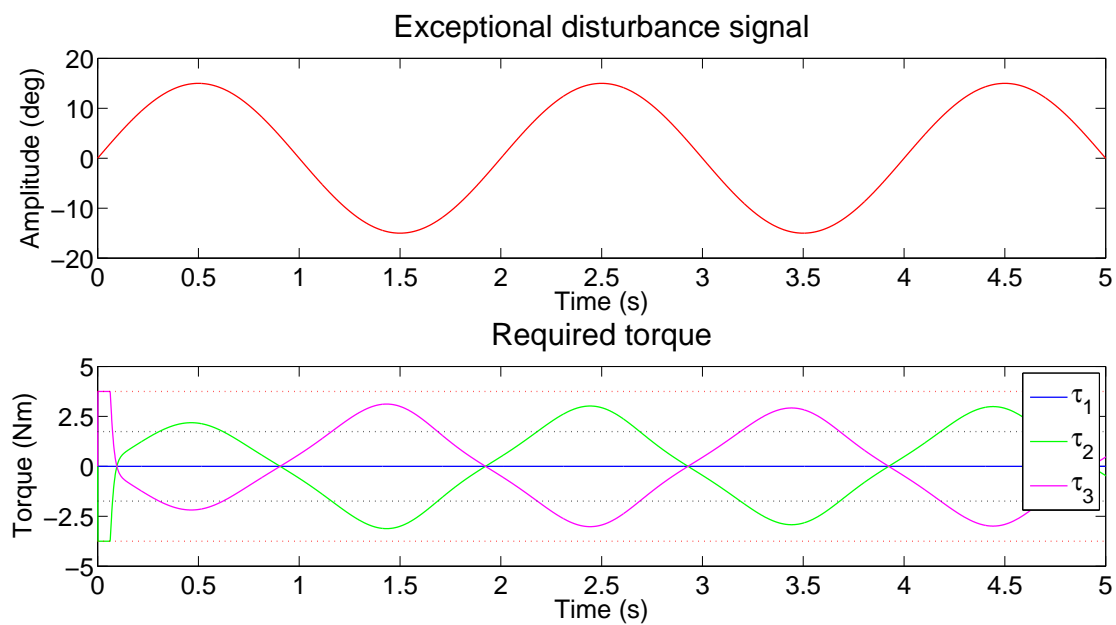


Figure 4.32: The required torque in case of a exceptional disturbance signal with a frequency of 0.5 Hz and an amplitude of 15° .

scenario will most probable for the BBR. But the torque saturation only happens shortly at the beginning and there is no indication that it will make the system unstable, so no additional actions seem to be needed to improve the minimization of torque saturation and the controller satisfies controller requirement 5).

Finally, the closed loop system will be simulated with a initial position of (0,0) and a desired position of (3,1) to illustrate the performance of the position controller. The results are shown in Fig. 4.33.

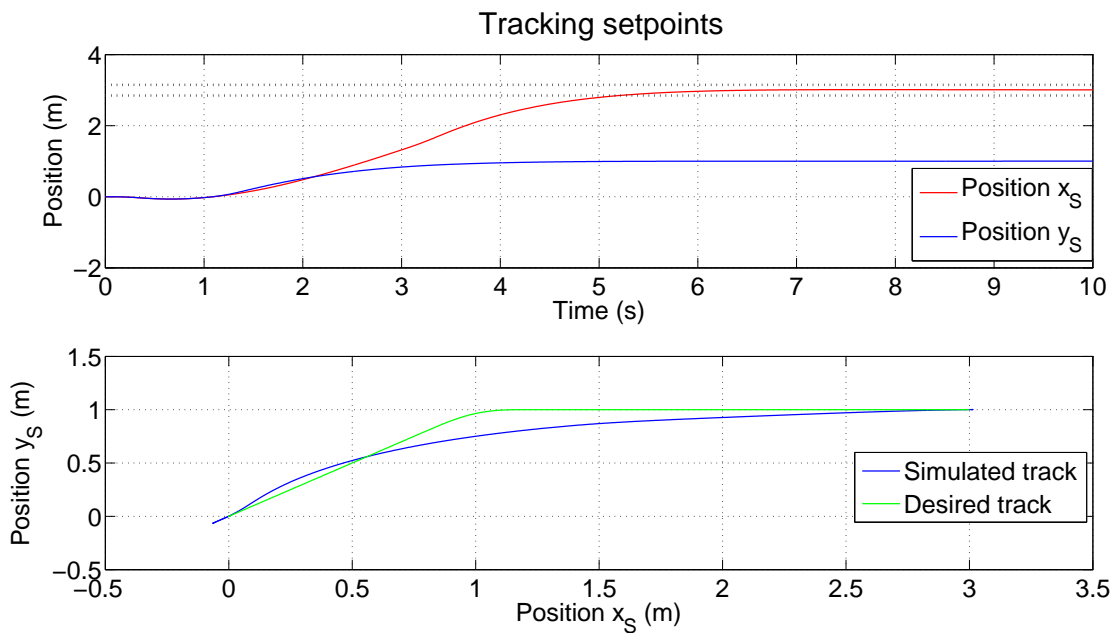


Figure 4.33: Simulation of position of the BBR from position (0,0) to (3,1) with the SISO controller.

Fig. 4.33 shows that the desired position (3,1) is reached in about 6 seconds, which means that the BBR has a linear speed of about 0.5 m/s.

4.5 Derivation of the ball's position

The orientation angles of the body are directly measured by the sensors in contrast with the translation and velocity of the ball. The translation and velocity of the ball can be derived from the data of the motor encoders, that measure the absolute angular position of the omni-wheels. This derivation of the translation of the ball will be done in the following steps:

1. Find the decoupled angular velocities of the omni-wheels in x -, y - and z -direction as functions of the angular velocities of the omni-wheels, using the Jacobian.
2. Integrate the decoupled angular velocities to the traveled angles.
3. Calculate the traveled angles of the ball from the traveled angles of the omni-wheels, which depend on the pitch and roll angles.
4. Calculate the traveled distance in x - and y -direction of the ball from the traveled angles of the ball.
5. Transform the variables to the right coordinate frame.

The velocity of the ball in x - and y -direction is calculated by differentiating the translation of the ball.

A relation between the angular rates of the omni-wheels $\dot{\phi}_1$, $\dot{\phi}_2$ and $\dot{\phi}_3$ and the decoupled angular rates $\dot{\phi}_x$, $\dot{\phi}_y$ and $\dot{\phi}_z$ can be easily derived, based on the relation between the torques, given by the decoupling matrix M in Eq. (4.8). The decoupling matrix M is in fact a Jacobian that relates the torques of the joints to the torques of the end-effectors.

So the angular rates in x , y and z -direction as functions of the angular rates of the omni-wheels are given by the transpose of this decoupling matrix M :

$$\begin{aligned} \begin{bmatrix} \dot{\phi}_x \\ \dot{\phi}_y \\ \dot{\phi}_z \end{bmatrix} &= M^T \cdot \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix} \\ &= \begin{bmatrix} \frac{2\sqrt{2}}{3} & -\frac{\sqrt{2}}{3} & -\frac{\sqrt{2}}{3} \\ 0 & \frac{\sqrt{6}}{3} & -\frac{\sqrt{6}}{3} \\ -\frac{\sqrt{2}}{3} & -\frac{\sqrt{2}}{3} & -\frac{\sqrt{2}}{3} \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix}. \end{aligned} \quad (4.37)$$

Integrating Eq. (4.37) yields

$$\begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} = M^T \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}. \quad (4.38)$$

Now the angles of the ball θ_x , θ_y and θ_z can be derived from ϕ_x , ϕ_y and ϕ_z based on the fact that the traveled distances are equal and the angle orientations are opposite:

$$r_S \cdot \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = -r_W \cdot \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix}. \quad (4.39)$$

From the angles θ_x , θ_y and θ_z the x - and y -position of the ball can be derived:

$$\begin{aligned} \begin{bmatrix} x_S \\ y_S \end{bmatrix} &= \begin{bmatrix} r_S \theta_y \\ -r_S \theta_x \end{bmatrix} \\ &= \begin{bmatrix} 0 & -r_W & 0 \\ r_W & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} \\ &= \begin{bmatrix} 0 & -r_W & 0 \\ r_W & 0 & 0 \end{bmatrix} \cdot M^T \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}. \end{aligned} \quad (4.40)$$

Until now, the angles ϕ_x , ϕ_y and ϕ_z are defined in coordinate frame Ψ_5 , while they in fact should be defined in coordinate frame Ψ_I , because x_S and y_S are defined in coordinate frame Ψ_I . However, the influence of ψ_x and ψ_y is assumed to be negligible, because the BBR should be upright at both the starting point and end point.

But the influence of ψ_z is of course important, because ψ_z determines the direction in which the BBR is moving. To let the BBR keep the right direction, the right-hand side of Eq. (4.40) needs to be pre-multiplied by the upper left part of the rotation matrix R_3^2 (which takes into account the rotation of the BBR around the z -axis) corresponding to the x - and y -position.

So the final x - and y -position of the ball as functions of the angles of the omni-wheels, are given by

$$\begin{bmatrix} x_S \\ y_S \end{bmatrix} = R_3^2(1:2, 1:2) \cdot \begin{bmatrix} 0 & r_W & 0 \\ -r_W & 0 & 0 \end{bmatrix} \cdot M^T \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}. \quad (4.41)$$

where R_3^2 is defined as

$$\begin{bmatrix} \cos(\psi_z) & \sin(\psi_z) & 0 \\ -\sin(\psi_z) & \cos(\psi_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.42)$$

4.6 Conclusion

This chapter has described the design of a controller. Based on the linearized 3D model, a linear controller is designed. The approach taken in the design of this controller is to design first a simple, easily implementable controller that has a short design time to make the BBR ‘fair ready’ as soon as possible and later design a more advanced controller that focuses more on performance. To achieve this, first a static controller is designed with LQR control theory, which controls all states simultaneously. Later, a dynamic cascade controller is designed with SISO loopshaping, which consists of an inner loop for controlling the orientation angles of the body and an outer loop for position control. Both controllers satisfy all controller requirements. However, the maximum bandwidth of the position controller, designed with SISO loopshaping, is limited by the non-minimum phase behaviour of the plant, which means that the desired performance of the position controller is limited.

Several simulations show the differences in these controllers. Firstly, comparing the inverse step response of the pitch angle with both controllers, the difference in performance of the controllers is clearly visible. The settling time with the LQR controller is about 3.5 seconds (see Fig. 4.2), where the settling time with the SISO controller is about 2 seconds (see Fig. 4.30).

Secondly, comparing the response of both controllers to disturbance signals shows that the required torque, calculated by both controllers, differs significantly. The required torque with the LQR controller, which can be found in Fig. 4.6 and Fig. 4.7, is significantly lower than the required torque with the SISO controller, which can be found in Fig. 4.31 and Fig. 4.32.

However, comparing the settling time of the simulations where the BBR moves from position (0,0) to (3,1), the difference in performance of the controllers is hardly visible. The settling time with the LQR controller is about 5.20 seconds (see Fig. 4.8), where the settling time with the SISO controller is about 5.15 seconds (see Fig. 4.33). This is due to the non-minimum phase behaviour of the system, which limits the performance of the position controller, designed with SISO loopshaping.

5. CONTROLLER IMPLEMENTATION

In the previous chapter, two controllers were designed, the LQR controller and the SISO controller. This chapter describes how the LQR controller is implemented. The SISO controller will only be implemented if the implementation of the LQR controller is successful.

During the tests, the robot will be attached to a moving test frame, see Fig. 5.1 and all relevant variables will be saved to a data file.

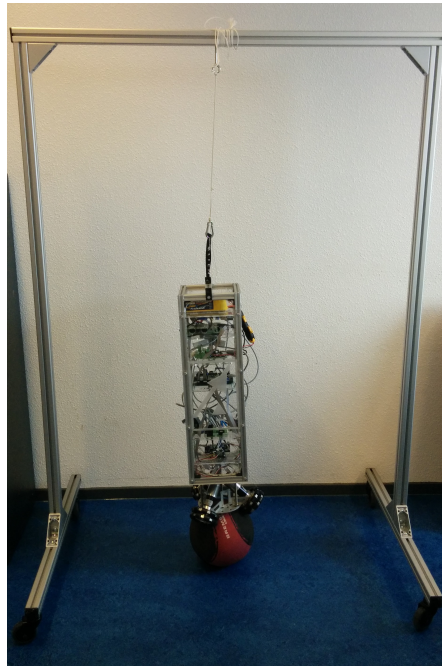


Figure 5.1: The BBR attached to the moving test frame.

To have the ideal situation for the implementation of a controller, especially two things are required:

1. Accurate sensor measurements.
2. Accurate tracking of controller output by the actuators.

The development and implementation of the sensor fusion algorithm and the tuning of the motors is done by ALTEN consultants. A brief overview will be given of the sensor fusion algorithm and the control of the motors.

Sensor fusion algorithm

The IMU contains an accelerometer and a gyroscope. The accelerometer measures linear accelerations of the robot for the x -, y - and z -axis and generally contains high frequency noise. The gyroscope measures the angular rates around the x -, y - and z -axis and generally contains low frequency noise.

The pitch, roll and yaw angular rates are directly based on the gyroscope data. The pitch and roll angles are derived using a complementary filter. A complementary filter tries to get the best data from both the accelerometer and the gyroscope and combines it. How a complementary filter works, is illustrated in Fig. 5.2.

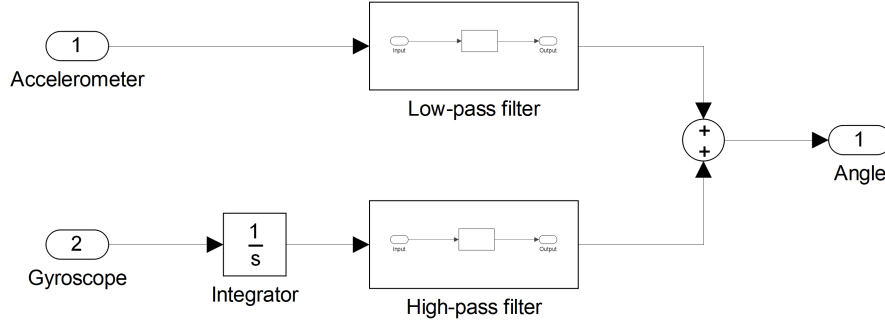


Figure 5.2: Sketch of the working of a complementary filter.

This yields the following equation for each angle:

$$\psi_i = \gamma \cdot (\psi_{i,\text{gyro}} + \dot{\psi}_i \cdot \frac{1}{f_s}) + (1 - \gamma) \cdot \psi_{i,\text{acc}}, \quad i = x, y \quad (5.1)$$

where $\psi_{i,\text{gyro}}$ is the current value of the integrated angular rate around axis i measured by the gyroscope, $\psi_{i,\text{acc}}$ is the angle around the i -axis calculated from the accelerometer data, f_s denotes the sample rate (200 Hz) and γ is a coefficient for the high- and low-pass filters, which is chosen to be 0.99.

Finally, the yaw angle is derived by integrating the yaw angular rate.

Current control of the motors

The three motors are brushless DC motors, controlled by three motor drivers. The three input torques for the motors, calculated by the controller, are converted to currents and the control of the motors consists of a current controller that compares the actual motor current with the applied set value. In case of deviation, the motor current is dynamically readjusted.

5.1 LQR controller for balancing

The design of the LQR controller is based on the continuous-time system, so to implement the LQR controller, it must be recalculated for the discrete-time system. Furthermore, firstly the controller will only be implemented for the pitch and the roll angle and angular rates and if they are successfully implemented and make the BBR balance, the controllers for the other states will be implemented step by step.

However, the gains for the pitch and roll angle and angular rates highly depend on the control of the position of the ball, so they will be different in case of only controlling the pitch and roll angle and angular rates. Therefore, to implement only the controller for the pitch and the roll angle and angular rates, the matrix N , defined in Sec. 4.3.1, will be adjusted such that the pitch and roll angles and angular rates are the only controlled plant outputs.

Based on the discrete-time system and the adjusted matrix N , the gain matrix K is recalculated,

which results in:

$$K = \begin{bmatrix} 0 & 0 & -34.3698 & 0 & 0 & 0 & 0 & -11.2363 & 0 & 0 \\ 0 & 0 & 17.1849 & 29.7651 & 0 & 0 & 0 & 5.6182 & 9.7309 & 0 \\ 0 & 0 & 17.1849 & -29.7651 & 0 & 0 & 0 & 5.6182 & -9.7309 & 0 \end{bmatrix}. \quad (5.2)$$

5.1.1 Encountered problems

Trembling of omni-wheels

Implementing this gain matrix results in heavy trembling of the omni-wheels, such that the omni-wheels lose grip with the ball within a second. The trembling reduces, when the gains for the pitch and roll angular rates are decreased, so the trembling seems to be related to the angular rates. To find out whether the pitch and roll angular rates are the main cause of the trembling, a new test is done where the gains for the pitch and roll angular rates are chosen to be about 25% of its nominal value, to reduce the trembling in order to be able to analyze the cause of the trembling. In this test, the pitch angle is kept around 0° and the roll angle is varied.

The angular rates $\dot{\phi}_i$ (with $i = 1,2,3$) of the omni-wheels are shown in Fig. 5.3. For sake of clarity, the angular rates are expressed in rounds per second, instead of degrees per second.

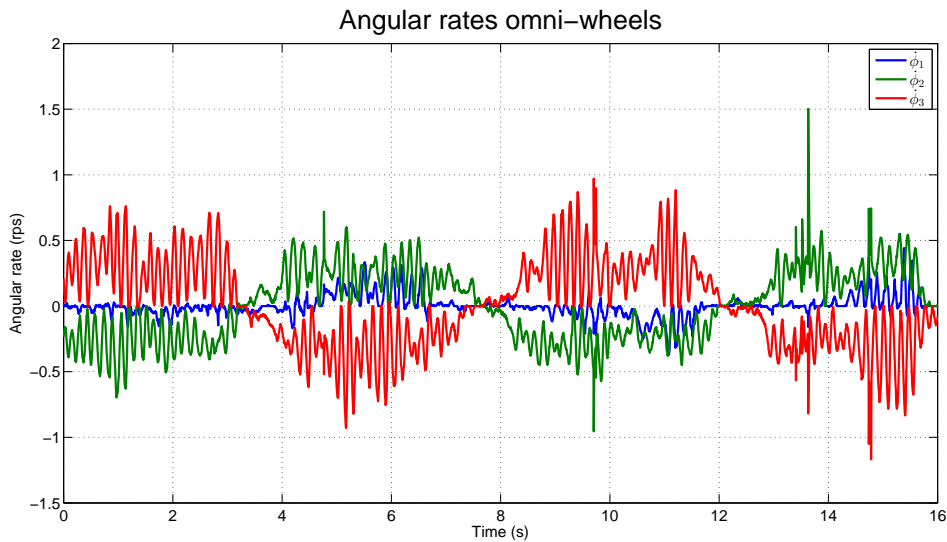


Figure 5.3: The angular rates of the omni-wheels.

Fig. 5.3 clearly shows the trembling of the omni-wheels, because the angular rates of the omni-wheels constantly switch between a zero rate and a nonzero rate. To discover what the cause is of the trembling of the omni-wheels, the torques calculated by the controller are plotted in Fig. 5.4.

Fig. 5.4 shows that the torques are also very noisy, so the trembling of the omni-wheels seems to be mainly caused by noisy sensor data. Therefore, the pitch and roll angle together with their rates are plotted in Fig. 5.5.

Fig. 5.5 shows that especially the angular rates are very noisy, which confirms the presumption that the trembling is related to the angular rates. However, it should be noted that this sensor

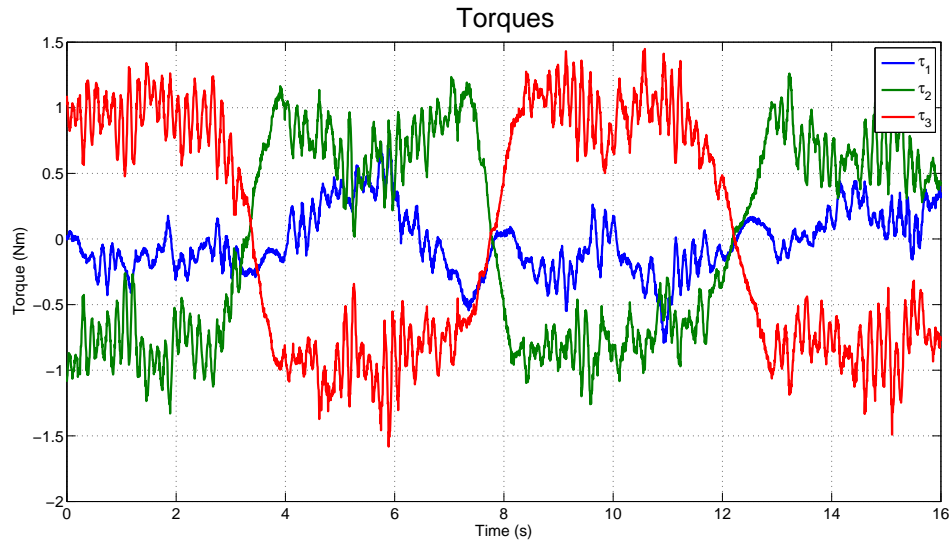


Figure 5.4: The torques generated by the LQR controller.

noise is a mix of both the noise that causes the trembling and the noise that is a result of the trembling.

The pitch and roll angular rates are directly based on the gyroscope data, so the noise is coming from the gyroscope data, but it is not easy to find out what the main cause is of the noisy gyroscope data. The noise can be caused by:

- The gyroscope itself.
- Vibrations of the system, which are measured by the gyroscope.

Gyroscope

To find out if noise caused by the gyroscope itself could play a significant role in the noisy sensor data, the BBR without the ball is attached to the moving test frame, such that the robot is hanging and does not touch the floor. Firstly, a test is done in which the motors are switched off and the robot is given a little push. Secondly, the test is repeated, but now the motors are switched on to see whether the vibrations of the motors influence the gyroscope data. The magnitude spectra of the resulting pitch and roll angular rates are shown in Fig. 5.6.

Fig. 5.6 shows that the magnitude spectra are very similar for both tests, so switching on the motors doesn't result in significant vibrations measured by the gyroscope. Furthermore, the magnitude spectra all have mainly two peaks, one at about 2.5 Hz and the other at about 10 Hz. By comparing the magnitude spectra to the angular rates in time domain, it can be identified whether these peaks represent noise or not. Therefore, the pitch and roll angular rates of the second test (where the motors were switched on) are shown in Fig. 5.7.

Comparing Fig. 5.7 to Fig. 5.6 shows clearly that in Fig. 5.6 the peaks at about 2.5 Hz represent the real signal and the peaks at about 10 Hz represent noise. So noise, caused by the gyroscope itself could seriously play a role in the noisy sensor data.

System vibrations

On the other hand, it is also likely that vibrations of the system play a role in noisy sensor data.

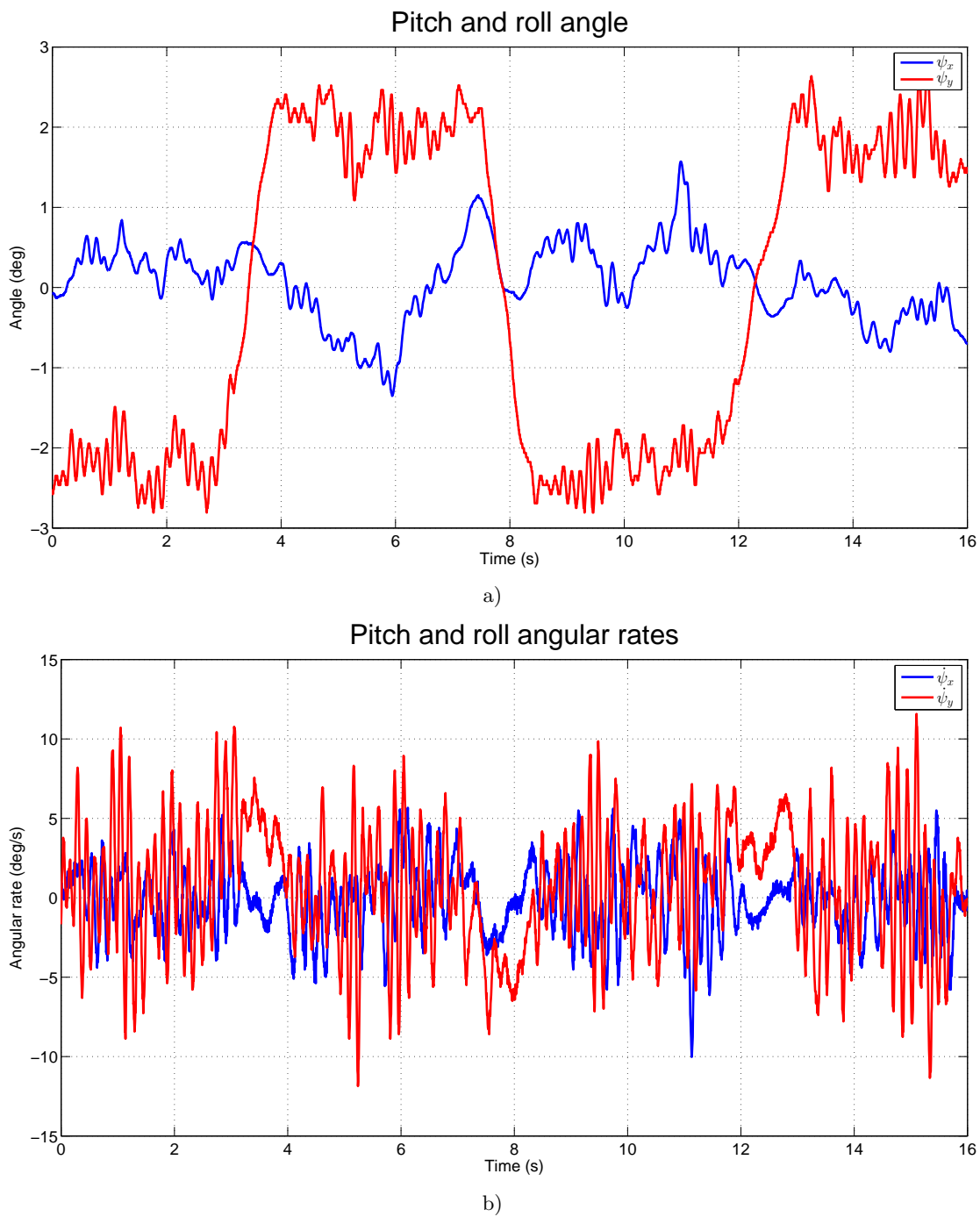


Figure 5.5: a) The pitch and roll angle. b) The pitch and roll angular rates.

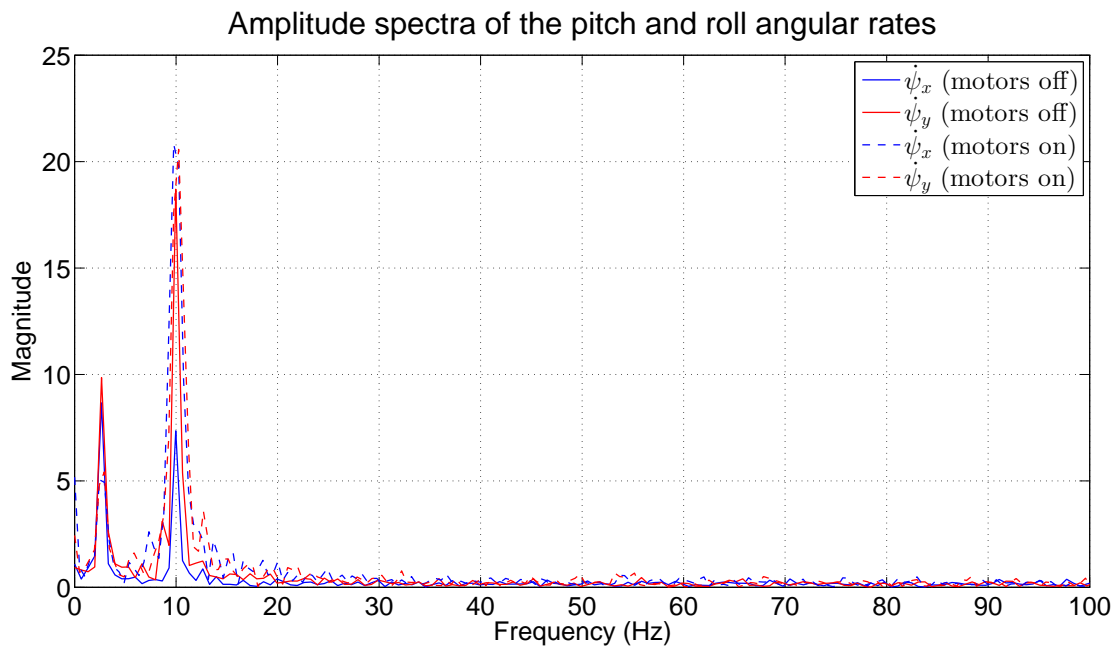


Figure 5.6: The magnitude spectra of the pitch and roll angular rates of the tests, where the BBR is hanging in the moving test frame.

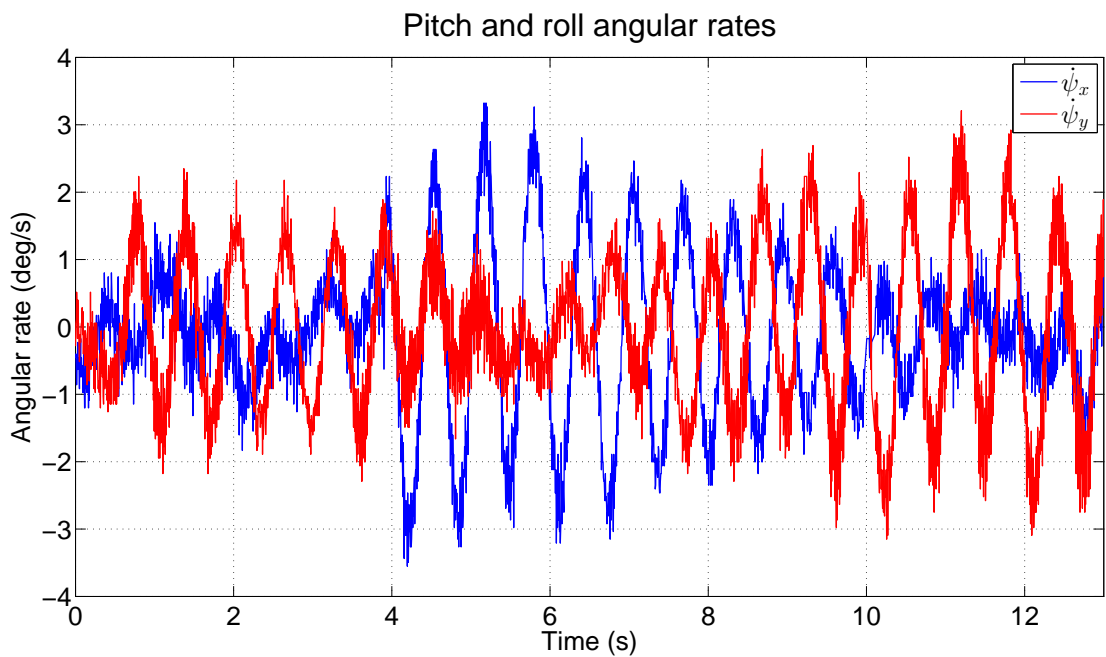


Figure 5.7: The pitch and roll angular rates of the test, where the BBR is hanging in the moving test frame, with the motors switched on.

For example, taking a close look at the omni-wheels, see Fig. 5.8, shows that there are gaps between the rollers of the omni-wheels that could easily be the cause of vibrations in the system, which will be measured by the gyroscope.

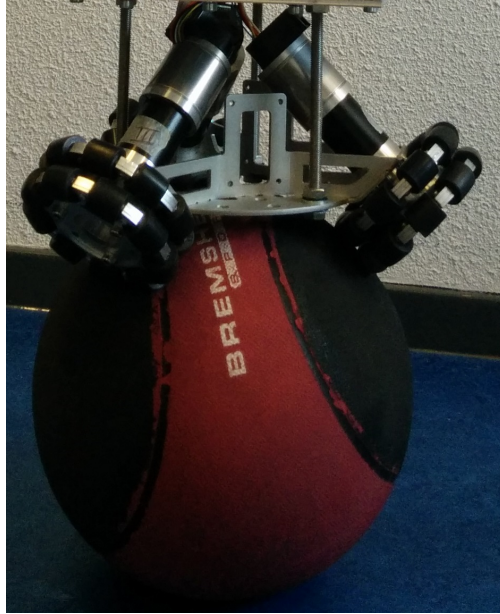


Figure 5.8: The omni-wheels, where the gap between the rollers is clearly visible.

So it is very likely that both noise caused by the gyroscope itself and vibrations of the system play a role in noisy sensor data. However, it is difficult to say with certainty what the *main* cause of the noisy sensor data is.

5.1.2 Applied solutions

In any event, it is important to find out which frequencies play a role in the noisy sensor data, when the BBR is trying to balance. Therefore, a test is done in which the BBR is trying to balance and the magnitude spectra of the pitch and roll angular rates are shown in Fig. 5.9.

Fig. 5.9 shows that, compared to Fig. 5.6, there is significantly more noise, especially between 20 and 50 Hz. Again, it should be noted that this noise between 20 and 50 Hz can be a mix of both the noise that is a result of vibrations of the mechanical system and the noise that is a result of noisy gyroscope data that causes the omni-wheels to tremble.

Different solutions are tried to reduce the noise of the gyroscope data. Due to limited time available, only simple solutions are considered that are easily implementable. As a first solution, foam is placed under the sensorboard to attenuate vibrations of the system. As a second solution, the sample rate of the gyroscope is increased to about four times the sample rate of the IMU, which is 200 Hz, and multiple samples are averaged to one output sample. Filtering the pitch and roll angular rates with a low-pass filter is tried as a third solution.

These solutions both affect the pitch and roll angles and the pitch and roll angular rates. Therefore, it is chosen to analyze the effects of the different solutions by means of the magnitude

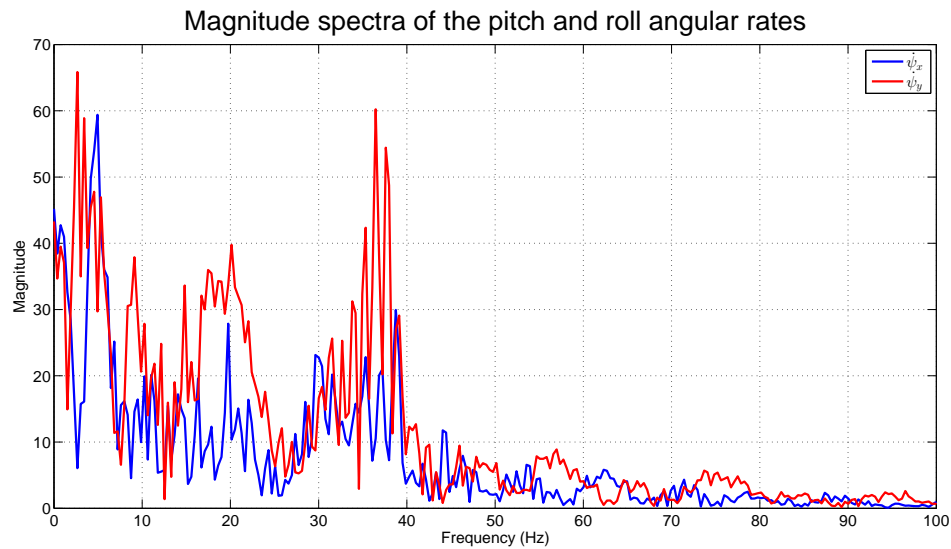


Figure 5.9: The magnitude spectra of the pitch and roll angular rates of the test, where the BBR is trying to balance.

spectra of the torques, calculated by the LQR controller. To be able to compare the results of the applied solutions to the previous test, in which the BBR was trying to balance, the magnitude spectra of the torques of this test are shown in Fig. 5.10.

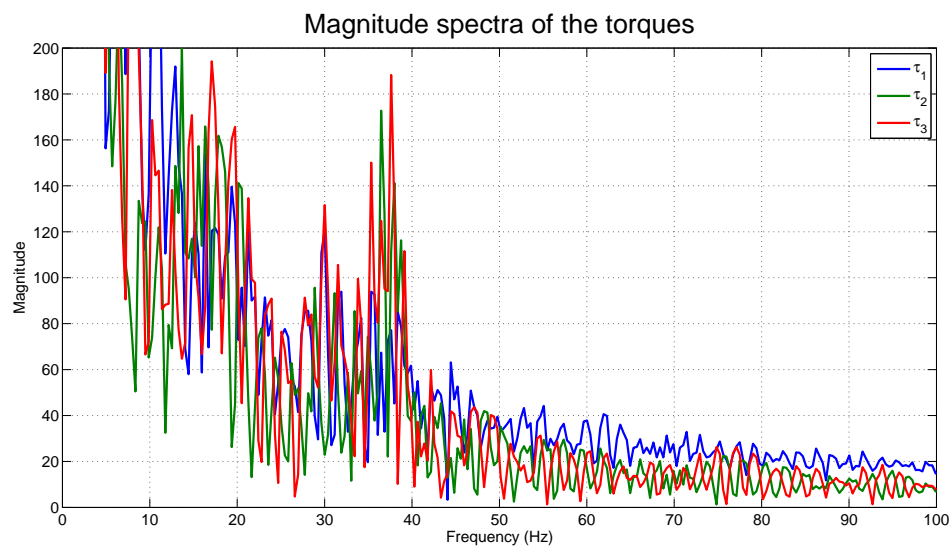


Figure 5.10: The magnitude spectra of the torques of the test, where the BBR is trying to balance.

Foam

Firstly, foam is placed under the sensorboard to attenuate vibrations of the system that might be measured by the gyroscope. To analyze the reduction of noise due to the addition of foam,

a test is done in which the BBR again tries to balance. The resulting magnitude spectra of the torques are shown in Fig. 5.11.

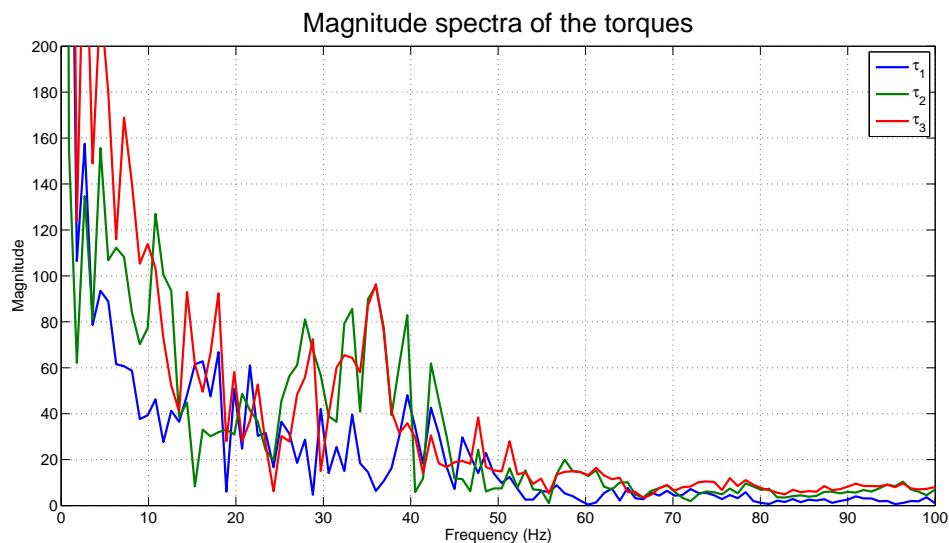


Figure 5.11: The magnitude spectra of the torques of the test, where the BBR is trying to balance after the foam is added.

Comparing the magnitude spectra of Fig. 5.11 to the magnitude spectra, shown in Fig. 5.10, where the BBR is trying to balance, shows a significant reduction of noise. This is also visible in the test itself, where the trembling of the omni-wheels significantly decreased and the BBR is able to balance for several seconds.

Averaging gyroscope data

Secondly, the sample rate of the gyroscope is increased to 760 Hz and multiple samples are averaged to one output sample to reduce the noise, caused by the gyroscope itself. To analyze the effect of averaging the gyroscope data, again a test is done in which the BBR tries to balance and the resulting magnitude spectra of the torques are shown in Fig. 5.12.

Comparing the magnitude spectra of Fig. 5.12 to the magnitude spectra, shown in Fig. 5.11, where the BBR is trying to balance after the foam was added, shows that also the averaging of the gyroscope data significantly reduces the noise. This results in a robot that is now able to balance for a longer time, although trembling of the omni-wheels still limits the performance of the robot.

Low-pass filter

Thirdly, a low-pass filter is designed to filter the torques, calculated by the LQR controller. Inspection of Fig. 5.12 shows that especially between 30 and 40 Hz the noise is significant, so a low-pass filter will be designed that especially attenuates the torques between 30 and 40 Hz.

This low-pass filter is designed with the Filter Design Toolbox of MATLAB, which uses either an Infinite Impulse Response (IIR) filter or a Finite Impulse Response (FIR) filter as design method. It is chosen to design a FIR filter, because it can better address specific frequencies. The resulting FIR filter is a 4th order filter with a passband up to 17 Hz and a stopband starting at 28 Hz. A Bode magnitude plot of the filter is shown in Fig. 5.13.

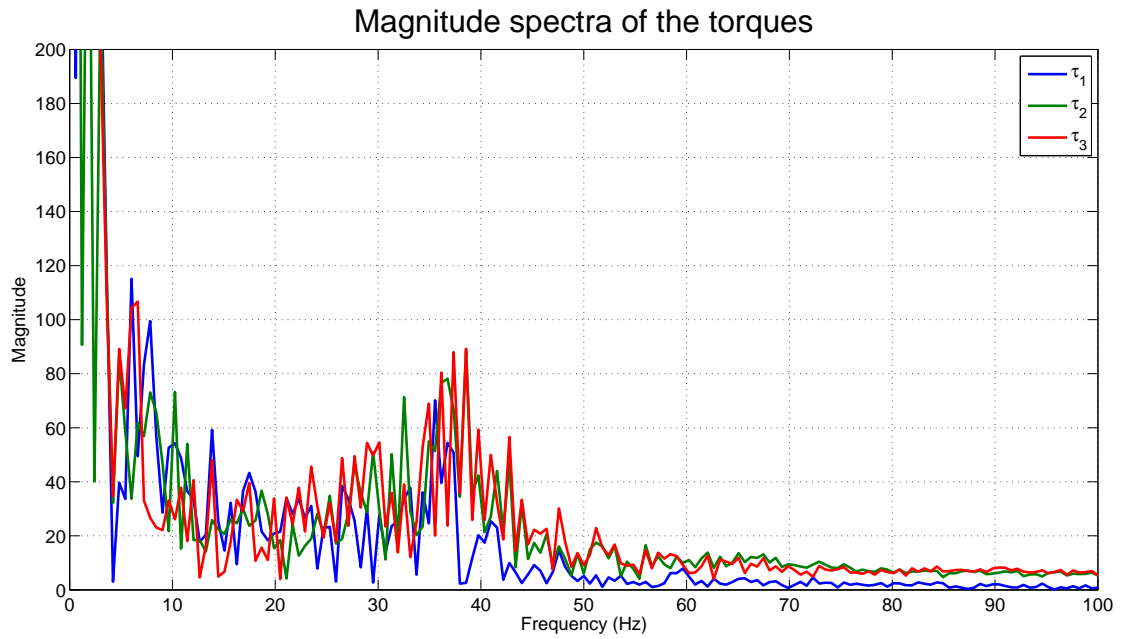


Figure 5.12: The magnitude spectra of the torques of the test, where the BBR is trying to balance after averaging the gyroscope data.

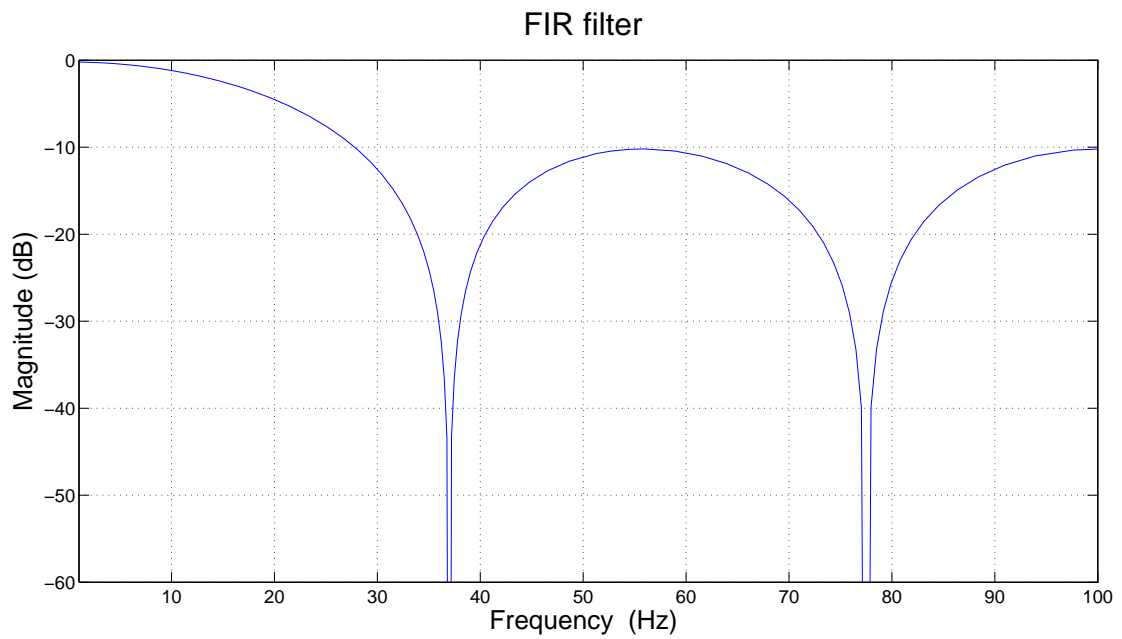


Figure 5.13: Bode magnitude plot of the 4th order FIR filter, used to filter the torques.

To see the effect of the filter, the unfiltered and the filtered torque are plotted together for motor one in Fig. 5.14.

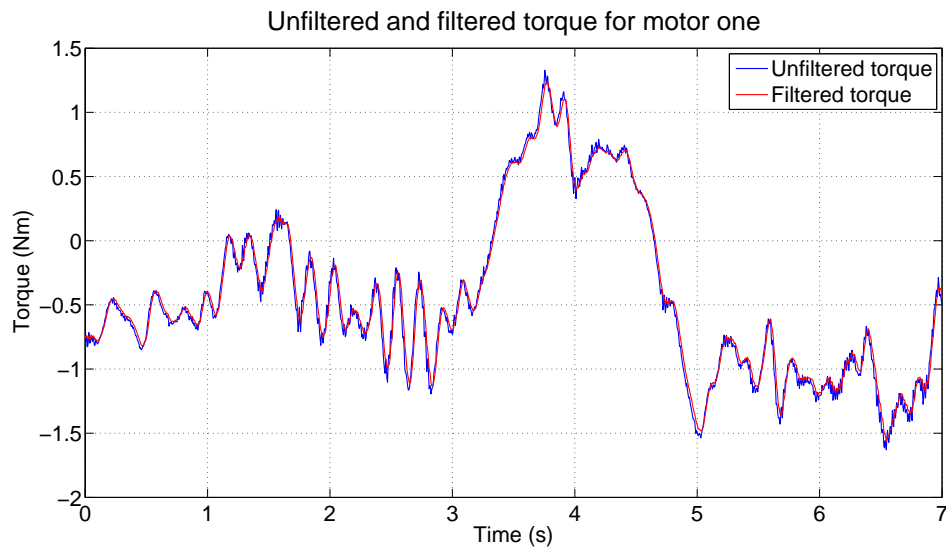


Figure 5.14: The unfiltered and filtered torque of motor one.

Fig. 5.14 shows that the noise is significantly reduced. After the implementation of this FIR filter, the trembling of the omni-wheels has almost disappeared and performance of the robot is further improved with regard to balancing, although the robot is now driving away with a relatively high velocity. To proof the effect of the FIR filter, the magnitude spectra of the torques are shown in Fig. 5.15 of a test after the implementation of the FIR filter.

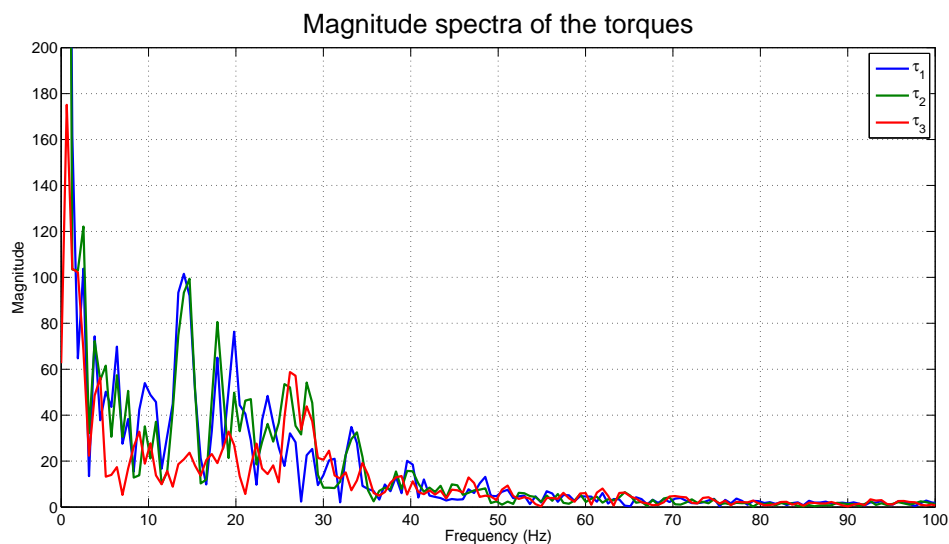


Figure 5.15: The magnitude spectra of the torques of a test, after the FIR filter is implemented.

Comparing the magnitude spectra of Fig. 5.15 to the magnitude spectra, shown in Fig. 5.12,

shows that the peaking around 37 Hz in Fig. 5.12 is significantly reduced due to the FIR filter. However, between 10 and 20 Hz, the noise significantly increased. Most likely, this is due to the fact that the robot is driving away, which results in an increase of system vibrations.

Finally, the gains for the pitch and roll angle are slightly increased to make the system respond a little faster to a nonzero tilt angle. This results in the following gain matrix K :

$$K = \begin{bmatrix} 0 & 0 & -39.4254 & 0 & 0 & 0 & 0 & -3.1443 & 0 & 0 \\ 0 & 0 & 19.7127 & 34.1434 & 0 & 0 & 0 & 1.5671 & 2.7143 & 0 \\ 0 & 0 & 19.7127 & -34.1434 & 0 & 0 & 0 & 1.5671 & -2.7143 & 0 \end{bmatrix}. \quad (5.3)$$

Due to the fact that the controller mainly penalizes the errors in pitch and roll angles and only slightly penalizes the errors in pitch and roll angular rates, the BBR is only able to balance within small variations of the pitch and roll angles.

5.2 LQR controller for yaw control

Now the BBR is able to balance, the controller gains for the yaw angle and angular rate can be implemented. The gains for the yaw angle and angular rate can be found in Eq. (4.13) and are given by respectively -4.1350 and -1.4307 . In Fig. 5.16 the results of a test in which yaw control is not applied are shown together with the results of a test in which yaw control is applied.

Fig. 5.16 shows that the controller gains for the yaw angle and angular rate are successfully implemented. However, the trembling of the omni-wheels slightly increased after implementing the gains for the yaw angle and angular rate, so it has to be considered if it is really necessary to control the yaw angle and angular rate.

5.3 LQR controller for position control

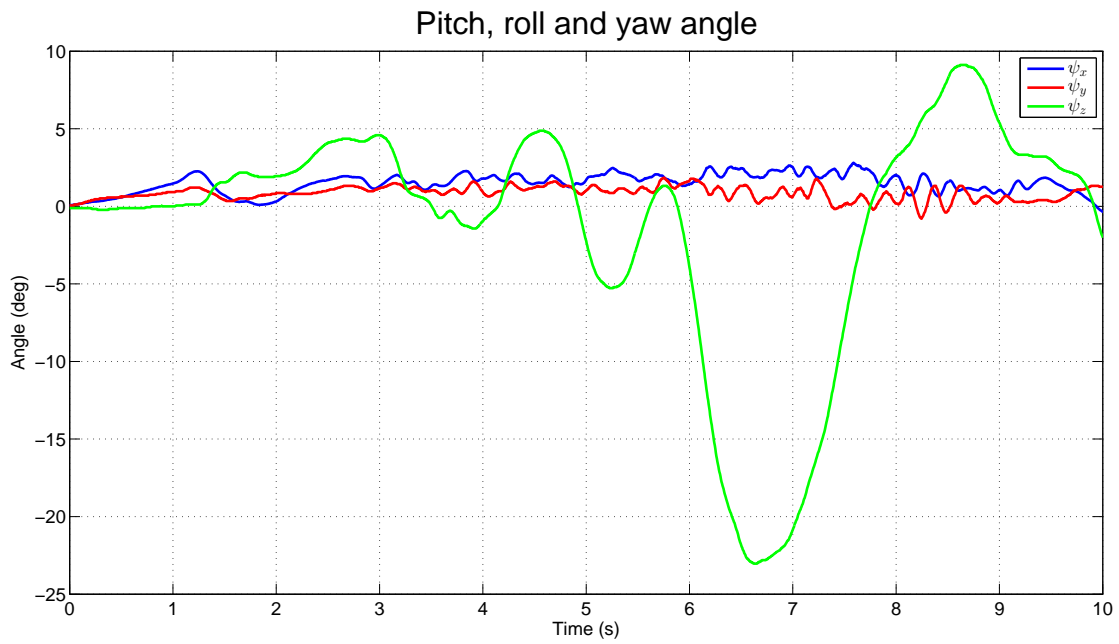
In order to control the position and velocity of the ball, the BBR must be able to cope with changes in pitch and roll angles. However, as mentioned in Sec. 5.1, at this moment the BBR is only able to balance with small variations in pitch and roll angles, due to the fact that the controller mainly penalizes the errors in pitch and roll angles and only slightly penalizes the errors in pitch and roll angular rates. So in order to be able to control the position and velocity of the ball, higher controller gains for the pitch and roll angular rates are required.

5.4 Conclusion

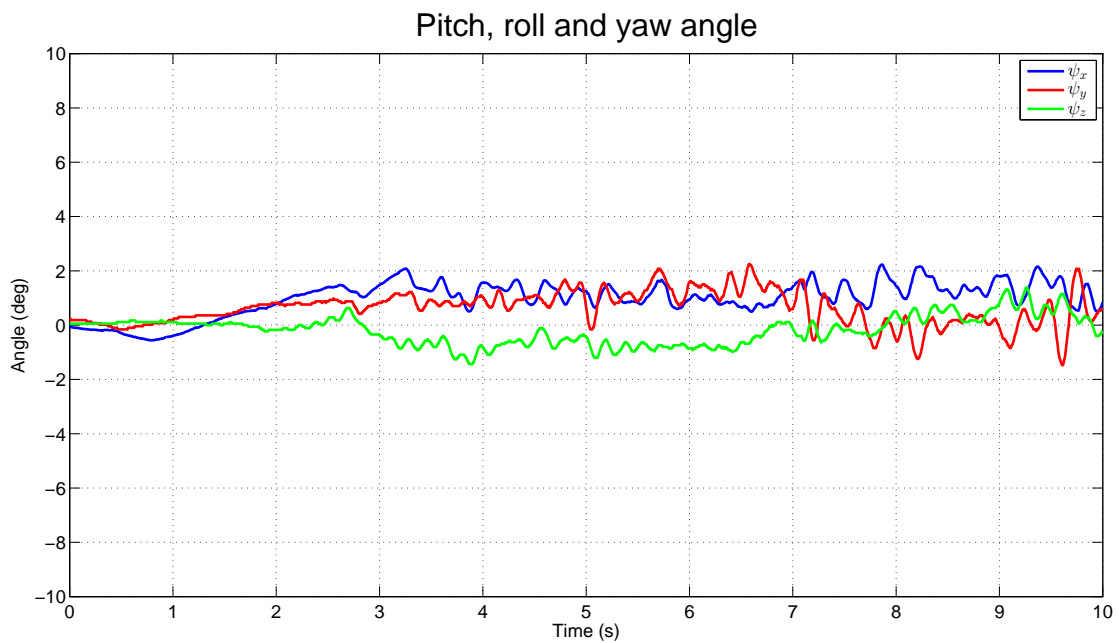
This chapter has described the implementation of the LQR controller. For the controller to be implemented as desired, it is especially important that the controller receives accurate sensor measurements as input and that the controller output is accurately tracked by the motors and omni-wheels.

5.4.1 Balancing control

However, during the implementation of the LQR controller, it turned out that noisy sensor data limits the controller implementation. In particular, the magnitude of the controller gains for the pitch and roll angular rates is seriously limited by noisy gyroscope data.



a)



b)

Figure 5.16: The pitch, roll and yaw angle in case: a) Yaw control is applied. b) Yaw control is not applied.

Different solutions are tried to reduce the noisy gyroscope data. Firstly, foam is placed under the sensorboard to attenuate vibrations of the system that might be measured by the gyroscope. Secondly, the sample rate of the gyroscope is increased to about four times the sample rate of the IMU and multiple samples are averaged to one output sample, in order to reduce the noise caused by the gyroscope itself.

Furthermore, a FIR low-pass filter is implemented to reduce the consequences of the noisy gyroscope data and to smoothen the torque inputs of the motors.

The implementation of these solutions significantly reduces the noise of the sensor data and results in a BBR that is able to balance.

5.4.2 Control of yaw motion

The controller gains for the yaw angle and angular rate are successfully implemented, but this slightly increases the trembling of the omni-wheels. Therefore, it is recommended to leave out the controller gains for the yaw angle and angular rate if control of the yaw angle and angular rate is not strictly necessary for the purpose of the robot.

5.4.3 Position control

To be able to control the position of the BBR, the BBR must be able to cope with large variations in pitch and roll angles in order to move around. However, the controller now mainly penalizes the errors in pitch and roll angles and only slightly penalizes the errors in pitch and roll angular rates. This implies that the BBR is only able to balance within small variations of the pitch and roll angle. Therefore, position control is impossible as long as the magnitude of the gains for the pitch and roll angular rates is limited by noisy sensor data.

5.4.4 SISO controller

Finally, it is recommended to only implement the SISO controller if the LQR controller is successfully implemented for all states. Therefore, the implementation of the SISO controller is left as future work.

6. SYSTEM STABILITY RESEARCH

In this chapter, research will be done under what requirements the system will remain stable with the developed controller, assuming that the developed controller will stabilize the system. For example, in the future new features might be added to the BBR that increase the mass of the robot and therefore influence the stability of the system. It is therefore important to know whether the implemented controller can still stabilize the system if system parameters change. In the first paragraph the approach of this research will be formulated. In the second paragraph a sensitivity analysis of the system will be done with the LQR controller. In the third paragraph a sensitivity analysis of the system will be done with the SISO controller. Finally, in the fourth paragraph, the results will be summarized in a short conclusion.

6.1 Research approach

6.1.1 The aim of the research

This research will investigate under what requirements the system remains stable with the developed controller. To be able to answer this question, especially three things are important:

1. Firstly, to find out the influence of *unmodeled parameters/phenomena* on the stability of the system with the developed controller.
2. Secondly, to find out the influence of *uncertainties in estimated model parameters* on the stability of the system with the developed controller.
3. Thirdly, to predict whether *future changes* on the robot will destabilize the system with the developed controller or not.

First of all, it is important to make clear what is meant with stability. Stability in this case does not only mean that the closed loop poles have a negative real part, but means that the BBR is able to balance with tilt angles of at most 10° . Therefore, also the required torque will be taken into account in this research, because it might be possible that with certain parameter values, the closed loop poles of the system all have a negative real part, but that the motors are not strong enough to let the BBR balance.

Of course, being able to balance also means that things like the maximum motor load must be taken into account, but the research on this chapter will only focus on stability in terms of closed loop poles and the required torque, because they depend on the developed controller. Stability requirements that are controller independent will not be taken into account in this research.

Secondly, it is important to make clear which parameters/phenomena are relevant for this research. For example, based on the aim of this research an analysis on the effect of the slope of the floor is thought to be irrelevant for the stability of the system. The parameters/phenomena can be divided into two groups, namely unmodeled parameters/phenomena and modeled parameters.

6.1.2 Unmodeled parameters/phenomena

The unmodeled parameters can mainly be derived from the assumptions made for developing the 3D model, which can be found in Sec. 3.1. These unmodeled parameters/phenomena are

- Deformation of the ball and the body of the BBR.
- Slip between the omni-wheels and the ball or between the ball and the ground.
- Rolling and kinetic friction.
- The slope of the floor.
- A time delay between the measurements of the sensors and the control of the actuators.
- The effect of 2-row omni-wheels instead of the modeled 1-row omni-wheels.

Based on the goal of this research and the current performance of the system, most of these unmodeled parameters/phenomena are thought to be irrelevant for this research, because their influence on the stability of the system is negligible. But some of them, like deformation of the ball, slip and rolling friction could possibly play a role in future changes, such as replacing the ball or increasing the maximum tilt angle from 10° to 15° .

6.1.3 Modeled parameters

All modeled parameters can be found in Appendix A. Most of these parameters are accurately measured and will not change in future, but some of these parameters are estimated (like the moments of inertia) and other parameters might probably change in future (like the mass of the upper part of the robot and the height of the COM). These uncertain parameters are relevant for this research. An overview of the uncertain model parameters, together with their nominal values, is given in Table 6.1.

Table 6.1: Description of the uncertain model parameters with their nominal value.

Parameter	Description	Nominal value
I_S	Moment of inertia of the ball	$2.82 \cdot 10^{-2} \text{ kgm}^2$
I_M	Moment of inertia of the rotor of the motor	$2.42 \cdot 10^{-6} \text{ kgm}^2$
I_{OW}	Moment of inertia of the omni-wheel	$6.94 \cdot 10^{-4} \text{ kgm}^2$
$I_{B,x}$	Moment of inertia of the body about the x -axis of the coordinate frame centered in its COM	$8.76 \cdot 10^{-1} \text{ kgm}^2$
$I_{B,y}$	Moment of inertia of the body about the y -axis of the coordinate frame centered in its COM	$8.76 \cdot 10^{-1} \text{ kgm}^2$
$I_{B,z}$	Moment of inertia of the body about the z -axis of the coordinate frame centered in its COM	$6.67 \cdot 10^{-2}$
m_S	Mass of the ball	3.2 kg
m_{B1}^a	Mass of the upper part of the body	6.2 kg
l	Distance between COM of the ball and COM of the body of the robot	0.405 m
α	Angle the omni-wheels make with the top of the ball	45°

^a m_{B2} is chosen to be the mass of the drive system, which will remain fixed. Therefore, all the mass that will be added to the body of the BBR will be taken into account in m_{B1} .

From all the uncertain parameters, mentioned in Table 6.1, I_M and I_{OW} have a very low nominal value and the influence of uncertainty in I_M and I_{OW} is therefore assumed to be negligible.

6.1.4 Execution of the research

Due to the fact that the BBR is not stable with regard to the position of the ball at the moment, it is impossible to do meaningful experiments with the real system. Therefore, it is chosen to do the research with the developed dynamical model. The research in this chapter will therefore only focus on the second and third point of the aim of the research, mentioned in Sec. 6.1.1 and an analysis on the influence of unmodeled parameters/phenomena on the stability of the system with the developed controller, is left as future work.

Therefore, no quantitative conclusions can be made, based on this research, but the results of the simulations only give an indication for the real system.

In the next sections, a sensitivity analysis will be done on these uncertain model parameters with the LQR controller and the SISO controller. Firstly, the system will be simulated for uncertainties in each of the moments of inertia I_S , $I_{B,x}$, $I_{B,y}$ and $I_{B,z}$. Secondly, the system will be simulated for a combined uncertainty in the parameters m_{B1} and l , because l mostly depends on m_{B1} . Thirdly, the system will be simulated for an uncertainty in the parameters m_S and α .

In most of these simulations the focus will be threefold:

1. Firstly, it will be analyzed whether the total system is stable or not, based on the closed loop poles of the system.
2. Secondly, the stability of the balancing controller will be analyzed by means of the gain and phase margin, provided that the uncertain parameter influences the balancing controller (which is not the case for $I_{B,z}$).
3. Thirdly, in simulations of possible future changes also the required torque will be calculated for an inverse step response (with a starting angle of 10°) of the roll angle to check whether torque saturation might destabilize the system. These simulations will be executed for a stable worst-case scenario and in these simulations no position control will be applied.

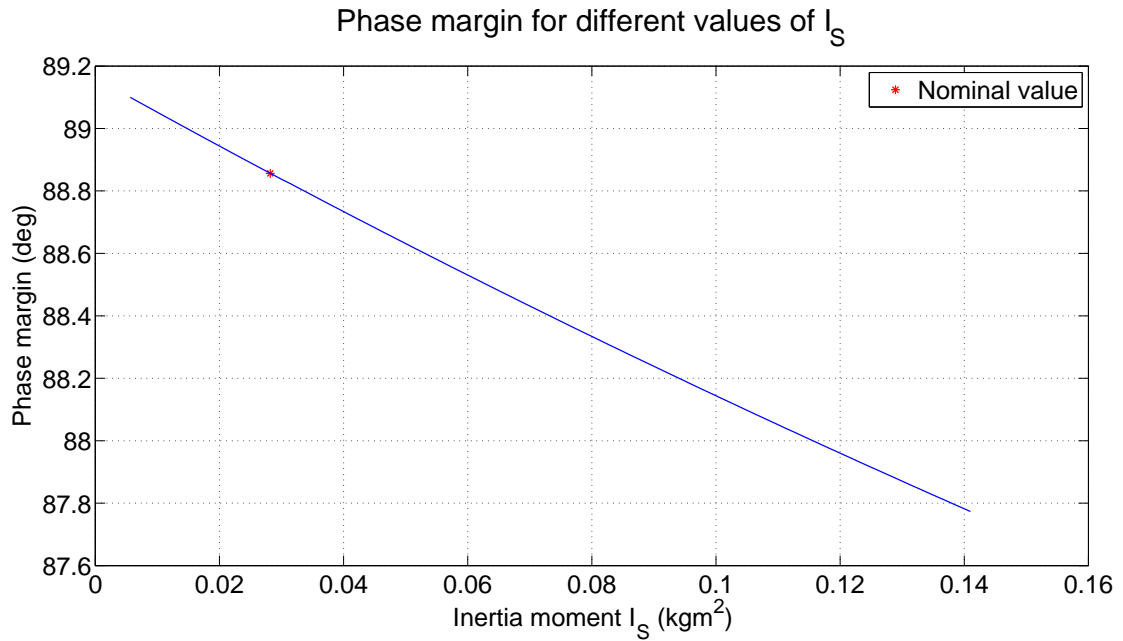
6.2 Sensitivity analysis with LQR controller

In this paragraph the sensitivity analysis, as described in Sec. 6.1, will be done with the developed LQR controller.

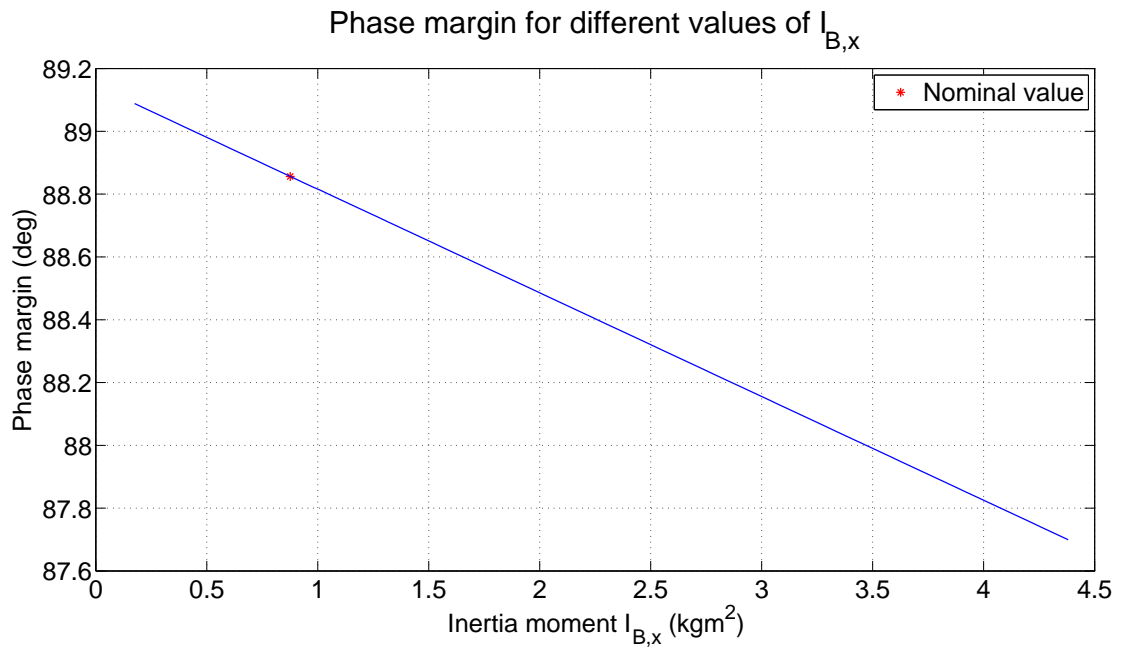
6.2.1 Uncertainties in I_S , $I_{B,x}$, $I_{B,y}$ and $I_{B,z}$

Firstly, the system is simulated from 20% up to 500% of the nominal value of I_S , $I_{B,x}$ and $I_{B,z}$ ($I_{B,x}$ and $I_{B,y}$ are identical, so the simulation for $I_{B,y}$ is left out). It turns out that for all of these uncertainties in I_S , $I_{B,x}$ and $I_{B,z}$ the total system remains stable.

Also, for analyzing the stability of the balancing controller, the gain and phase margin are calculated. For all uncertainties in I_S and $I_{B,x}$ the gain margins are infinite and the phase margins are shown in Fig. 6.1.



a)



b)

Figure 6.1: LQR controller: The phase margin for different values of the moment of inertia. a) I_S . b) $I_{B,x}$.

Fig. 6.1 shows that the deviation in phase margin from the nominal value for both I_S and $I_{B,x}$ is at most 2%. These results prove that the system is robust to decreases of more than 80% and increases of more than 500% of the nominal values of I_S , $I_{B,x}$ and $I_{B,z}$.

The required torque for an inverse step response of the roll angle will not be calculated in this case, because this simulation only analyzes uncertainties in estimated model parameters and does not analyze possible future changes.

6.2.2 Uncertainties in m_{B1} and l

Secondly, the system is simulated for a large range of values of m_{B1} and l . The results are displayed in a grid, shown in Fig. 6.2, where the gray region indicates the values of m_{B1} and l for which the system remains stable.

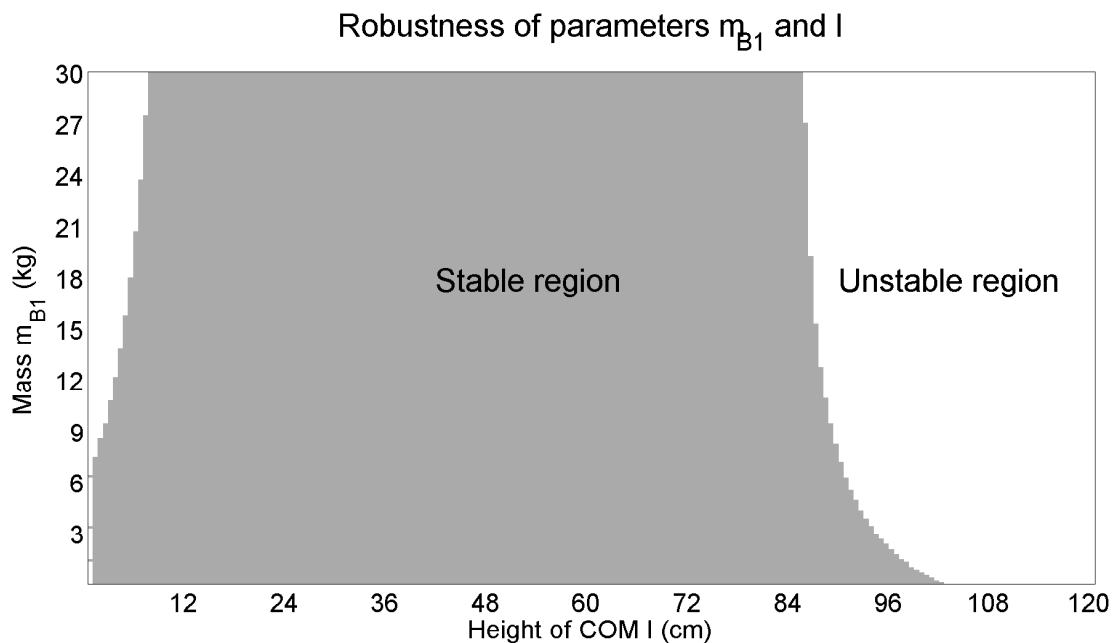


Figure 6.2: LQR controller: Stability of the system for uncertainties in model parameters m_{B1} and l .

Fig. 6.2 shows that increasing the mass of the upper part of the robot up to 30 kg doesn't influence the stability of the robot, provided that the height of the COM is between 10 and 85 cm. However, increasing the height of the COM above 85 cm quickly results in instability for almost all values of the mass of the upper part of the robot.

Concerning the balancing controller, the gain margin is infinite for all values of m_{B1} and l for which the system is stable and the phasemargin is shown in Fig. 6.3.

Fig. 6.3 shows that for the values of m_{B1} and l for which the system is stable, the phase margin remains close to its nominal value. Fig. 6.3 also shows that for values of l above about 90 cm, the stability of the total system seems to be determined by either the position controller or the controller for the yaw angle.

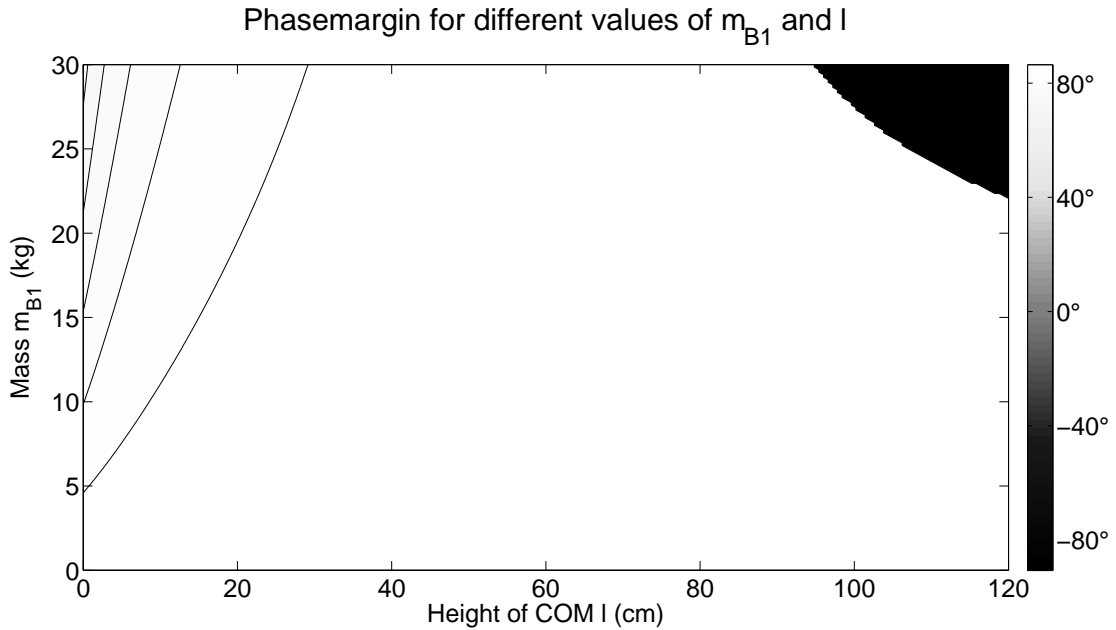


Figure 6.3: LQR controller: The phase margin for different values of m_{B1} and l .

There is no obvious stable worst-case scenario to check whether torque saturation will destabilize the system, but an increase of mass is assumed to be more likely than an increase of the height of the COM. Therefore, the worst-case scenario is chosen to be the stable situation with maximal mass and the corresponding maximal height of the COM, because this seems to be the most realistic worst-case scenario. This corresponds to $m_{B1} = 30$ kg and $l = 85$ cm. The required torque for an inverse step response of the roll angle is calculated for this scenario and is shown in Fig. 6.4 together with the inverse step response of the roll angle. According to Fig. 6.4, torque saturation will not destabilize the system, so (based on the developed model) increasing the mass of the upper part of the robot up to 30 kg with a height of the COM of at most 85 cm is possible with the current motors.

6.2.3 Uncertainties in m_S and α

Thirdly, the system is simulated for values of m_S between 0.5 kg and 5 kg. It turns out that for all values of m_S the system remains stable. Concerning the balancing controller, the gain margin is infinite for all values of m_S and the phase margin is shown in Fig. 6.5.

Fig. 6.5 shows that the maximal deviation in phasemargin from the nominal value of m_S is only 1%.

The required torque for an inverse step response of the roll angle has been calculated for the case $m_S = 5$ kg and is shown in Fig. 6.4 together with the inverse step response of the roll angle.

According to Fig. 6.4, torque saturation will not destabilize the system, so (based on the developed model) replacing the ball by a lighter or heavier ball is fine with these motors, as long as the ball is rough and stiff enough and has the same diameter as the original ball.

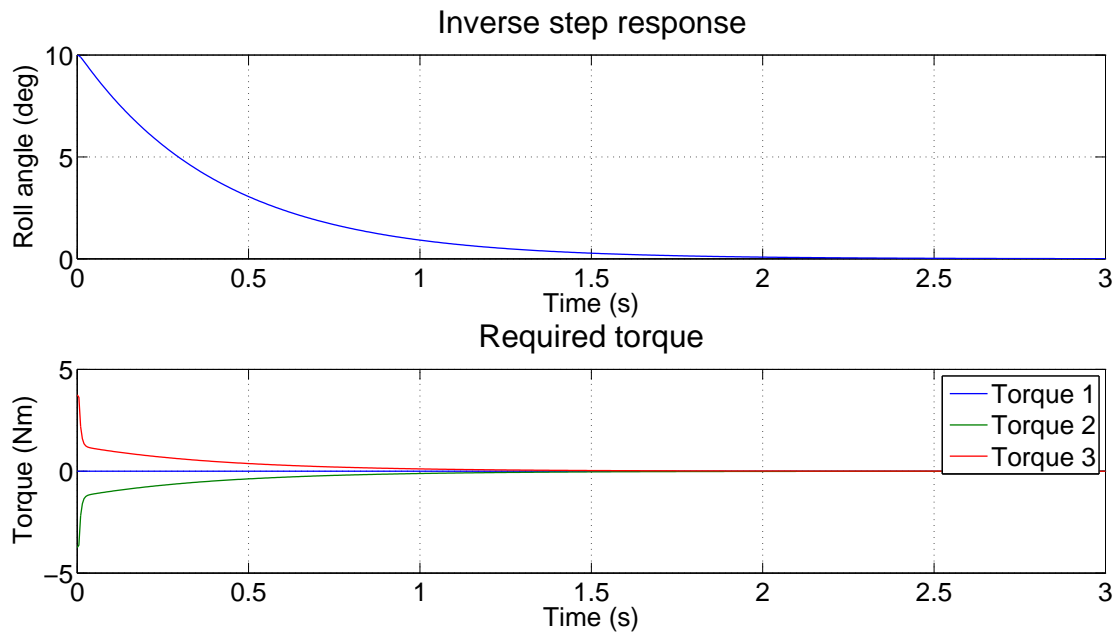


Figure 6.4: LQR controller: Required torque for the inverse step response of the roll angle with $m_{B1} = 30$ kg and $l = 85$ cm.

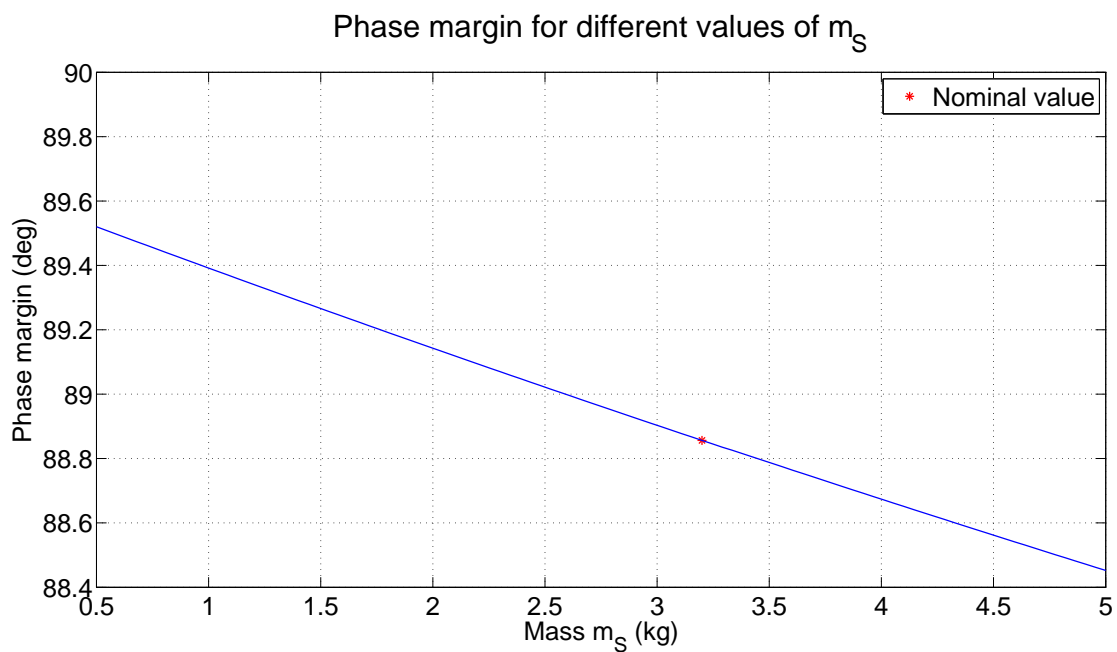


Figure 6.5: LQR controller: The phase margin for different values of m_s .

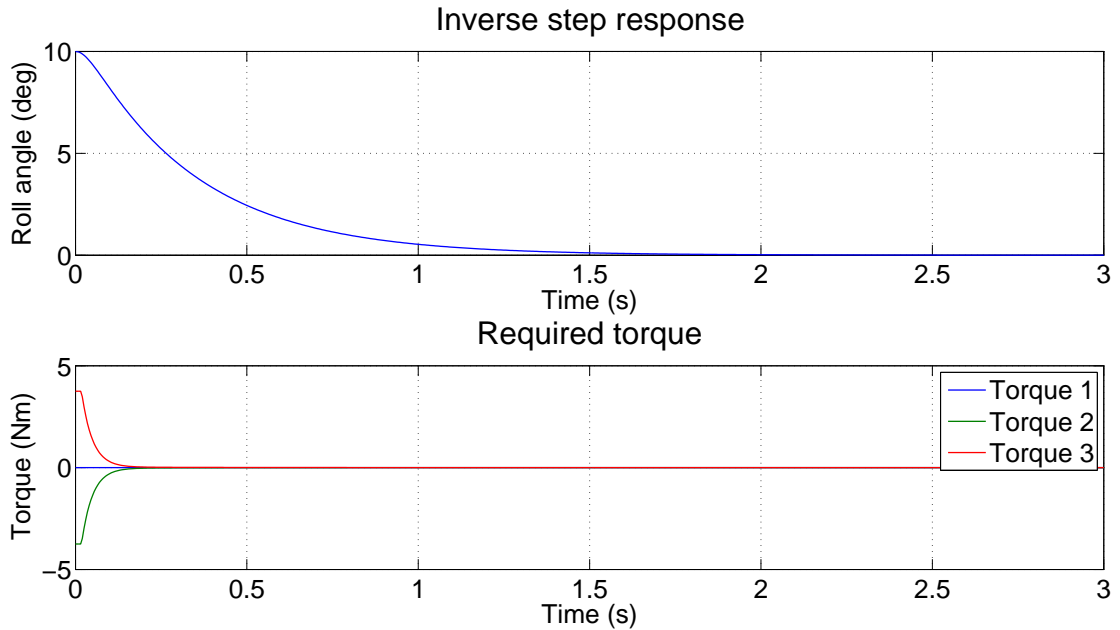


Figure 6.6: LQR controller: Required torque for the inverse step response of the roll angle with $m_s = 5$ kg.

The system is also simulated for α between 0° and 90° . For $\alpha = 0^\circ$ the system is of course unstable, because it is impossible to control the yaw angle. Moreover, it is practically impossible with three omniwheels. For $\alpha > 0^\circ$ the system remains stable up to $\alpha = 87^\circ$. Concerning the balancing controller, the gain margin is infinite for all values of α and the phase margin is shown in Fig. 6.7.

Fig. 6.7 shows that the phasemargin concerning α stays close to its nominal value for α up to 80° . Above 85° the system quickly becomes unstable.

The stable worst-case is the situation in which $\alpha = 87^\circ$, but this will never happen in practice, because in that case it is hardly possible to control the pitch and roll angle. To have a more realistic stable worst-case scenario, α is chosen to be 80° . The required torque for an inverse step response of the roll angle is calculated for this scenario and is shown in Fig. 6.8 together with the inverse step response of the roll angle.

According to Fig. 6.8, torque saturation will not destabilize the system, so (based on the developed model) increasing the angle α up to 80° is possible with the current motors.

6.3 Sensitivity analysis with SISO controller

In this paragraph the sensitivity analysis, as described in Sec. 6.1, will be done with the developed SISO controller.

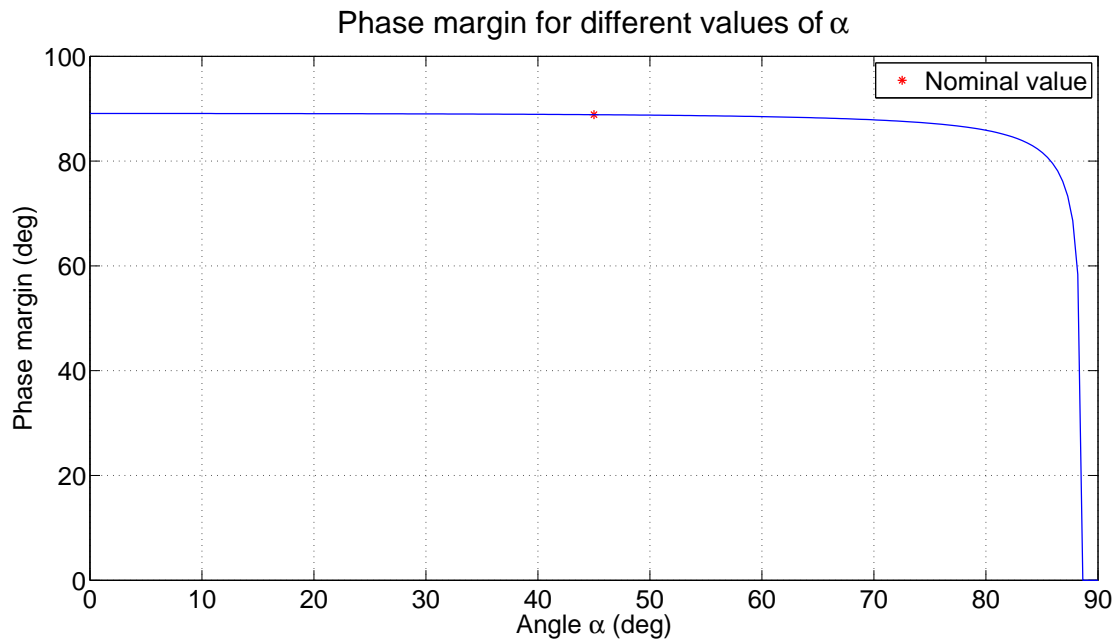


Figure 6.7: LQR controller: The phase margin for different values of α .

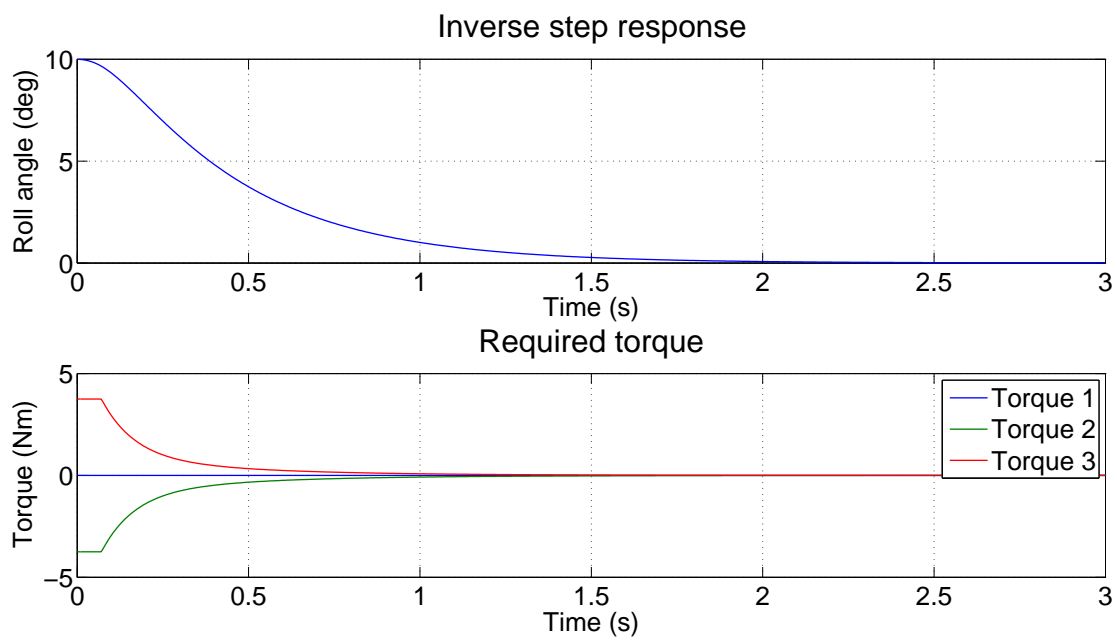


Figure 6.8: LQR controller: Required torque for the inverse step response of the roll angle with $\alpha = 80^\circ$.

6.3.1 Uncertainties in I_S , $I_{B,x}$, $I_{B,y}$ and $I_{B,z}$

Firstly, the system is simulated from 20% up to 500% of the nominal value of I_S , $I_{B,x}$ and $I_{B,z}$ ($I_{B,x}$ and $I_{B,y}$ are identical, so the simulation for $I_{B,y}$ is left out). It turns out that for all of these uncertainties in I_S , $I_{B,x}$ and $I_{B,z}$ the total system remains stable.

Also, for analyzing the stability of the balancing controller, the gain and phase margin are calculated. For all uncertainties in I_S and $I_{B,x}$ the gain margins are infinite and the phase margins are shown in Fig. 6.9.

Fig. 6.9 shows that the deviation in phase margin from the nominal value for both I_S and $I_{B,x}$ is at most 8%. These results prove that the system is robust to decreases of more than 80% and increases of more than 500% of the nominal values of I_S , $I_{B,x}$ and $I_{B,z}$.

Analogous to the simulations with the LQR controller, the required torque for an inverse step response of the roll angle will not be calculated in this case, because this simulation only analyzes uncertainties in estimated model parameters and does not analyze possible future changes.

6.3.2 Uncertainties in m_{B1} and l

Secondly, the system is simulated for a large range of values of m_{B1} and l . The results are displayed in a grid, shown in Fig. 6.10, where the gray region indicates the values of m_{B1} and l for which the system remains stable.

Fig. 6.10 shows that increasing the mass of the upper part of the robot up to 30 kg doesn't influence the stability of the robot, provided that the height of the COM is between 10 and 95 cm. If the height of the COM is above 95 cm, the stability of the system depends on the value of the mass of the upper part of the robot.

Concerning the balancing controller the gain margin is infinite for all values of m_{B1} and l for which the system is stable and the phasemargin is shown in Fig. 6.11.

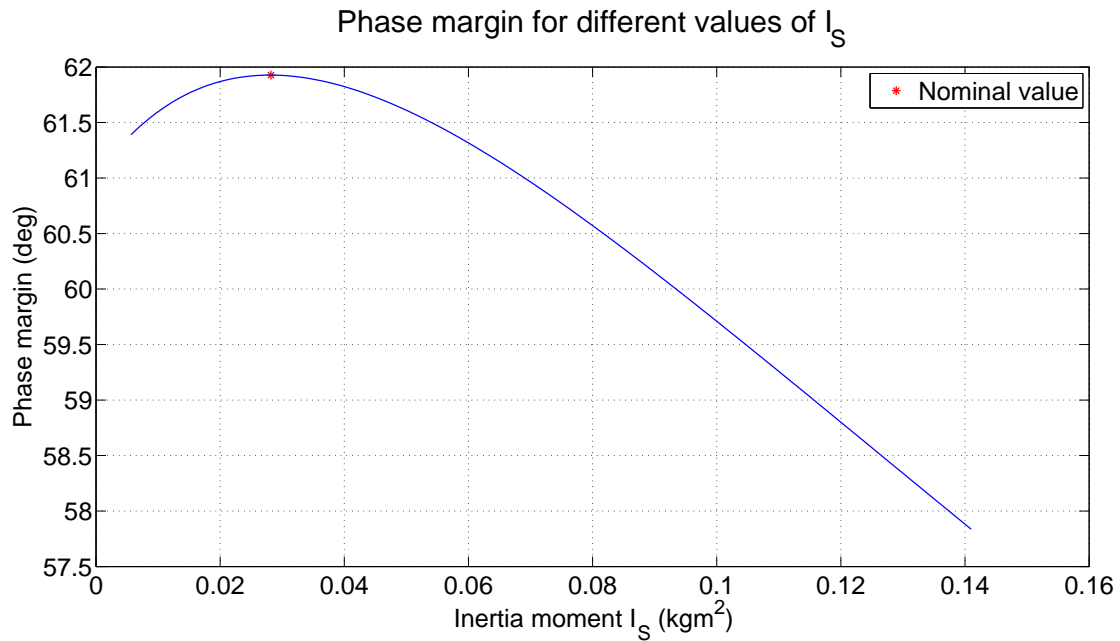
Fig. 6.11 shows that for the values of m_{B1} and l for which the system is stable, the phase margin remains between about 30° and its nominal value of about 62°. Fig. 6.11 also shows that the stability of the total system seems to be determined by the balancing controller, in contrast to the LQR controller.

Analogous to the simulations with the LQR controller, the worst-case scenario is chosen to be the stable situation with maximal mass and the corresponding maximal height of the COM. This corresponds to $m_{B1} = 30$ kg and $l = 95$ cm. The required torque for an inverse step response of the roll angle is calculated for this scenario and is shown in Fig. 6.12 together with the inverse step response of the roll angle.

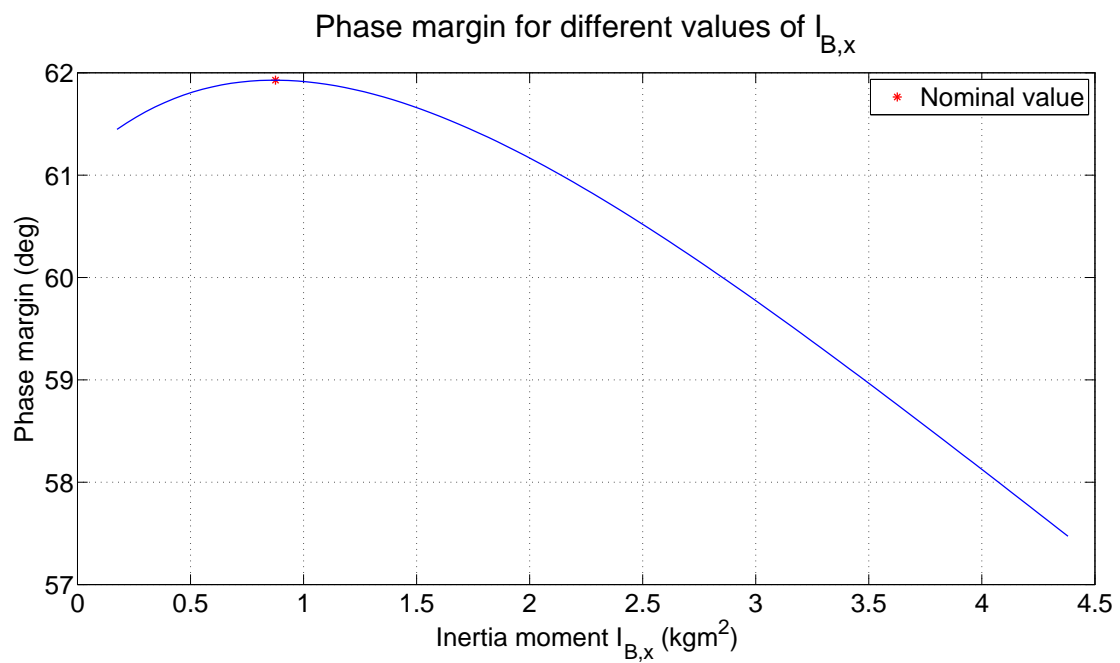
According to Fig. 6.12, torque saturation will destabilize the system, so (based on the developed model) increasing the mass of the upper part of the robot up to 30 kg with a height of the COM of 95 cm seems to be impossible with the current motors. It turns out that with $m_{B1} = 30$ kg, torque saturation does not happen for the inverse step response of the pitch angle, if the height of the COM is at most 92 cm.

6.3.3 Uncertainties in m_S and α

Thirdly, the system is simulated for values of m_S between 0.5 kg and 5 kg. For all values of m_S the system remains stable. Concerning the balancing controller, the gain margin is infinite for



a)



b)

Figure 6.9: SISO controller: The phase margin for different values of the moment of inertia. a) I_S . b) $I_{B,x}$.

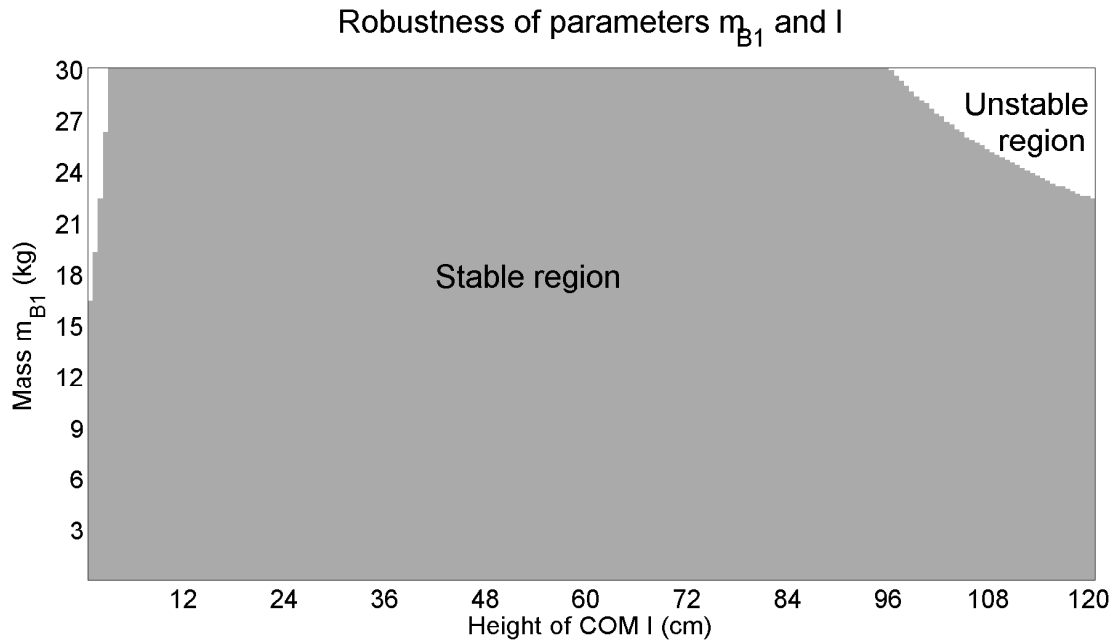


Figure 6.10: SISO controller: Stability of the system for uncertainties in plant parameters m_{B1} and l .

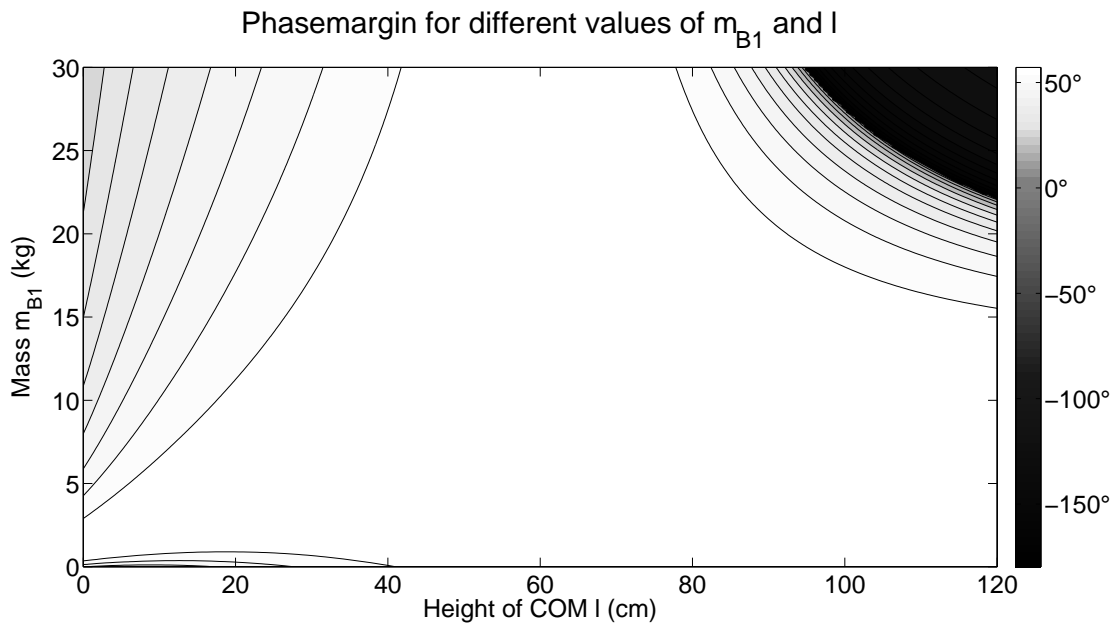


Figure 6.11: SISO controller: The phase margin for different values of m_{B1} and l .

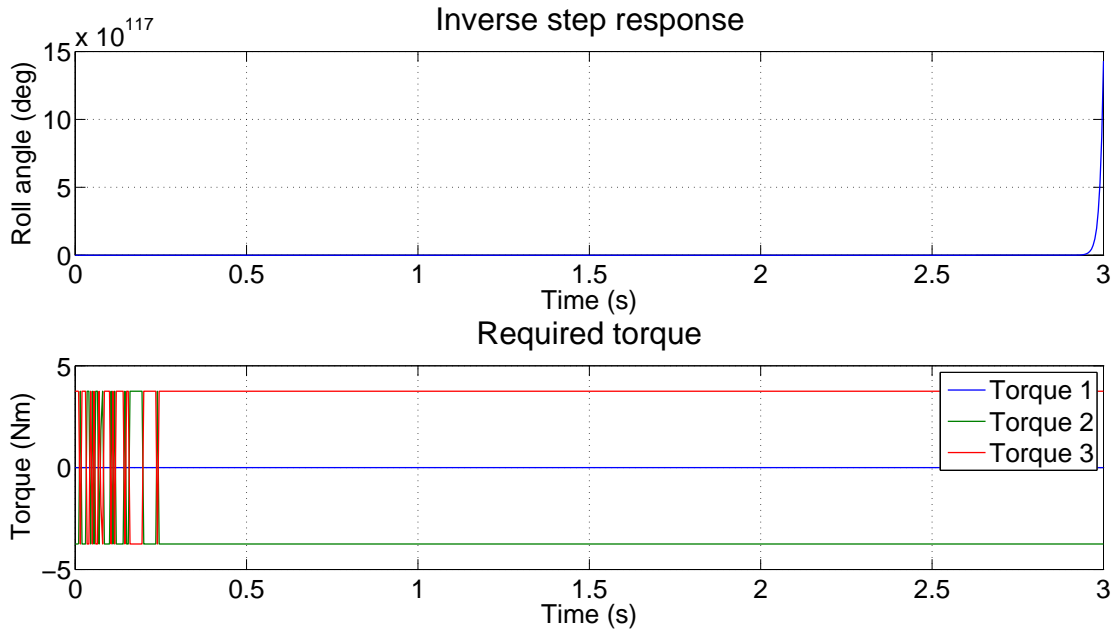


Figure 6.12: SISO controller: Required torque for the inverse step response of the roll angle with $m_{B1} = 30$ kg and $l = 95$ cm.

all values of m_S and the phase margin is shown in Fig. 6.13.

Fig. 6.13 shows that the deviation in phasemargin from the nominal value of m_S is at most 12%. The required torque for an inverse step response of the roll angle has been calculated for the case $m_S = 5$ kg and is shown in Fig. 6.14 together with the inverse step response of the roll angle.

According to Fig. 6.14, torque saturation will not destabilize the system, so (based on the developed model) replacing the ball by a lighter or heavier ball is fine with these motors, as long as the ball is rough and stiff enough and has the same diameter as the original ball.

The system is also simulated for α between 0° and 90° . For $\alpha = 0^\circ$ the system is of course unstable, because it is impossible to control the yaw angle. Moreover, it is practically impossible with three omniwheels. For $\alpha > 0^\circ$ the system remains stable up to $\alpha = 88^\circ$. Concerning the balancing controller, the gain margin is infinite for all values of α and the phase margin is shown in Fig. 6.15.

Fig. 6.15 shows that the phasemargin concerning α stays close to its nominal value for α up to 65° . Above 85° the system quickly becomes unstable.

Analogous to the simulations with the LQR controller, the required torque for an inverse step response of the roll angle has been calculated for the case $\alpha = 80^\circ$ and is shown in Fig. 6.16 together with the inverse step response of the roll angle.

According to Fig. 6.16, torque saturation will not destabilize the system, so (based on the developed model) increasing the angle α up to 80° is possible with the current motors.

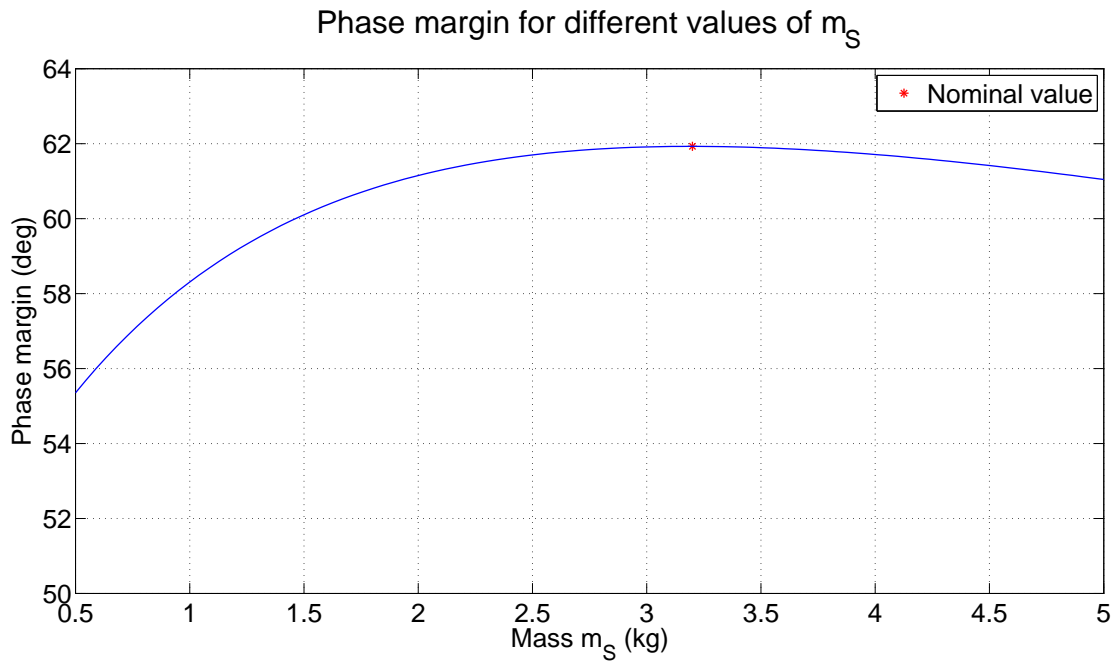


Figure 6.13: SISO controller: The phase margin for different values of m_s .

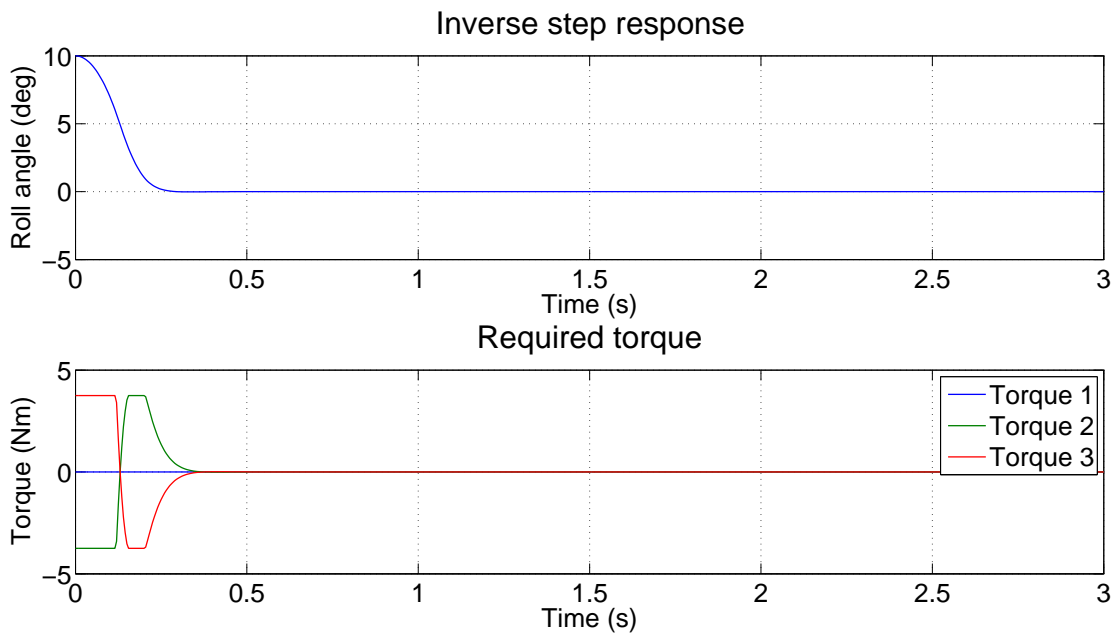


Figure 6.14: SISO controller: Required torque for the inverse step response of the roll angle with $m_s = 5$ kg.

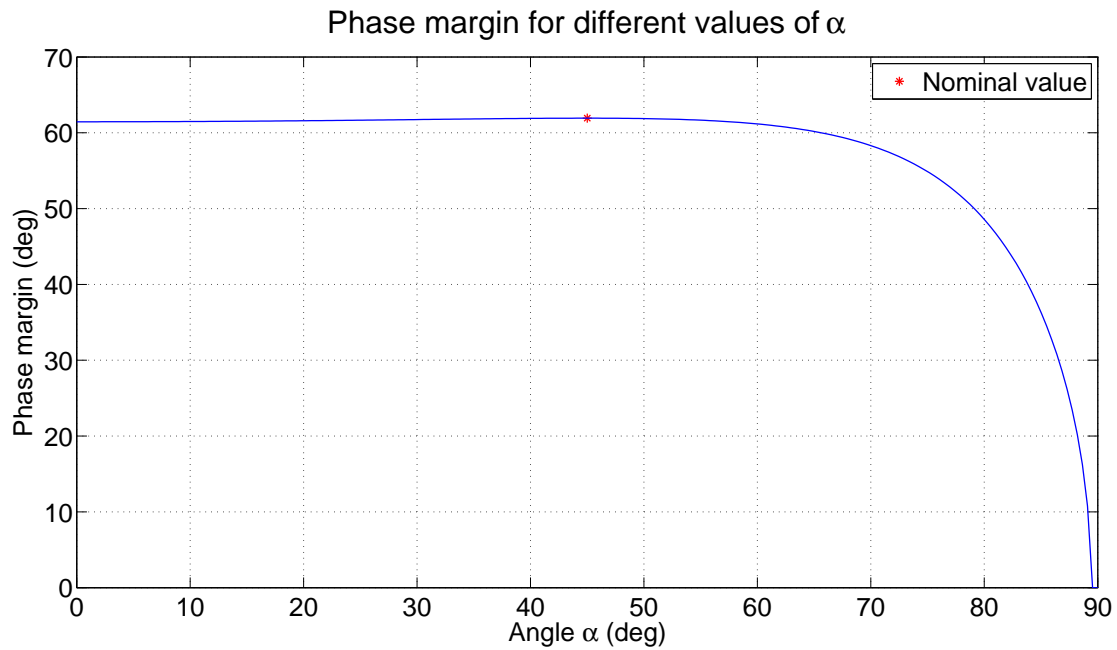


Figure 6.15: SISO controller: The phase margin for different values of α .

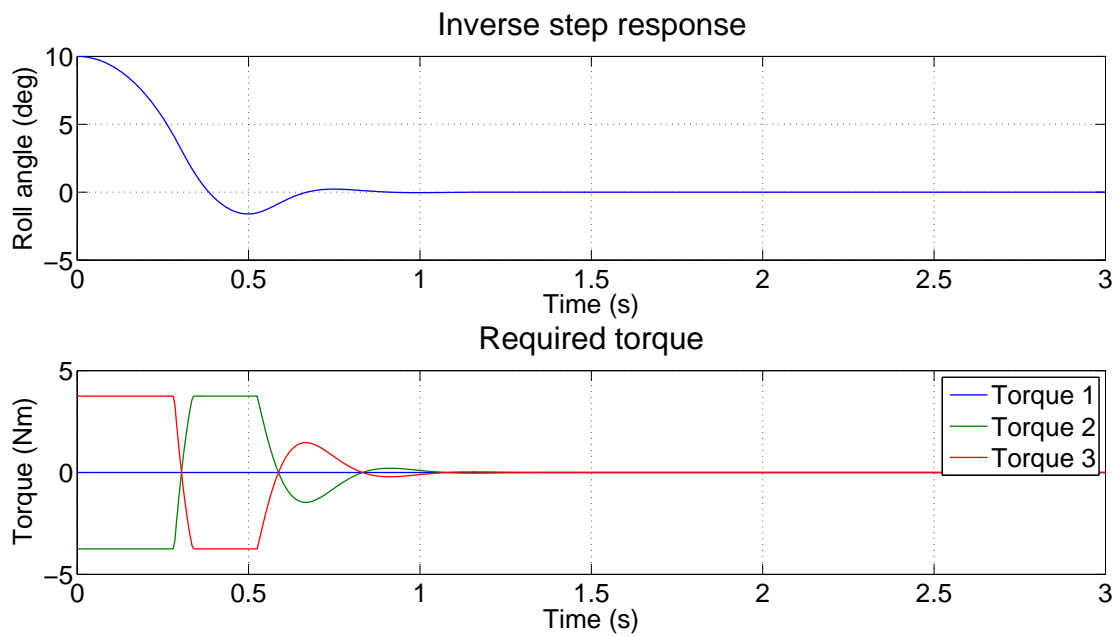


Figure 6.16: SISO controller: Required torque for the inverse step response of the roll angle with $\alpha = 80^\circ$.

6.4 Conclusion

The aim of this chapter was to investigate under which requirements the system remains stable with the developed controller. Due to the fact that the real system is not stable, the research is done with the developed dynamical model. This means that no quantitative conclusions can be made for the real system, based on this research. The results of this research only give an indication for the real system.

The sensitivity analysis shows that uncertainties in estimations of the moments of inertia do not influence the stability of the system, so the system is robust to uncertainties in the moments of inertia.

With regard to possible future changes, it can be concluded that adding some mass to the real system will most likely be no problem. Also, increasing the height of the COM also does not seem to be a problem for the stability of the system. Furthermore, replacing the ball with a lighter or somewhat heavier ball is fine, as long as the ball is rough and stiff enough and has the same diameter as the original ball. Finally, a small change for the angle α is fine, but for large changes no conclusions can be made for the real system.

Comparing the results of the sensitivity analysis done with the LQR controller to the results of the sensitivity analysis done with the SISO controller concerning the balancing controller shows that the system with the LQR controller is more robust to uncertain parameters, with regard to the phase margin, than the system with the SISO controller. This is mainly caused by the extra pole that is introduced by the lead compensator, which results in a lower maximum of phase lead and a faster decrease of phase above the bandwidth. Also, torque saturation happens more frequently with the SISO controller than with the LQR controller, which is in line with the expectations, because the SISO controller focuses on performance and will therefore use the motors maximally to have maximal performance.

On the other hand, comparing the settling time of the inverse step responses with the LQR controller and the SISO controller, proves that the settling time is much lower with the SISO controller, which proves that the performance of the system with regard to balancing is significantly higher with the SISO controller.

7. CONCLUSION & RECOMMENDATIONS

7.1 Conclusion

The goal of this project is to model and control a Ball-Balancing Robot, such that it is able to balance and move around as a demonstrator on fairs. To achieve this goal, the dynamical behaviour of the robot is approximated by three independent 2D models. The motors and omni-wheels are modeled as virtual motors and wheels in order to approximate the 3D behaviour in a 2D model and conversions are made for the torques and moments of inertia. Due to the conversions that need to be made and the neglected dependencies between the 2D models, like centrifugal forces, a 3D model is developed to describe the full dynamical behaviour of the BBR and it is linearized around the position the BBR stands upright.

Based on this linearized 3D model, a linear controller is designed. The approach taken in the design of the controller is to design first a simple, easily implementable controller to make the BBR 'fair ready' as soon as possible. Later, a more advanced controller is designed, which focuses on performance.

The first controller is a static controller, designed with LQR control theory, which controls all states simultaneously. The second controller is a dynamic cascade controller, designed with SISO loopshaping, which consists of an inner loop for controlling the orientation angles of the body and an outer loop for position control.

Simulations proved that the performance of the system with regard to balancing is significantly higher with the controller, designed with SISO loopshaping, compared to the performance of the system with the controller, designed with LQR control theory. However, the performance of the system with regard to position control is similar for both controllers, because the performance is limited by the non-minimum phase behaviour of the robot.

During the implementation of the LQR controller, it turned out that noisy sensor data prevents the controller from being implemented as desired. More specifically, the magnitude of the controller gains for the pitch and roll angular rates is seriously limited by noisy gyroscope data.

To reduce the noise of the gyroscope data, different solutions are tried like placing foam under the sensorboard to attenuate system vibrations and oversampling of the gyroscope in order to average multiple samples to one sample to reduce the noise. Furthermore, a FIR low-pass filter is implemented to limit the consequences of the noisy gyroscope data and to smoothen the torque inputs of the motors.

The implementation of these solutions significantly reduces the noise of the sensor data and results in a BBR that is able to balance.

Due to the fact that the controller mainly penalizes the errors in pitch and roll angles and only slightly penalizes the errors in pitch and roll angular rates, the BBR is only able to cope with small variations in pitch and roll angle. However, to be able to control the position of the BBR, such that the BBR is able to move around, the BBR must be able to cope with large variations in pitch and roll angles. Therefore, position control is impossible as long as the magnitude of the gains for the pitch and roll angular rates is limited by noisy sensor data.

Based on the performance of the system with the developed controller, it can be concluded that the developed model has proved to be a sufficient representation of reality for the design of a controller. The different experiments have shown that none of the model assumptions imposed a fundamental restriction on the performance of the robot.

Finally, research is done to investigate under what requirements the system remains stable with the developed controller. The research is done with the developed 3D model, so the results of the research are only indicative for the real system. From this research, it can be concluded that the system is robust to large variations in model parameters with the developed controller. Based on the 3D model, the system will most likely remain stable when new features are added to the robot in the future that will increase the mass of the robot and the height of the COM.

7.2 Recommendations & Future work

For further improvements on the performance of the BBR, several recommendations can be made.

First of all, to further reduce the noise of the sensor data and improve the performance of the system, it is recommended to design and implement a more advanced and effective filter to filter the sensor data, like for example a Kalman filter.

Secondly, it is recommended to improve the attenuation of system vibrations, to prevent them from being measured by the gyroscope.

Thirdly, brackets around the ball are recommended to increase the grip between the omni-wheels and the ball, which makes slip less likely to happen and makes it possible to allow larger tilt angles for higher performance.

Fourthly, system identification is recommended to improve the developed dynamical model, provided that the system is robustly stable.

Furthermore, the implementation of position control is left as future work. After a successful implementation of position control, the controller, designed with SISO loopshaping, can be implemented to increase the performance of the robot.

Also, an elaborate analysis on the influence of unmodeled parameters/phenomena (like deformation of the ball, slip and rolling friction) on the stability of the system with the developed controller, is left as future work, because a stable system is required to do useful experiments.

BIBLIOGRAPHY

- [1] O. Bosgra, H. Kwakernaak, and G. Meinsma. Effect of right-half plane open-loop zeros. In *Robust Control*, page 22. 2011.
- [2] P. Fankhauser and C. Gwerder. Modeling and control of a ballbot. Bachelor thesis, Swiss Federal Institute of Technology Zürich, 2010.
- [3] J. Fong and S. Uppill. Ballbot: Preliminary report. Mechatronics honours project, University of Adelaide, 2009.
- [4] G. Franklin, J. Powell, and A. Emami-Naeini. Bryson’s rule. In *Feedback Control of Dynamic Systems*, volume 5, page 493. Pearson, 2006.
- [5] A. N. Inal, O. Morgul, and U. Saranli. A 3d dynamic model of a spherical wheeled self-balancing robot. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5381–5386. IEEE, 2012.
- [6] M. Kumaga and T. Ochiai. Development of a robot balanced on a ball: Application of passive motion to transport. In *Robotics and Automation, 2009. ICRA ’09. IEEE International Conference on*, pages 4106–4111. IEEE, 2009.
- [7] M. Kumagai and T. Ochiai. Development of a robot balancing on a ball. In *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, pages 433–438. IEEE, 2008.
- [8] T. Lauwers, G. A. Kantor, and R. L. Hollis. A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2884–2889. IEEE, 2006.
- [9] Z. Li, S. S. Sastry, and R. Murray. *A mathematical introduction to robotic manipulation*. CRC Press, 1994.
- [10] C.-W. Liao, C.-C. Tsai, Y. Y. Li, and C.-K. Chan. Dynamic modeling and sliding-mode control of a ball robot with inverse mouse-ball drive. In *SICE Annual Conference, 2008*, pages 2951–2955. IEEE, 2008.
- [11] J. Meriam and L. Kraige. Properties of homogeneous solids. In *Engineering Mechanics: Dynamics*, volume 2, pages 699–702. John Wiley & Sons Incorporated, 2012.
- [12] J. Meriam and L. Kraige. Parallel axis theorem. In *Engineering Mechanics: Dynamics*, volume 2, pages 651–655. John Wiley & Sons Incorporated, 2012.
- [13] U. Nagarajan. Dynamic constraint-based optimal shape trajectory planner for shape-accelerated underactuated balancing systems. In *Robotics: Science and Systems*. IEEE, 2010.
- [14] U. Nagarajan, G. Kantor, and R. L. Hollis. Trajectory planning and control of an underactuated dynamically stable single spherical wheeled mobile robot. In *Robotics and Automation, 2009. ICRA ’09. IEEE International Conference on*, pages 3743–3748. IEEE, 2009.
- [15] U. Nagarajan, A. Mampetta, G. A. Kantor, and R. L. Hollis. State transition, balancing, station keeping, and yaw control for a dynamically stable single spherical wheel mobile

- robot. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 998–1003. IEEE, 2009.
- [16] U. Nagarajan, G. Kantor, and R. Hollis. Integrated planning and control for graceful navigation of shape-accelerated underactuated balancing mobile robots. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 136–141. IEEE, 2012.
- [17] U. Nagarajan, B. Kim, and R. Hollis. Planning in high-dimensional shape space for a single-wheeled balancing mobile robot with arms. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 130–135. IEEE, 2012.
- [18] M. Triantafyllou and F. Hover. Linear quadratic regulator. In *Maneuvering and control of marine vehicles*, pages 92–98. MIT OpenCourseWare, 2003.
- [19] C.-C. Tsai, C.-K. Chan, and L.-C. Kuo. Lqr motion control of a ball-riding robot. In *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on*, pages 861–866. IEEE, 2012.
- [20] I. G. Witvoet. Motion control: Digital filters. University Lecture, Eindhoven University of Technology, 2011.

A. PARAMETERS

Table A.1: Model parameters.

Parameter	Description	Value
	General	
m_S	Mass of the ball	3.2 kg
m_{OW}	Mass of an omni-wheel (including clamping bush and flange)	0.555 kg
m_{gh}	Mass of a gearhead	0.160 kg
m_M	Mass of a motor	0.240 kg
r_S	Radius of the ball	0.115 m
w_B	Width of the body of the robot	0.2 m
h	Height of the body of the robot	0.80 m
h_1	Height of the upper part of the body	0.66 m
h_2	Height of the lower part of the body	0.14 m
I_S	Moment of inertia of the ball	$2.65 \cdot 10^{-2} \text{ kgm}^2$
I_M	Moment of inertia of the rotor of the motor	$2.42 \cdot 10^{-6} \text{ kgm}^2$
I_{OW}	Moment of inertia of the omni-wheel	$6.94 \cdot 10^{-4} \text{ kgm}^2$
l	Distance between COM of the ball and COM of the body	0.405 m
g	Gravitational constant	9.81 m/s ²
α	Angle the omni-wheels make with the top of the ball	45°
β	Angle that determines the horizontal position of the omni-wheels	0°
	2D model	
m_B	Mass of the body	7.135 kg
m_{B1}	Mass of the upper part of the body	6.2 kg
m_{B2}	Mass of the lower part of the body	0.935 kg
m_W	Mass of a virtual actuating wheel	0.995 kg
r_W	Radius of virtual actuating wheel	0.050 m
I_W	Moment of inertia of the virtual actuating wheel in yz -/ xz -plane	$1.90 \cdot 10^{-3} \text{ kgm}^2$
$I_{W,xy}$	Moment of inertia of the virtual actuating wheel in xy -plane	$3.81 \cdot 10^{-3} \text{ kgm}^2$
I'_B	Moment of inertia of the body about rotation axis in yz -/ xz -plane	2.40 kgm ²
$I_{B,xy}$	Moment of inertia of the body in xy -plane	$4.76 \cdot 10^{-2} \text{ kgm}^2$
	3D model	
m_B	Mass of the body	10.0 kg
m_{B1}	Mass of the upper part of the body	6.2 kg
m_{B2}	Mass of the lower part of the body	0.935 kg
$m_{B2,tot}$	Mass of the lower part of the body	3.8 kg
r_W	Radius of omni-wheel	0.050 m
$I_{B,x}$	Moment of inertia of the body about the x -axis of the coordinate frame centered in its COM	$8.76 \cdot 10^{-1} \text{ kgm}^2$
$I_{B,y}$	Moment of inertia of the body about the x -axis of the coordinate frame centered in its COM	$8.76 \cdot 10^{-1} \text{ kgm}^2$
$I_{B,z}$	Moment of inertia of the body about the x -axis of the coordinate frame centered in its COM	$6.67 \cdot 10^{-2} \text{ kgm}^2$

Table A.2: Model variables.

Variable	Description	
	General	
ψ_j	Angle of the body of the robot around the j -axis	$j = x,y,z$
$\dot{\psi}_j$	Angular rate of the body of the robot around the j -axis	$j = x,y,z$
ϕ_i	Angle of omni-wheel i	$i = 1,2,3$
$\dot{\phi}_i$	Angular rate of omni-wheel i	$i = 1,2,3$
τ_i	Torque of omni-wheel i	$i = 1,2,3$
	2D model	
θ_j	Angle of the ball around the j -axis	$j = x,y$
$\dot{\theta}_j$	Angular rate of the ball around the j -axis	$j = x,y$
ϕ_j	Angle of the virtual actuating wheel around the j -axis	$j = x,y,z$
$\dot{\phi}_j$	Angular rate of the virtual actuating wheel around the j -axis	$j = x,y,z$
τ_j	Torque of the virtual actuating wheel around the j -axis	$j = x,y,z$
	3D model	
x_S	Translation of the ball along the x -axis	
\dot{x}_S	Linear speed of the ball along the x -axis	
y_S	Translation of the ball along the y -axis	
\dot{y}_S	Linear speed of the ball along the y -axis	
H_i^j	Homogeneous matrix from Ψ_i to Ψ_j	
R_i^j	Rotation matrix from frame Ψ_i to Ψ_j	
σ_i^j	Position of origin of Ψ_i expressed in Ψ_j	
$\mathbf{r}_k^{i,j}$	Position of Ψ_k w.r.t. Ψ_j , expressed in Ψ_i	
$\mathbf{v}_k^{i,j}$	Linear velocity of Ψ_k w.r.t. Ψ_j , expressed in Ψ_i	
$\boldsymbol{\omega}_k^{i,j}$	Angular velocity of Ψ_k w.r.t. Ψ_j , expressed in Ψ_i	
$\mathbf{T}_k^{i,j}$	Twist of Ψ_k w.r.t. Ψ_j , expressed in Ψ_i	
$Ad_{H_i^j}$	Adjoint matrix of H_i^j that maps twists from Ψ_i to Ψ_j	

B. DERIVATIONS

In this section derivations will be shown of equations, from which only the results were stated in Chap. 2 and Chap. 3.

B.1 Derivations 2D model

Derivation of $v_{W,yz}^T \cdot v_{W,yz}$

$$\begin{aligned}
 v_{W,yz}^T \cdot v_{W,yz} &= |v_{W,yz}|^2 \\
 &= \dot{y}_W^2 + \dot{z}_W^2 \\
 &= \left[\frac{d}{dt} (r_S \theta_x + (r_S + r_W) \sin \psi_x) \right]^2 + \left[\frac{d}{dt} ((r_S + r_W) \cos \psi_x) \right]^2 \\
 &= [r_S \dot{\theta}_x + (r_S + r_W) \dot{\psi}_x \cos \psi_x]^2 + [-(r_S + r_W) \dot{\psi}_x \sin \psi_x]^2 \\
 &= r_S^2 \dot{\theta}_x^2 + 2r_S(r_S + r_W) \dot{\theta}_x \dot{\psi}_x \cos \psi_x + (r_S + r_W)^2 \dot{\psi}_x^2 \cos^2 \psi_x \\
 &\quad + (r_S + r_W)^2 \dot{\psi}_x^2 \sin^2 \psi_x \\
 &= r_S^2 \dot{\theta}_x^2 + 2r_S(r_S + r_W) \dot{\theta}_x \dot{\psi}_x \cos \psi_x + (r_S + r_W)^2 \dot{\psi}_x^2.
 \end{aligned}$$

Derivation of $\dot{\phi}_x$

Before $\dot{\phi}_x$ can be derived, first ϕ_x needs to be found and expressed in the generalized coordinates. The virtual actuating wheel rotates either when the ball rotates or when the body rotates. It rotates in positive direction when the ball rotates in positive direction. Equating the traveled distance yields

$$r_W \phi_x = r_S \theta_x. \quad (\text{B.1})$$

The virtual actuating wheel rotates in positive direction when the body rotates in negative direction. Equating the traveled distance yields

$$r_W \phi_x = -r_S \psi_x. \quad (\text{B.2})$$

Combining Eq. (B.1) and Eq. (B.2) yields the following equation for the traveled distance of the virtual actuating wheel:

$$\begin{aligned}
 r_W \phi_x &= r_S \theta_x - r_S \psi_x \\
 &= r_S (\theta_x - \psi_x).
 \end{aligned} \quad (\text{B.3})$$

Dividing by r_w and taking the time derivative yields the following equation for $\dot{\phi}_x$:

$$\dot{\phi}_x = \frac{r_S}{r_W} (\dot{\theta}_x - \dot{\psi}_x).$$

Derivation of $v_{B,yz}^T \cdot v_{B,yz}$

$$\begin{aligned}
v_{B,yz}^T \cdot v_{B,yz} &= |v_{B,yz}|^2 \\
&= \dot{y}_b^2 + \dot{z}_b^2 \\
&= \left[\frac{d}{dt} (r_S \theta_x + l \sin \psi_x) \right]^2 + \left[\frac{d}{dt} (l \cos \psi_x) \right]^2 \\
&= [r_S \dot{\theta}_x + l \dot{\psi}_x \cos \psi_x]^2 + [-l \dot{\psi}_x \sin \psi_x]^2 \\
&= r_S^2 \dot{\theta}_x^2 + 2r_S l \dot{\theta}_x \dot{\psi}_x \cos \psi_x + l^2 \dot{\psi}_x^2 \cos^2 \psi_x + l^2 \dot{\psi}_x^2 \sin^2 \psi_x \\
&= r_S^2 \dot{\theta}_x^2 + 2r_S l \dot{\theta}_x \dot{\psi}_x \cos \psi_x + l^2 \dot{\psi}_x^2.
\end{aligned}$$

Derivation of $v_{W,xy}^T \cdot v_{W,xy}$

$$\begin{aligned}
v_{W,xy}^T \cdot v_{W,xy} &= |v_{W,xy}|^2 \\
&= \dot{x}_{W,xy}^2 + \dot{y}_{W,xy}^2 \\
&= \left[\frac{d}{dt} ((r_S + r_W) \cos(\psi_z)) \right]^2 + \left[\frac{d}{dt} ((r_S + r_W) \sin \psi_z) \right]^2 \\
&= [-(r_S + r_W) \dot{\psi}_z \sin \psi_z]^2 + [(r_S + r_W) \dot{\psi}_z \cos \psi_z]^2 \\
&= (r_S + r_W)^2 \dot{\psi}_z^2 \sin^2 \psi_z + (r_S + r_W)^2 \dot{\psi}_z^2 \cos^2 \psi_z \\
&= (r_S + r_W)^2 \dot{\psi}_z^2.
\end{aligned}$$

Derivation of $\dot{\phi}_z$

Analogous to the derivation of $\dot{\phi}_x$, first an expression for ϕ_z as a function of the generalized coordinate ψ_z needs to be found. This can be done by equating the traveled distances:

$$r_W \phi_z = -r_S \sin(\alpha) \psi_z.$$

Dividing by r_w and taking the time derivative yields the following equation for $\dot{\phi}_z$:

$$\dot{\phi}_z = -\frac{r_S}{r_W} \sin(\alpha) \dot{\psi}_z.$$

Derivations of moments of inertia of virtual actuating wheel

Energy equilibrium of rotational energies around x -axis:

$$\begin{aligned}
\frac{1}{2} I_{W,x} \dot{\phi}_x^2 &= \frac{1}{2} I_{OW} (\dot{\phi}_x \cos(\alpha))^2 + \frac{1}{2} I_M (k \dot{\phi}_x \cos(\alpha))^2 \\
&+ 2 \left(\frac{1}{2} I_{OW} \left(-\frac{1}{2} \dot{\phi}_x \cos(\alpha) \right)^2 + \frac{1}{2} I_M \left(-k \frac{1}{2} \dot{\phi}_x \cos(\alpha) \right)^2 \right) \\
&= \frac{1}{2} [I_{OW} \cos(\alpha)^2] \dot{\phi}_x^2 + \frac{1}{2} [I_M k^2 \cos(\alpha)^2] \dot{\phi}_x^2 \\
&+ \frac{1}{2} [2 I_{OW} \frac{1}{4} \cos(\alpha)^2] \dot{\phi}_x^2 + \frac{1}{2} [2 I_M \frac{1}{4} k^2 \cos(\alpha)^2] \dot{\phi}_x^2.
\end{aligned} \tag{B.4}$$

This yields the following formula for the moment of inertia of the virtual actuating wheel $I_{w,x}$ around the x -axis:

$$\begin{aligned} I_{W,x} &= \cos(\alpha)^2(I_{OW} + k^2I_M + \frac{1}{2}I_{OW} + \frac{1}{2}k^2I_M) \\ &= \frac{3}{2}\cos(\alpha)^2(I_{OW} + k^2I_M). \end{aligned} \quad (\text{B.5})$$

Energy equilibrium of rotational energies around y -axis:

$$\begin{aligned} \frac{1}{2}I_{W,y}\dot{\phi}_y^2 &= \frac{1}{2}I_{OW}\left(\frac{1}{2}\sqrt{3}\dot{\phi}_y\cos(\alpha)\right)^2 + \frac{1}{2}I_M\left(k\frac{1}{2}\sqrt{3}\dot{\phi}_y\cos(\alpha)\right)^2 \\ &+ \frac{1}{2}I_{OW}\left(-\frac{1}{2}\sqrt{3}\dot{\phi}_y\cos(\alpha)\right)^2 + \frac{1}{2}I_M\left(-k\frac{1}{2}\sqrt{3}\dot{\phi}_y\cos(\alpha)\right)^2 \\ &= \frac{1}{2}\left[2I_{OW}\frac{3}{4}\cos(\alpha)^2\right]\dot{\phi}_y^2 + \frac{1}{2}\left[2I_Mk^2\frac{3}{4}\cos(\alpha)^2\right]\dot{\phi}_y^2. \end{aligned} \quad (\text{B.6})$$

Analogous to $I_{W,x}$, the formula for $I_{W,y}$ will be

$$\begin{aligned} I_{W,y} &= \frac{3}{2}I_{OW}\cos(\alpha)^2 + \frac{3}{2}I_Mk^2\cos(\alpha)^2 \\ &= \frac{3}{2}\cos(\alpha)^2(I_{OW} + k^2I_M). \end{aligned} \quad (\text{B.7})$$

Energy equilibrium of rotational energies around z -axis:

$$\begin{aligned} \frac{1}{2}I_{W,z}\dot{\phi}_z^2 &= 3\left(\frac{1}{2}I_{OW}(\sin(\alpha)\dot{\phi}_z)^2 + \frac{1}{2}I_M(k\sin(\alpha)\dot{\phi}_z)^2\right) \\ &= \frac{1}{2}\left[3I_{OW}\sin(\alpha)^2\right]\dot{\phi}_z^2 + \frac{1}{2}\left[3I_Mk^2\sin(\alpha)^2\right]\dot{\phi}_z^2. \end{aligned} \quad (\text{B.8})$$

Analogous to $I_{W,x}$ and $I_{W,y}$, the formula for $I_{W,z}$ will be

$$I_{W,z} = 3\sin(\alpha)^2(I_{OW} + k^2I_M). \quad (\text{B.9})$$