Bachelor Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Usage of the LEGO Mindstorms EV3 Robot

Design and Realization of the "Ball-riding robot" for the Promotion of the Faculty

Jakub Malý

Supervisor: Ing. Martin Hlinovský, Ph.D.
May 2018

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Malý Jakub**	Personal ID number:	**457194**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Cybernetics and Robotics**		
Branch of study:	**Robotics**		

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Usage of the LEGO Mindstorms EV3 Robot - Design and Realization of the "Ball-riding robot" for the Promotion of the Faculty

Bachelor’s thesis title in Czech:

Využití robota LEGO Mindstorms EV3 - návrh a realizace robota "Ball-riding robot" pro propagační účely fakulty

Guidelines:

1. Learn about the Lego Mindstorms Education EV3 robot (current status, HW and SW equipment).
2. Design and implement the Ball-riding robot for promotional purposes of the faculty.
3. Create a website for the tasks you have done (task description, design of the controller or principle of operation, explanation of the proposed software, photogallery and, if necessary, instructions for building the robot).

Bibliography / sources:

[1] James Floyd Kelly - LEGO MINDSTORMS NXT-G programming Guide, Second Edition
[2] Daniele Benedettelli - Programming LEGO NXT Robots using NXC
[3] https://www.youtube.com/watch?v=mt-9AmD_yrk

Name and workplace of bachelor’s thesis supervisor:

Ing. Martin Hlinovský, Ph.D., Department of Control Engineering, FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **12.01.2018** Deadline for bachelor thesis submission: **25.05.2018**

Assignment valid until: **30.09.2019**

Ing. Martin Hlinovský, Ph.D. Supervisor’s signature	doc. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Ing. Pavel Ripka, CSc. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt Student’s signature

Acknowledgements

I would like to express my immense gratitude to my advisor Ing. Martin Hlinovský, Ph.D. for his patience, motivation, and continuous support throughout my Bachelor study. His constructive criticism and friendy advice helped me in times of need. Without the education provided by CTU FEL, the successful completion of this project would not have been possible. I thank all those who helped me in this work, and the school itself for the provision of the materials for robot construction.

Last, I would like to thank my family: my parents, my sister, and my girlfriend for supporting me continuously. This accomplishment would not have been possible without them.

Declaration

Ideclare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May 25, 2018

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 25. května 2018

Abstract

This thesis presents methodologies for modeling and motion control of a Ball-riding robot (further BRB). The LEGO Mindstorm EV3 building kit is used to create this robot, which is driven by three omnidirectional wheels (further omni-wheels) with a bowling ball serving as a balancing ball. Lagrangian mechanics is used to derive a dynamic model of the robot moving on a flat plane. To achieve stabilization in both the equilibrium point and general motion control, an LQR controller is developed and tuned.

Keywords: Bachelor thesis, Ball-riding robot, Lagrangian mechanics, LQR, LEGO Mindstorms EV3

Supervisor: Ing. Martin Hlinovský, Ph.D.
Katedra Řídicí Techniky,
Resslova 9,
Praha 2

Abstrakt

Tato bakalářská práce prezentuje metody pro modelování a ovládání pohybu Ball-riding robotu. K sestavení robotu je použita LEGO Mindstorm EV3 stavebnice. Robot je poháněn třemi všesměrovými koly a jakožto koule k balancování slouží bowlingová koule. Lagrangeovská mechanika je použita k získání dynamického modelu robotu, pohybujícího se na ploché rovině. K dosažení stabilizace v rovnovážném bodě a v celkovém pohybu, LQR regulátor je vytvořen a vyladěn.

Klíčová slova: Bakalářská práce, Ball-riding robot, Lagrangeova mechanika, LQR, LEGO Mindstorms EV3

Překlad názvu: Využití robota LEGO Mindstorms EV3 — návrh a realizace robota „Ball-riding robot“ pro propagační účely katedry

Contents

Figures

Tables

Chapter 1

Introduction

1.1 Why this thesis was created

At the beginning of the fifth semester of my studies at the Czech Technical University in Prague, I started looking for the topic of my bachelor thesis. As I have always been interested in the concept of robots, and there were several modules on the topic throughout my studies, I decided to create my own. I wrote to one of the lecturers of these subjects and asked if we could agree on a topic for my work. After receiving a warm reply, and meeting with the guarantor, the theme was agreed upon. My task is to design and realize a "Ball-riding robot", further BRB, for promotional purposes. Thus the thesis is a report of an accomplishment of this difficult challenge and presented as my final work, ending my three-year undergraduate study.

1.2 Thesis structure

The thesis consists of 7 chapters. Each chapter addresses an important issue and is divided into several sections and subsections. In the first chapter, an introduction to the subject takes place. The reason for creating this thesis has already been clarified, and the BRB system will be discussed in this chapter. The "2D model" chapter aims at modeling the BRB in decoupled plains and describes a reduced dynamic model and the basics of linearization. The reduced dynamic model is the main element in deriving the complex model used in the "3D model" chapter. After obtaining such a model, the linearization is done again and an appropriate controller is designed in the following chapter: "Controller design". The design of the robot and its two parts - the body and the ball - is listed in the "Robot design" chapter. In the chapter "Controller implementation", it is shown how the controller designed in the fourth chapter can be applied to a realistic robot. A conclusion is presented followed by the Appendices which contain derivations, a list of the content of the enclosed CD and finally an overview of all bibliography.

1.3 What is a BRB?

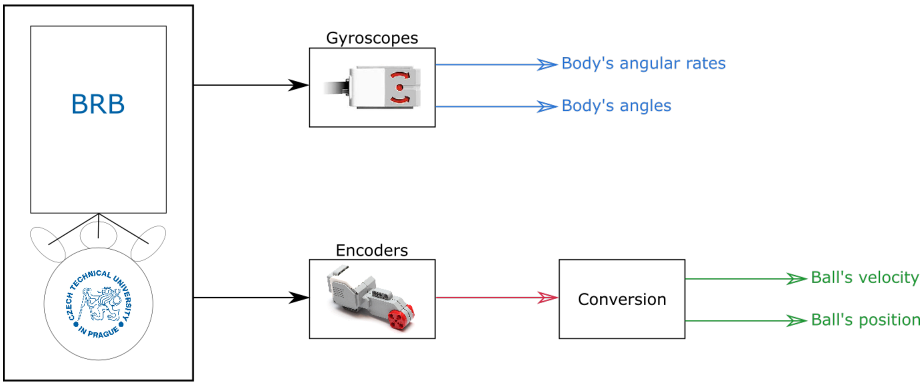


Figure 1.1: Simplified model of the BRB system

BRB, as already mentioned, is an abbreviation of Ball-Riding roBot. Ball-riding robots are robots which balance on top of a ball. Driving the ball and balancing the robot is done using multiple omni-wheels (usually 3 or 4) and angular sensors attached to the body of the robot. In our case, three omni-wheels will be used along with two gyroscopes for each axis to estimate angular rates and angles of the body. The velocity and the position of the ball is derived by mathematical operations using the angles of the omni-wheels. Due to the lack of a transmission between the omni-wheel and the motor, the rotation angle of a single motor is equal to the rotation angle of a single omni-wheel.

1.4 Previously created BRBs

The BRB is not a new concept in technical spheres. Several studies have been done by multiple universities, including:

- Carnegie Mellon University (CMU) in the United States developed the first BRB (shown in Fig. 1.2a) in 2006. The robot is human sized, and aimed at interaction with humans. In the final version of the robot, a pair of arms were added, and a total of 5 DC motors were needed to maintain system balance despite the fact that the robot was not able to rotate around the vertical axis z. The relevant references are [1], [2], [3], [4], and [5].

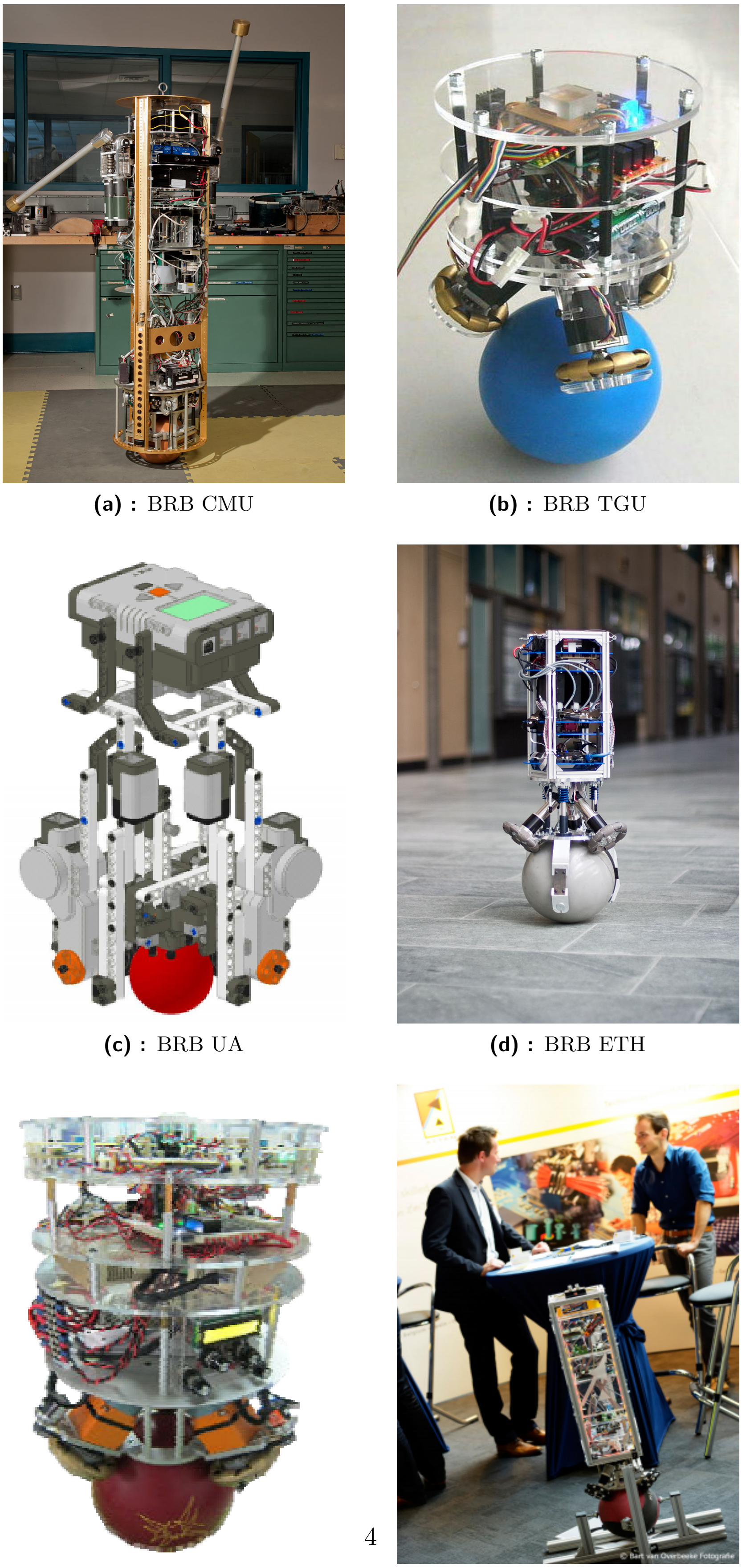
- Tohoku Gakuin University (TGU) in Japan developed a BRB (shown in Fig. 1.2b) in 2008. Only three motors were needed to complete the goals of the CMU robot as well as allowing rotation around the vertical axis z. This robot is able to carry loads in excess of 10kg. References are [6] and [7].

- The University of Adelaide (UA) in Australia developed a BRB using LEGO Mindstorms NXT (shown in Fig. 1.2c) in 2009. This robot is a small sized robot built completely out of LEGO. It uses a pair of normal wheels to balance on a plastic ball, which limits its performance and makes pivoting around the vertical axis z impossible. More in [8].
- The Swiss Federal Institute of Technology in Zurich (ETH) in Switzerland developed a BRB (shown in Fig. 1.2d) in 2010. The BRB was named Rezero and is the result of work by several students. Similar to the TGU robot, Rezero uses three omni-wheels to drive. In references [9], [12] a video is included, which demonstrates the robot's high dynamic robustness and the future goals for the project.
- The National Chung Hsing University (NCHU) in Taiwan developed a BRB (shown in Fig. 1.2e) in 2012. This robot is very similar to the TGU robot. It is about the same size, construction, and also uses three omni-wheels to perform drive actions. Corresponding references are [10].
- The University of Twente (UT) in Enschede in the Netherlands developed a BRB (shown in Fig. 1.2f) in 2014. The robot also uses only three omni-wheels, and is designed for use in fairs. The original text is given in [11].

When equations and procedures derived and developed in these studies have been used in this thesis, the original source was referenced in IEEE standard.

1.5 Main thesis goals

The goals of this bachelor thesis are: to describe system dynamics, derive a linearized model, develop a controller for point stabilization and trajectory tracking, build a robot from the LEGO EV3 building kit, adjust the bowling ball, and successfully implement the developed controller.



(a) : BRB CMU

(b) : BRB TGU

(c) : BRB UA

(d) : BRB ETH

(e) : BRB NCHU

(f) : BRB UT

Figure 1.2: An overview of previously created BRBs

Chapter 2

2D model

This chapter aims to describe the system structure and physical model of the BRB. The dynamic model with equations of motion is derived using Lagrangian mechanics and is then converted into a linearized model. While linearizing, an equilibrium point in a balancing position is used. While the 2D model is insufficient as an overall control solution, it does allow for a better understanding of dynamic problems, and serves as a basis for the 3D model.

2.1 Assumptions

Note that everything in this chapter is based on the following assumptions:

- **Rigid bodies**
  The BRB is composed of two rigid bodies: the body and the ball. Deformation of these bodies is negligible.

- **Rigid floor**
  Deformation of the floor is negligible.

- **Horizontal floor**
  The BRB moves only on horizontal floor, thus the ball has no potential energy.

- **Friction**
  Besides static friction, which guarantees the "No slip" assumption, all other types of frictions are negligible.

- **No slip**
  There is no slip between the body and the ball and between the ball and the floor.

- **Omni-wheels**
  Using 2-row omni-wheels with more than one contact point can be modeled as 1-row omni-wheels with a single contact point.

- **Negligible time delay**
  The time delay between the measurements of the sensors and the control of the motors is negligible.

- **Independent vertical planes**
  The two designed vertical planes (x-z, y-z) are assumed to be independent.

These or similar assumptions can be found in references up to [11].

2.2 Model description

To obtain a 2D model of a 3D system some simplifications, as done in [6], [9], [10], and [11], are needed. The first is that the whole model is divided into three planes (shown in Fig. 2.1): two vertical (x-z, y-z) and one horizontal (x-y). Each plane describes, using its generalized coordinates, a 2D motion which takes place within the plane. Secondly, the omni-wheels together with motors and shafts are modeled as virtual wheels. Each plane is affected by one virtual wheel, which rotates around the orthogonal axis to that plane. For the vertical planes, the ball is modeled as a disk, which rotates around the orthogonal axis to the plane. The virtual wheel and the centre of mass (further COM) of the body are attached to the rotation axis of the disk with a rigid rod. The body rotates around the rigid rod independent of the rotation of the disk, creating an inverted pendulum system (shown in Fig. 2.2a). For the horizontal plane, the ball is also modeled as a disk and rotates around the orthogonal axis to the plane. The virtual wheel is attached to the rotation axis of the disk with a rigid rod. The COM of the body is attached to the rod and shares the same coordinate system as the centre of the disk (shown in Fig. 2.2b). Lastly, the independence of rotational motion is assumed.

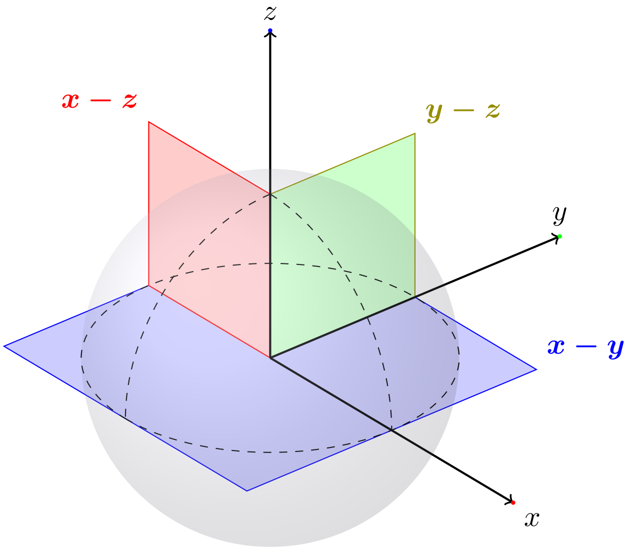
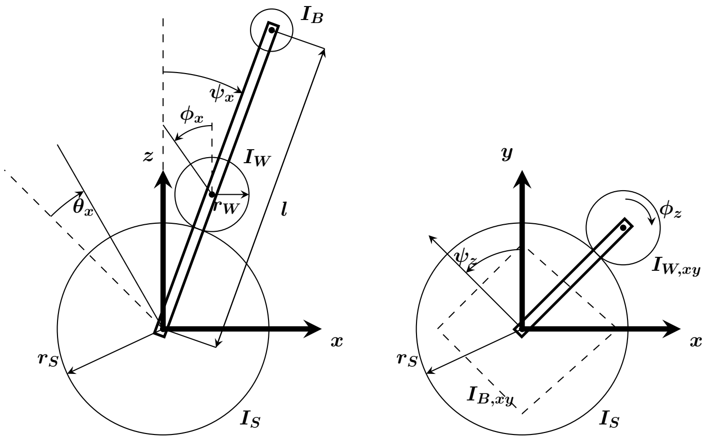


Figure 2.1: Illustration of 2D planes in 3D space



(a) : Sketch of the y-z plane **(b) :** Sketch of the x-y plane

Figure 2.2: Sketches of 2D planes - inspired by [11]

2.3 Coordinates

It is necessary to define the coordinates for future procedures. The coordinates will be defined as they are shown in Fig. 2.2.

2.3.1 Minimal coordinates

From Fig. 2.2, it can be seen that the vertical planes have two DOFs (Degrees of Freedom - the rotation of the ball and the body) and the horizontal plane has only one DOF (the rotation of the body). Hence the minimal coordinates can be defined as:

$$\vec{q}_{yz} = \begin{bmatrix} \theta_x \\ \psi_x \end{bmatrix} \tag{2.1}$$

$$\vec{q}_{xz} = \begin{bmatrix} \theta_y \\ \psi_y \end{bmatrix} \tag{2.2}$$

$$\vec{q}_{xy} = \begin{bmatrix} \psi_x \end{bmatrix} \tag{2.3}$$

2.3.2 Cartesian coordinates for the y-z/x-z planes

The position of the ball expressed in Cartesian coordinates:

$$\begin{bmatrix} y_S \\ z_S \end{bmatrix} = \begin{bmatrix} r_S\theta_x \\ 0 \end{bmatrix} \tag{2.4}$$

Of the body:

$$\begin{bmatrix} y_B \\ z_B \end{bmatrix} = \begin{bmatrix} r_S\theta_x + l\sin(\psi_x) \\ l\cos(\psi_x) \end{bmatrix} \tag{2.5}$$

Of the virtual wheel:

$$\begin{bmatrix} y_W \\ z_W \end{bmatrix} = \begin{bmatrix} r_S\theta_x + (r_S + r_W)\sin(\psi_x) \\ (r_S + r_W)\cos(\psi_x) \end{bmatrix} \tag{2.6}$$

For the x-z plane the Cartesian coordinates have the same form, substituting x for y and vice versa.

These equations, along with Eq. (2.7), appear in many texts, but their acquisition is considered to be elementary and familiar. Hence no reference is quoted.

2.3.3 Cartesian coordinates for the x-y plane

The body in the x-y plane only rotates, making its position invariable. Therefore, only the position of the virtual wheel needs to be determined:

$$\begin{bmatrix} x_{W,xy} \\ y_{W,xy} \end{bmatrix} = \begin{bmatrix} (r_S + r_W)\cos(\psi_z) \\ (r_S + r_W)\sin(\psi_z) \end{bmatrix} \tag{2.7}$$

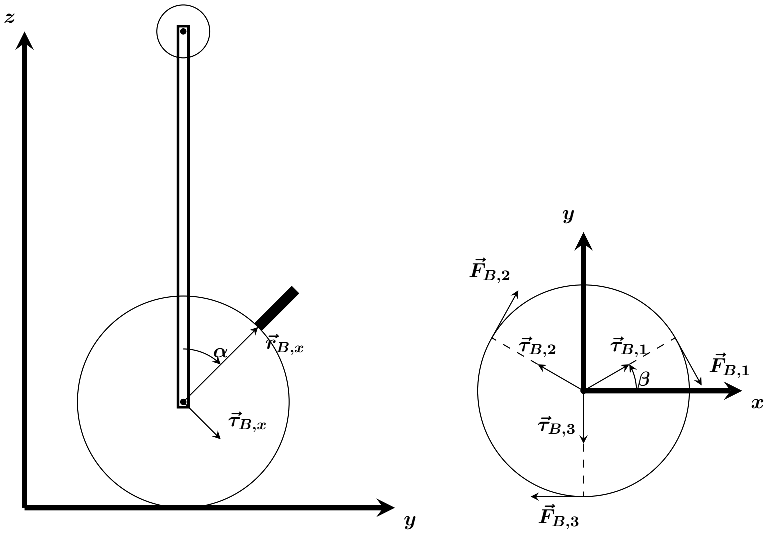
2.4 Conversion of virtual parameters

Before the equations of motion can be looked at, it is necessary to clarify the relation between torques of virtual wheels and torques of real omni-wheels and vice versa. As in [9] and [11], it can be shown that the resulting torque of the body is conserved. Mathematically:

$$\vec{\tau}_{B,x} + \vec{\tau}_{B,y} + \vec{\tau}_{B,z} = \vec{\tau}_{B,1} + \vec{\tau}_{B,2} + \vec{\tau}_{B,3} \tag{2.8}$$

Where $\vec{\tau}_{B,i}; i = 1, 2, 3$ is the torque vector of the body, generated by the i-th omni-wheel with size of τ_i and the $\vec{\tau}_{B,j}; j = x, y, z$ is the torque vector of the body, generated by the virtual wheel in the direction of the j-th axis.

In the next figures, $\vec{F}_{B,i}$ is a force generated by the torque of the i-th omni-wheel, orthogonal to the $\vec{\tau}_{B,i}$, and $\vec{r}_{B,i}$ is a vector from the center of the plane disk to the beginning of the force. Parameter α denotes the angle for vertical position of the omni-wheels and β the angle for horizontal position of the first omni-wheel. The omni-wheels are placed with equal spacing (120 degrees) between them.

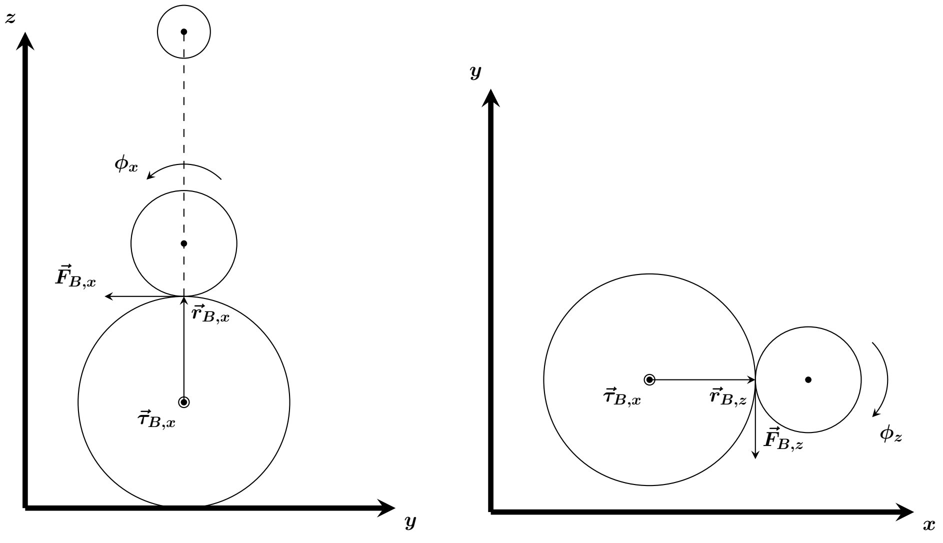


(a) : Side view

(b) : Top view

Figure 2.3: Sketches of torques generated by the real drive system - inspired by [11]

For the virtual case, $\vec{F}_{B,j}$ is defined as a force generated by the torque of the j-th virtual wheel, orthogonal to the $\vec{\tau}_{B,j}$ and also having $\vec{r}_{B,j}$ as a vector from the centre of the plane disk to the beginning of the force.



(a) : Side view

(b) : Top view

Figure 2.4: Sketches of torques generated by the virtual drive system - inspired by [11]

With upper definitions, the torque vectors can be calculated using the following relationships:

$$\vec{\tau}_{B,i} = \vec{r}_{B_i} \times \vec{F}_{B_i}; \qquad i = 1, 2, 3 \tag{2.9}$$

$$\vec{\tau}_{B,j} = \vec{r}_{B_j} \times \vec{F}_{B_j}; \qquad i = x, y, z \tag{2.10}$$

Note that the following equations are taken from [11].

2.4.1 Torques generated by the real drive system

The vectors in the real drive system can be derived from the omni-wheels torques as shown in Fig. 2.3 as:

$$\begin{aligned}
\vec{F}_{B,1} &= \frac{\tau_1}{r_W} \begin{bmatrix} \sin(\beta) \\ -\cos(\beta) \\ 0 \end{bmatrix} \\
\vec{F}_{B,2} &= \frac{\tau_2}{r_W} \begin{bmatrix} \sin(\beta + \frac{2\pi}{3}) \\ -\cos(\beta + \frac{2\pi}{3}) \\ 0 \end{bmatrix} \\
\vec{F}_{B,3} &= \frac{\tau_3}{r_W} \begin{bmatrix} \sin(\beta - \frac{2\pi}{3}) \\ -\cos(\beta - \frac{2\pi}{3}) \\ 0 \end{bmatrix}
\end{aligned} \tag{2.11}$$

$$\begin{aligned}
\vec{r}_{B,1} &= r_S \begin{bmatrix} \sin(\alpha)\cos(\beta) \\ \sin(\alpha)\sin(\beta) \\ \cos(\alpha) \end{bmatrix} \\
\vec{r}_{B,2} &= r_S \begin{bmatrix} \sin(\alpha)\cos(\beta + \frac{2\pi}{3}) \\ \sin(\alpha)\sin(\beta + \frac{2\pi}{3}) \\ \cos(\alpha) \end{bmatrix} \\
\vec{r}_{B,3} &= r_S \begin{bmatrix} \sin(\alpha)\cos(\beta - \frac{2\pi}{3}) \\ \sin(\alpha)\sin(\beta - \frac{2\pi}{3}) \\ \cos(\alpha) \end{bmatrix}
\end{aligned} \tag{2.12}$$

After substituting Eq. (2.11) and (2.12) into Eq. (2.9), the torque vectors of the body generated by the real drive system can be obtained as:

$$\begin{aligned}
\vec{\tau}_{B,1} &= \begin{bmatrix} \frac{r_S \cos(\alpha)\tau_1 \cos(\beta}{r_W} \\ \frac{r_S \cos(\alpha)\tau_1 \sin(\beta}{r_W} \\ -\frac{r_S \sin(\alpha)\tau_1}{r_W} \end{bmatrix} \\
\vec{r}_{B,2} &= \begin{bmatrix} \frac{r_S \cos(\alpha)\tau_2 \cos(\beta + \frac{2\pi}{3})}{r_W} \\ \frac{r_S \cos(\alpha)\tau_2 \sin(\beta + \frac{2\pi}{3})}{r_W} \\ -\frac{r_S \sin(\alpha)\tau_2}{r_W} \end{bmatrix} \\
\vec{r}_{B,3} &= \begin{bmatrix} -\frac{r_S \cos(\alpha)\tau_3 \cos(\beta - \frac{2\pi}{3})}{r_W} \\ \frac{r_S \cos(\alpha)\tau_3 \sin(\beta - \frac{2\pi}{3})}{r_W} \\ -\frac{r_S \sin(\alpha)\tau_3}{r_W} \end{bmatrix}
\end{aligned} \tag{2.13}$$

2.4.2 Torques generated by the virtual drive system

In the virtual drive system (shown in Fig. 2.4), the vectors can also be derived from wheels torques, in this case from the torques of the virtual wheels as:

$$\begin{aligned}\vec{F}_{B,x} &= \frac{\tau_x}{r_W}\begin{bmatrix}0\\-1\\0\end{bmatrix}\\ \vec{F}_{B,y} &= \frac{\tau_y}{r_W}\begin{bmatrix}1\\0\\0\end{bmatrix}\\ \vec{F}_{B,z} &= \frac{\tau_z}{r_W}\begin{bmatrix}\sin(\beta)\\-\cos(\beta)\\0\end{bmatrix}\end{aligned} \tag{2.14}$$

$$\begin{aligned}\vec{r}_{B,x} &= r_S\begin{bmatrix}0\\0\\1\end{bmatrix}\\ \vec{r}_{B,y} &= r_S\begin{bmatrix}0\\0\\1\end{bmatrix}\\ \vec{r}_{B,z} &= r_S\begin{bmatrix}\cos(\beta)\\\sin(\beta)\\0\end{bmatrix}\end{aligned} \tag{2.15}$$

Now substituting Eq (2.14) and (2.15) into Eq. (2.10), the torque vectors of the body, generated by the virtual drive system, can be worked out as:

$$\begin{aligned}\vec{\tau}_{B,x} &= \begin{bmatrix}\frac{r_S\tau_x}{r_W}\\0\\0\end{bmatrix}\\ \vec{\tau}_{B,y} &= \begin{bmatrix}0\\\frac{r_S\tau_y}{r_W}\\0\end{bmatrix}\\ \vec{\tau}_{B,z} &= \begin{bmatrix}0\\0\\-\frac{r_S\tau_z}{r_W}\end{bmatrix}\end{aligned} \tag{2.16}$$

2.4.3 Conclusion

Substituing previously derived equations of torque vectors into Eq. (2.8) and solving for the torques generated by the real drive system and the virtual drive system separately yields two equations. These equations describe the relationship between the two drive systems and serves for conversion between them.

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \mathbf{J} \cdot \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \tag{2.17}$$

$$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \mathbf{J}^T \cdot \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{2.18}$$

Where $\mathbf{J}$ is a Jacobian matrix and $\mathbf{J}^T$ its transposition:

$$\mathbf{J} = \begin{bmatrix} \frac{2\cos(\beta)}{3\cos(\alpha)} & \frac{2\sin(\beta)}{3\cos(\alpha)} & \frac{1}{3\sin(\alpha)} \\ -\frac{\cos(\beta)+\sqrt{3}\sin(\beta)}{3\cos(\alpha)} & -\frac{\sin(\beta)+\sqrt{3}\cos(\beta)}{3\cos(\alpha)} & \frac{1}{3\sin(\alpha)} \\ -\frac{\cos(\beta)+\sqrt{3}\sin(\beta)}{3\cos(\alpha)} & -\frac{\sin(\beta)+\sqrt{3}\cos(\beta)}{3\cos(\alpha)} & \frac{1}{3\sin(\alpha)} \end{bmatrix} \tag{2.19}$$

$$\mathbf{J}^T = \begin{bmatrix} \cos(\alpha)\cos(\beta) & -\frac{\cos(\alpha)[\cos(\beta)+\sqrt{3}\sin(\beta)]}{2} & -\frac{\cos(\alpha)[\cos(\beta)-\sqrt{3}\sin(\beta)]}{2} \\ \cos(\alpha)\sin(\beta) & -\frac{\cos(\alpha)[\sin(\beta)-\sqrt{3}\cos(\beta)]}{2} & -\frac{\cos(\alpha)[\sin(\beta)+\sqrt{3}\cos(\beta)]}{2} \\ \sin(\alpha) & \sin(\alpha) & \sin(\alpha) \end{bmatrix} \tag{2.20}$$

2.5 Equations of motion for the y-z/x-z planes

Equations of motion will be obtained using Lagrangian mechanics. Hence minimal coordinates, kinetic and potential energy, and external torques are needed. The minimal coordinates were already derived in Subsec. 2.3.1. The rest will be derived below.

2.5.1 Kinetic and potential energy

The basic formula for kinetic energy is defined as the sum of translational and rotational energy:

$$T = \frac{1}{2}mv^2 + \frac{1}{2}I\dot{\theta}^2 \tag{2.21}$$

For potential energy:

$$V = mgh \tag{2.22}$$

Using these two equations and references [9] and [11] the next subsections were created.

Energy of the ball

For the ball, Eq. (2.21) becomes:

$$T_{S,yz} = \frac{1}{2}m_S\vec{v}_{S,yz}^T \cdot \vec{v}_{S,yz} + \frac{1}{2}I_S{\dot{\theta}_x}^2 \tag{2.23}$$

m_S is defined as the mass of the ball, $\vec{v}_{S,yz}$ as the velocity of the ball, I_S as already mentioned is the moment of inertia of the ball, and $\vec{v}_{S,yz}^T \cdot \vec{v}_{S,yz}$ is defined as:

$$\begin{aligned} \vec{v}_{S,yz}^T \cdot \vec{v}_{S,yz} &= |v_{S,yz}|^2 \\ &= \dot{y}_S^2 + \dot{z}_S^2 \\ &= r_S^2 \dot{\theta}_x^2 \end{aligned} \tag{2.24}$$

Hence the final form of the kinetic energy of the ball is:

$$T_{S,yz} = \frac{1}{2} m_S r_S^2 \dot{\theta}_x^2 + \frac{1}{2} I_S {\dot{\theta}_x}^2 \tag{2.25}$$

Due to Assumption 3, the potential energy of the ball is equal to zero:

$$V_{S,yz} = 0 \tag{2.26}$$

Energy of the body

Similarly, the kinetic energy of the body is defined as:

$$T_{B,yz} = \frac{1}{2} m_B \vec{v}_{B,yz}^T \cdot \vec{v}_{B,yz} + \frac{1}{2} I_B {\dot{\psi}_x}^2 \tag{2.27}$$

Where m_B is the mass of the body, $\vec{v}_{B,yz}$ the velocity of the body, I_B the moment of inertia of the body mentioned earlier and $\vec{v}_{B,yz}^T \cdot \vec{v}_{B,yz}$ is defined as:

$$\begin{aligned} \vec{v}_{B,yz}^T \cdot \vec{v}_{B,yz} &= |v_{B,yz}|^2 \\ &= \dot{y}_B^2 + \dot{z}_B^2 \\ &= r_S^2 \dot{\theta}_x^2 + 2 r_S l \dot{\theta}_x \dot{\psi}_x \cos(\psi_x) + l^2 \dot{\psi}_x^2 \end{aligned} \tag{2.28}$$

(A more detailed calculation can be found in Appendix A.)

The equation in its final form is:

$$\begin{aligned} T_{B,yz} &= \frac{1}{2} m_B \left(r_S^2 \dot{\theta}_x^2 + 2 r_S l \dot{\theta}_x \dot{\psi}_x \cos(\psi_x) + l^2 \dot{\psi}_x^2 \right) + \frac{1}{2} I_B {\dot{\psi}_x}^2 \\ &= \frac{1}{2} m_B \left(r_S^2 \dot{\theta}_x^2 + 2 r_S l \dot{\theta}_x \dot{\psi}_x \cos(\psi_x) \right) + \frac{1}{2} (I_B + m_B + l^2) {\dot{\psi}_x}^2 \\ &= \frac{1}{2} m_B \left(r_S^2 \dot{\theta}_x^2 + 2 r_S l \dot{\theta}_x \dot{\psi}_x \cos(\psi_x) \right) + \frac{1}{2} I'_B {\dot{\psi}_x}^2 \end{aligned} \tag{2.29}$$

Where I'_B is the moment of inertia of the body around the axis the body rotates (axis through the centre of the wheel).

The potential energy of the body is calculated as:

$$V_{B,yz} = m_B g l \cos(\psi_z) \tag{2.30}$$

Energy of the virtual wheel

The kinetic energy in this case is defined as:

$$T_{W,yz} = \frac{1}{2} m_W \vec{v}_{W,yz}^T \cdot \vec{v}_{W,yz} + \frac{1}{2} I_W \dot{\phi_x}^2 \tag{2.31}$$

Where m_W is the mass of the virtual wheel, $\vec{v}_{W,yz}$ the velocity of the virtual wheel, I_W the moment of inertia of the virtual wheel, and $\vec{v}_{W,yz}^T \cdot \vec{v}_{W,yz}$ is defined as:

$$\begin{aligned} \vec{v}_{W,yz}^T \cdot \vec{v}_{W,yz} &= |v_{W,yz}|^2 \\ &= \dot{y}_W^2 + \dot{z}_W^2 \\ &= r_S^2 \dot{\theta}_x^2 + 2r_S(r_S + r_W)\dot{\theta}_x \dot{\psi}_x \cos(\psi_x) + (r_S + r_W)^2 \dot{\psi}_x^2 \end{aligned} \tag{2.32}$$

And where:

$$\dot{\phi}_x = \frac{r_S}{r_W}(\dot{\theta}_x - \dot{\psi}_z) \tag{2.33}$$

(Appendix A contains the in-depth calculation.)

Therefore the kinetic energy of the virtual wheel is:

$$\begin{aligned} T_{W,yz} = &\frac{1}{2} m_W \left(r_S^2 \dot{\theta}_x^2 + 2r_S(r_S + r_W)\dot{\theta}_x \dot{\psi}_x \cos(\psi_x) + (r_S + r_W)^2 \dot{\psi}_x^2 \right) \\ &+ \frac{1}{2} I_W \left(\frac{r_S}{r_W}(\dot{\theta}_x - \dot{\psi}_z) \right)^2 \end{aligned} \tag{2.34}$$

The potential of the virtual wheel is calculated as:

$$V_{W,yz} = m_W g (r_S + r_W) \cos(\psi_z) \tag{2.35}$$

2.5.2 External torques

Using Eq. (2.33) from reference [11], the relationship between velocities can be established.

$$\dot{\phi}_x = \mathbf{J} \left[\dot{\theta}_x \dot{\psi}_z \right] \tag{2.36}$$

Where

$$\mathbf{J} = \begin{bmatrix} \frac{r_S}{r_W} & -\frac{r_S}{r_W} \end{bmatrix}$$

is the Jacobian matrix which maps joint velocities to the virtual wheel velocity.

Torque of the virtual wheel is only applied torque. Because it has a single component in x direction, τ_x is the input and the external torques are:

$$\begin{aligned} \vec{\tau}_{ext} &= \begin{bmatrix} \tau_{\theta_x} \\ \tau_{\psi_x} \end{bmatrix} \\ &= \mathbf{J}^T \tau_x \\ &= \begin{bmatrix} \frac{r_S}{r_W} \\ -\frac{r_S}{r_W} \end{bmatrix} \tau_x \end{aligned} \tag{2.37}$$

2.5.3 Lagrangian mechanics

The Lagrangian is defined as a function of minimal coordinates and their derivations:

$$L(\theta_x, \psi_x, \dot{\theta}_x, \dot{\psi}_x) = T_{S,yz} + T_{B,yz} + T_{W,yz} - V_{S,yz} - V_{B,yz} - V_{W,yz} \tag{2.38}$$

Euler-Lagrange equations distribute the Lagrangian to external torques as:

$$\begin{aligned} \frac{d}{dt}\Big(\frac{\partial L}{\partial \dot{\theta}_x}\Big) - \frac{\partial L}{\partial \theta_x} &= \tau_{\theta_x} \\ \frac{d}{dt}\Big(\frac{\partial L}{\partial \dot{\psi}_x}\Big) - \frac{\partial L}{\partial \psi_x} &= \tau_{\psi_x} \end{aligned} \tag{2.39}$$

This can be rewritten into a matrix form:

$$\mathbf{M}(\vec{q}_{yz})\ddot{\vec{q}}_{yz} + \mathbf{C}(\vec{q}_{yz}, \dot{\vec{q}}_{yz})\dot{\vec{q}}_{yz} + \mathbf{G}(\vec{q}_{yz}) = \vec{\tau}_{ext} \tag{2.40}$$

Where **M** is a matrix of inertial forces due to accelerations, **C** is a matrix of Coriolis and centrifugal forces, and **G** is a matrix of gravitational forces.

After computations in Matlab programming language, these matrices were calculated as:

$$\begin{aligned} \mathbf{M}(\vec{q}_{yz}) &= \begin{bmatrix} I_S + r_S^2 m_{tot} + \frac{r_S^2}{r_W^2} I_W & r_S \lambda \cos(\psi_x) - \frac{r_S^2}{r_W^2} I_W \\ r_S \lambda \cos(\psi_x) - \frac{r_S^2}{r_W^2} I_W & r_{tot}^2 m_W + \frac{r_S^2}{r_W^2} I_W + I'_B \end{bmatrix} \\ \mathbf{C}(\vec{q}_{yz}, \dot{\vec{q}}_{yz}) &= \begin{bmatrix} 0 & -r_S \lambda \dot{\psi}_x \sin(\psi_x) \\ 0 & 0 \end{bmatrix} \\ \mathbf{G}(\vec{q}_{yz}) &= \begin{bmatrix} 0 \\ -\lambda g \sin(\psi_z) \end{bmatrix} \end{aligned}$$

Where:

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

These matrices are similar to the result in [11], therefore the acquisition process is considered as correct.

2.5.4 Linearization

A linear model needs to be created to lessen the difficulty of controlling tasks. This is done by linearizing equations of motion, which were derived in a previous section. Firstly, the state vector needs to be determined as:

$$\vec{x} = \begin{bmatrix} \vec{q}_{yz} \\ \dot{\vec{q}}_{yz} \end{bmatrix} = \begin{bmatrix} \theta_x \\ \psi_x \\ \dot{\theta}_x \\ \dot{\psi}_x \end{bmatrix} \tag{2.41}$$

Secondly, the input needs to be decided upon. This was done as follows:

$$u = \tau_x \tag{2.42}$$

The linearization will be done at the equilibrium point, which is in an unstable position on the top of the ball. At this point, all states and inputs are equal to zero:

$$\begin{aligned} \bar{\vec{x}} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \bar{u} &= 0 \end{aligned} \tag{2.43}$$

A well known state-space form is defined as:

$$\begin{aligned} \dot{\vec{x}} &= \mathbf{A}\vec{x} + \mathbf{C}u \\ \vec{y} &= \mathbf{C}\vec{x} + \mathbf{D}u \end{aligned} \tag{2.44}$$

Where $\mathbf{D}$ is a 4x1 zero matrix and the rest of matrices are obtained by:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\partial\ddot{\theta}_x}{\partial\theta_x} & \frac{\partial\ddot{\theta}_x}{\partial\psi_x} & \frac{\partial\ddot{\theta}_x}{\partial\dot{\theta}_x} & \frac{\partial\ddot{\theta}_x}{\partial\dot{\psi}_x} \\ \frac{\partial\ddot{\psi}_x}{\partial\theta_x} & \frac{\partial\ddot{\psi}_x}{\partial\psi_x} & \frac{\partial\ddot{\psi}_x}{\partial\dot{\theta}_x} & \frac{\partial\ddot{\psi}_x}{\partial\dot{\psi}_x} \end{bmatrix}_{(\vec{x},u)=(\bar{\vec{x}},\bar{u})} \\ \mathbf{B} &= \begin{bmatrix} 0 \\ 0 \\ \frac{\partial\ddot{\theta}_x}{\partial u} \\ \frac{\partial\ddot{\psi}_x}{\partial u} \end{bmatrix}_{(\vec{x},u)=(\bar{\vec{x}},\bar{u})} \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2.45}$$

As shown in [11], after substituting the equilibrium point (2.43) into Eq. (2.40) and partially deriving it by each coordinate, the following reduction can be made:

$$\begin{aligned} \left(\frac{\partial\mathbf{M}(\vec{q}_{yz})}{\partial x_i}\ddot{\vec{q}}_{yz} + \mathbf{M}(\vec{q}_{yz})\frac{\partial \ddot{\vec{q}}_{yz}}{x_i} + \frac{\partial\mathbf{C}(\vec{q}_{yz},\dot{\vec{q}}_{yz})}{\partial x_i}\dot{\vec{q}}_{yz} + \frac{\partial\mathbf{G}(\vec{q}_{yz})}{\partial x_i}\right)_{(\vec{x},u)=(\bar{\vec{x}},\bar{u})} &= 0 \\ \left(\mathbf{M}(\bar{\vec{q}}_{yz})\frac{\partial \ddot{\vec{q}}_{yz}}{x_i} + \frac{\partial\mathbf{G}(\vec{q}_{yz})}{\partial x_i}\right)_{(\vec{x},u)=(\bar{\vec{x}},\bar{u})} &= 0 \end{aligned} \tag{2.46}$$

Where i denotes the i-th element of the state vector $\vec{x}$.

Therefore:

$$\frac{\partial \ddot{\vec{q}}_{yz}}{x_i} = -\mathbf{M}^{-1}(\bar{\vec{q}}_{yz})\frac{\partial\mathbf{G}(\vec{q}_{yz})}{\partial x_i}_{(\vec{x},u)=(\bar{\vec{x}},\bar{u})} \tag{2.47}$$

For the input, the reduction has the form of:

$$\begin{aligned}\left(\mathbf{M}(\vec{q}_{yz})\frac{\partial\ddot{\vec{q}_{yz}}}{u}\right)_{(\vec{x},u)=(\bar{\vec{x}},\bar{u})} &= \left(\frac{\partial\vec{\tau}_{ext}}{\partial u}\right)_{(\vec{x},u)=(\bar{\vec{x}},\bar{u})}\\ \mathbf{M}(\bar{\vec{q}}_{yz})\frac{\partial\ddot{\vec{q}_{yz}}}{u} &= \left(\frac{\partial\vec{\tau}_{ext}}{\partial u}\right)_{(\vec{x},u)=(\bar{\vec{x}},\bar{u})}\end{aligned} \tag{2.48}$$

And therefore:

$$\frac{\partial\ddot{\vec{q}_{yz}}}{u} = \mathbf{M}^{-1}(\bar{\vec{q}}_{yz})\left(\frac{\partial\vec{\tau}_{ext}}{\partial u}\right)_{(\vec{x},u)=(\bar{\vec{x}},\bar{u})} \tag{2.49}$$

Using Eq. (2.47) and Eq. (2.49), the elements of matrices **A** and **B** can be calculated. After substituing the remaining uncalculated parameters, the final state-space form is:

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & -23.1180 & 0 & 0\\ 0 & 48.9145 & 0 & 0 \end{bmatrix}\vec{x} + \begin{bmatrix} 0\\ 0\\ 136.9568\\ -127.3498 \end{bmatrix} u \tag{2.50}$$

$$\vec{y} = \begin{bmatrix} 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}\vec{x} + \begin{bmatrix} 0\\ 0\\ 0\\ 0 \end{bmatrix} u \tag{2.51}$$

The controllability and observability of this system can be evaulated using equations:

$$con = rank\begin{bmatrix}\mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \mathbf{A}^3\mathbf{B}\end{bmatrix} \overset{?}{=} 4 \tag{2.52}$$

$$obs = rank\begin{bmatrix}\mathbf{C} & \mathbf{CA} & \mathbf{CA}^2 & \mathbf{CA}^3\end{bmatrix} \overset{?}{=} 4 \tag{2.53}$$

Since the conditions of both equations have been met, the system is fully controllable and observable.

2.6 Equations of motion for the x-y plane

2.6.1 Kinetic and potential energy

Energy of the ball

Due to Assumption 5, the kinetic energy of the ball is equal to zero:

$$T_{S,xy} = 0 \tag{2.54}$$

Energy of the body

Because the body only rotates, the kinetic energy will consist purely of rotational energy:

$$T_{B,xz} = \frac{1}{2} I_{B,xy} \dot{\psi}_x^2 \tag{2.55}$$

Energy of the virtual wheel

The kinetic energy in this case is defined as:

$$T_{W,yz} = \frac{1}{2} m_W \vec{v}_{W,xy}^T \cdot \vec{v}_{W,xy} + \frac{1}{2} I_{W,xy} \dot{\phi_z}^2 \tag{2.56}$$

Where:

$$\begin{aligned} \vec{v}_{W,xy}^T \cdot \vec{v}_{W,xy} &= |v_{W,xy}|^2 \\ &= \dot{x}_{W,xy}^2 + \dot{y}_{W,xy}^2 \\ &= (r_S + r_W)^2 \dot{\psi}_z^2 \end{aligned} \tag{2.57}$$

And where:

$$\dot{\phi}_z = -\frac{r_S}{r_W} \dot{\psi}_z \tag{2.58}$$

(The entire calculation can be found in Appendix A.)

Therefore the kinetic energy of the virtual wheel is:

$$\begin{aligned} T_{W,xy} &= \frac{1}{2} m_W (r_S + r_W)^2 \dot{\psi}_z^2 + \frac{1}{2} I_{W,xy} \Big(-\frac{r_S}{r_W} \dot{\psi}_z \Big)^2 \\ &= \frac{1}{2} \left(m_W (r_S + r_W)^2 + I_{W,xy} \Big(\frac{r_S}{r_W} \Big)^2 \right) \dot{\psi}_z^2 \end{aligned} \tag{2.59}$$

The potential energy in all cases is equal to zero.

2.6.2 External torques

The only input in the x-y plane is τ_z. From Eq. (2.58), the Jacobian is already known. Hence, the relationship between external torques and τ_z can be rewritten as:

$$\begin{aligned} \tau_{ext,xy} &= J^T \tau_z \\ \tau_{ext,xy} &= -\frac{r_S}{r_W} \tau_z \end{aligned} \tag{2.60}$$

2.6.3 Lagrangian mechanics

The Lagrangian in this case is:

$$L(\psi_z, \dot{\psi}_z) = T_{B,yz} + T_{W,yz} \tag{2.61}$$

Euler-Lagrange equation distributes the Lagragian to external torques as:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\psi}_z}\right) - \frac{\partial L}{\partial \psi_z} = \tau_{ext,xy} \tag{2.62}$$

In the matrix form:

$$\mathbf{M}(\vec{q}_{yz})\ddot{\vec{q}}_{yz} = \tau_{ext,xy} \tag{2.63}$$

Where:

$$\mathbf{M}(\vec{q}_{yz}) = I_{B,xy} + m_W(r_S + r_W)^2 + I_{W,xy}\left(\frac{r_S}{r_W}\right)^2 \tag{2.64}$$

2.6.4 Linearization

The state vector is defined as:

$$\vec{x} = \begin{bmatrix} \psi_z \\ \dot{\psi}_z \end{bmatrix} \tag{2.65}$$

With the input:

$$u = \tau_z \tag{2.66}$$

Solving Eq. (2.63) for $\ddot{\psi}_z$ yields:

$$\ddot{\psi}_z = -\frac{r_S u}{I_{B,xy}r_W^2 + m_W(r_S + r_W)^2 r_W^2 + I_{W,xy}r_S^2} \tag{2.67}$$

After deriving and substituing, the state-space form is:

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\vec{x} + \begin{bmatrix} 0 \\ -28006.1898 \end{bmatrix}u \tag{2.68}$$

$$\vec{y} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\vec{x} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}u \tag{2.69}$$

Also in this case, the controllability and observability was checked. The system is fully controllable and observable.

2.7 Calculation of parameters

In previous sections, many of the parameters were listed. This section aims to calculate them exactly or with an acceptable error.

2.7.1 Moment of inertia of the ball

The moment of inertia of the full ball with mass m and radius r is defined as:

$$I = \frac{2}{5}mr^2 \tag{2.70}$$

Thus the moment of inertia of the used ball is:

$$I_S = \frac{2}{5}m_S r_S^2 \tag{2.71}$$

2.7.2 Moment of inertia of the body

Because the body can not be exactly described geometrically, an approximation is made. The body is indentified as a cuboid with mass m_B, width w_B, height h_B, and COM as height r_B. The computation of the moment of inertia of the cuboid rotation around the axis through its COM is:

$$I = \frac{1}{12} m(w^2 + h^2) \tag{2.72}$$

Therefore the moment of inertia in the y-z/x-z planes is defined as:

$$I_B = \frac{1}{12} m_B(w_B^2 + h_B^2) \tag{2.73}$$

For the calculation of the moment of inertia about the axis the body rotates around, as in [11] the parallel axis theorem [13] is used. It states that:

$$I' = I + ml^2 \tag{2.74}$$

Where l is in our case defined as the height difference between the COM of the ball and the COM of the body. Hence:

$$l = r_S + r_B \tag{2.75}$$

And the moment:

$$I'_B = I_B + m_B l^2 \tag{2.76}$$

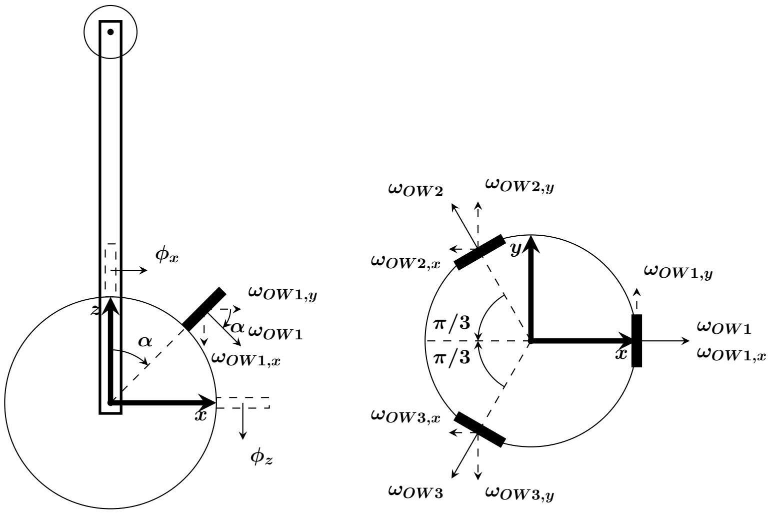
In the x-y plane, the moment of inertia of the body is defined as:

$$I = \frac{1}{6} mw^2 \tag{2.77}$$

Therefore:

$$I_{B,xy} = \frac{1}{6} m_B w_B^2 \tag{2.78}$$

2.7.3 Moment of inertia of the virtual wheel



(a) : Side view

(b) : Top view

Figure 2.5: Sketches of the angular rates of the omni-wheels - inspired by [11]

As shown in Fig. 2.5, the angular rates of the virtual omni-wheels are split into individual components in the direction of the axis. This task is done according to [6], [9], [10], and [11].

The angular rates around the x axis are:

$$\begin{aligned}\omega_{OW1,x} &= \dot{\phi}_x\cos(\alpha)\\ \omega_{OW2,x} = \omega_{OW3,x} &= \cos\left(\frac{\pi}{3}\right)(-\dot{\phi}_x)\cos(\alpha)\\ &= -\frac{1}{2}\dot{\phi}_x\cos(\alpha)\end{aligned}\tag{2.79}$$

Around the y axis they are:

$$\begin{aligned}\omega_{OW1,y} &= 0\\ \omega_{OW2,y} &= \sin\left(\frac{\pi}{3}\right)\dot{\phi}_y\cos(\alpha)\\ &= \frac{\sqrt{3}}{2}\dot{\phi}_y\cos(\alpha)\\ \omega_{OW3,y} &= \sin\left(\frac{\pi}{3}\right)(-\dot{\phi}_y)\cos(\alpha)\\ &= -\frac{\sqrt{3}}{2}\dot{\phi}_y\cos(\alpha)\end{aligned}\tag{2.80}$$

And finally around the z axis they are:

$$\begin{aligned}\omega_{OW1,z} = \omega_{OW2,z} = \omega_{OW3,z} &= \cos\left(\frac{\pi}{2}-\alpha\right)\dot{\phi}_z\\ &= \sin(\alpha)\dot{\phi}_z\end{aligned}\tag{2.81}$$

The moment of inertia of the solid cylinder is defined as:

$$I = \frac{1}{2}mr^2 \tag{2.82}$$

Thus the moment of inertia of the omni-wheel is calculated as:

$$I_{OW} = \frac{1}{2}m_{OW}r_{OW}^2 \tag{2.83}$$

For decomposition, calculations are needed (fully listed in Appendix A). The result is:

$$\begin{gathered} I_W = I_{W,x} = I_{W,y} = \frac{3}{2}\cos(\alpha)^2(I_{OW} + k^2 I_M) \\ I_{W,xy} = I_{W,z} = 3\sin(\alpha)^2(I_{OW} + k^2 I_M) \end{gathered} \tag{2.84}$$

Where k is gearbox reduction and I_M is the inertia of the rotor in the real motor.

2.7.4 Mass of the virtual wheel

The mass of the virtual wheel consists of three parts: the mass of the motor, the mass of the shaft, and the mass of the omni-wheel.

$$m_W = m_M + m_{Sh} + m_{OW} \tag{2.85}$$

2.7.5 Angle α

As shown in Fig. 2.5, the angle α denotes the position of the omni-wheels on the ball. This angle determines the geometrical transcription between the angular velocities of the omni-wheels and the angular velocity of the ball. A lot of research was done (especially in [6] and [11]) to estimate the best angle for driving the ball. When $\alpha = 90°$, the rotation around the x and y axes is impossible, while rotation around the z axis is elementary. The opposite is the case with $\alpha = 0°$, the rotation around the x and y axes should be simple and rotation around the z axis should be impossible. However, due to missing a supporting triangle, the robot falls down. Hence an angle of $\alpha = 45°$ is chosen, which gives the robot the ability to balance and rotate around the x, y, and z axes.

2.7.6 Overview of parameters

For clarity, in the following table, all used parameters are listed.

Parameter	Description	Value
r_S	Radius of the ball	0.108 [m]
w_B	Width of the body	0.15 [m]
h_B	Height of the body	0.24 [m]
r_B	Half of the height of the body	0.12 [m]
r_{OW}	Radius of the omni-wheel	0.024 [m]
r_W	Radius of the virtual wheel	0.024 [m]
l	Distance between ball's COM and body's COM	0.228 [m]
m_S	Mass of the ball	2.722 [kg]
m_B	Mass of the body	1.07 [kg]
m_M	Mass of the motor	0.076 [kg]
m_{Sh}	Mass of the shaft	0.0034 [kg]
m_{OW}	Mass of the omni-wheel	0.038 [kg]
m_W	Mass of the virtual wheel	0.1174 [kg]
α	Angle of the omni-wheels	45 [deg]
I_S	Moment of inertia of the ball	0.0127 $[kgm^2]$
I_B	Moment of inertia of the body in y-z/x-z	0.0071 $[kgm^2]$
I'_B	Moment of inertia of the body about rotation axis	0.0628 $[kgm^2]$
$I_{B,xy}$	Moment of inertia of the body in x-y	0.004 $[kgm^2]$
I_M	Moment of inertia of the motor	$1 \cdot 10^{-5}$ $[kgm^2]$
I_{OW}	Moment of inertia of the omni-wheel	$1.0944 \cdot 10^{-5}$ $[kgm^2]$
I_W	Moment of inertia of the virtual wheel in y-z/x-z	$1.5708 \cdot 10-5$ $[kgm^2]$
$I_{W,xy}$	Moment of inertia of the virtual wheel in x-y	$3.1416 \cdot 10^{-5}$ $[kgm^2]$
g	Gravitational acceleration	9.81 $[m/s^2]$
k	Reduction of the gearbox	1 [-]

Table 2.1: Table of 2D parameters

2.8 Simulations

With computed parameters, the simulations of the physical and linearized models can be made. Simulations are carried out to see whether the linearized model is a good replacement for the physical model, as well as to see whether there are any deviations and determine their size. Because the model is identical in y-z and x-z planes, only y-z plane with x-y plane are simulated. In the first simulation, the initial condition of the angle ψ_x was set 0.1°. Below the courses for angles are shown:

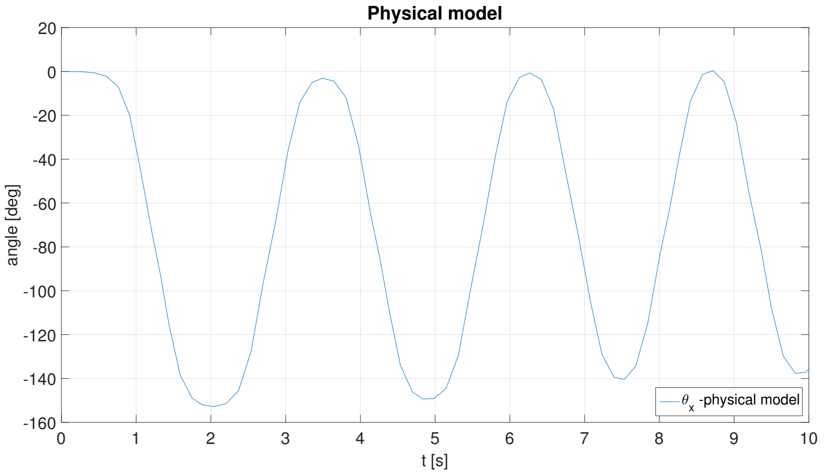


Figure 2.6: Physical model - course of θ_x - $\psi_x(t = 0) = 0.1°$

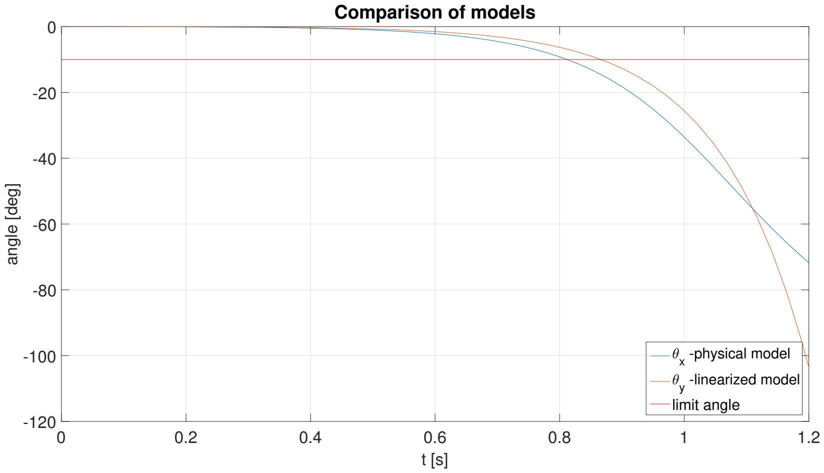


Figure 2.7: Comparison for θ_x - $\psi_x(t = 0) = 0.1°$

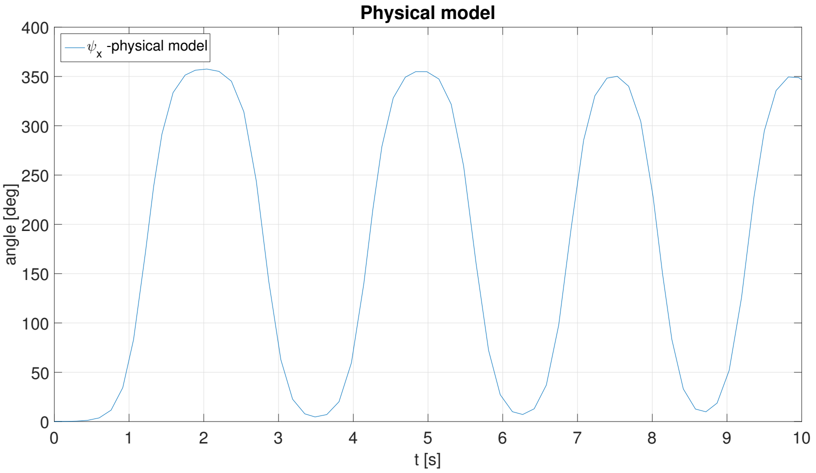


Figure 2.8: Physical model - course of ψ_x - $\psi_x(t = 0) = 0.1°$

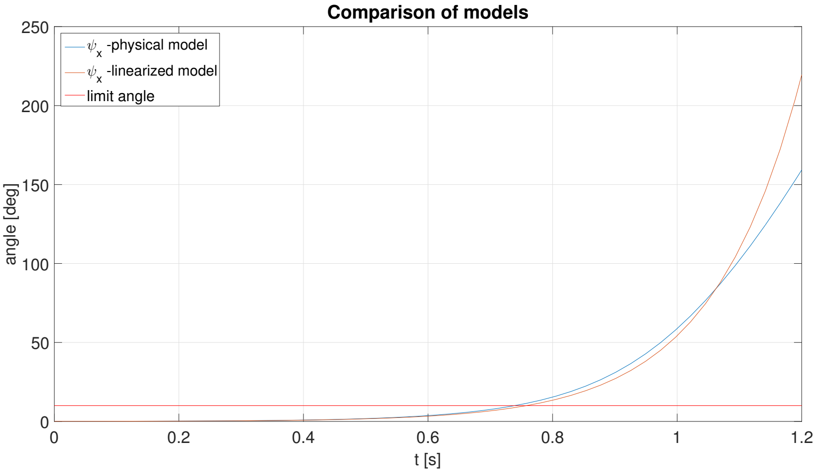


Figure 2.9: Comparison for ψ_x - $\psi_x(t = 0) = 0.1°$

In Fig. 2.6 and 2.8 oscillations can be seen. The robot falls through the floor and starts to oscillate around the centre of the ball. This is due to a lack of floor definition. In Fig. 2.7 and 2.9 a limit angle of 10° is observed (see Subsec. 4.3.2 for more information on the limit angle).

In the second simulation, the input of each system was changed to 0.01N/m. As before, the angles θ_x and ψ_x were inspected for both systems.

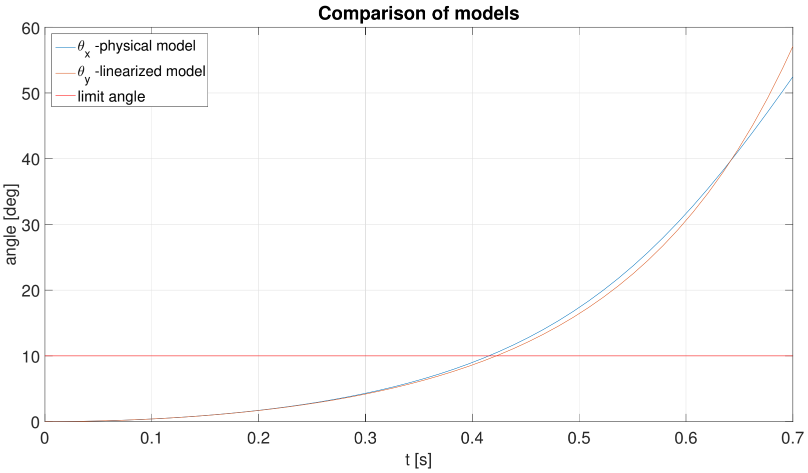


Figure 2.10: Comparison for θ_x - $u = 0.01\text{N/m}$

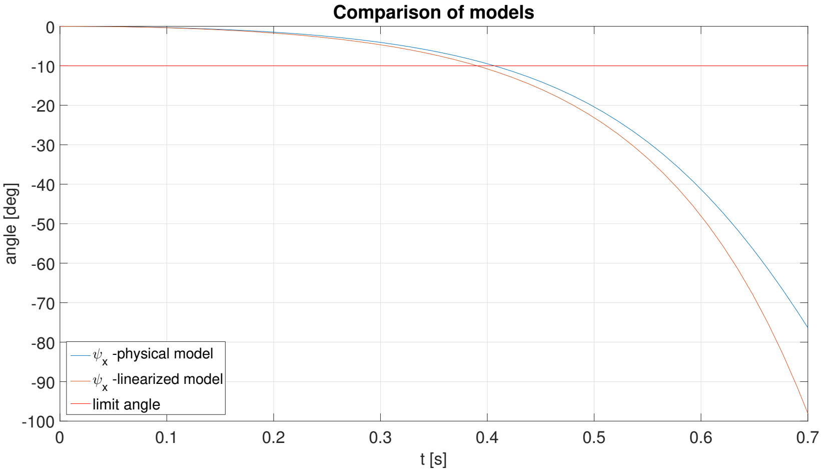


Figure 2.11: Comparison for ψ_x - $u = 0.01\text{N/m}$

Thirdly, the x-y plane was observed with an input of 0.001N/m.

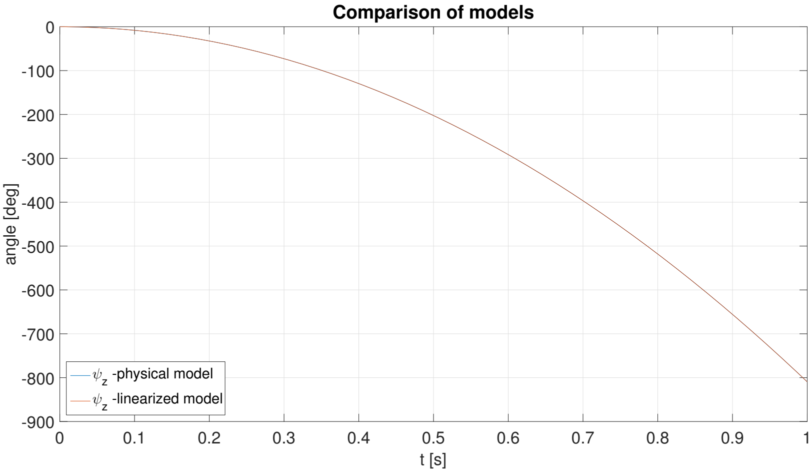


Figure 2.12: Comparison for ψ_z - $u = 0.001\text{N/m}$

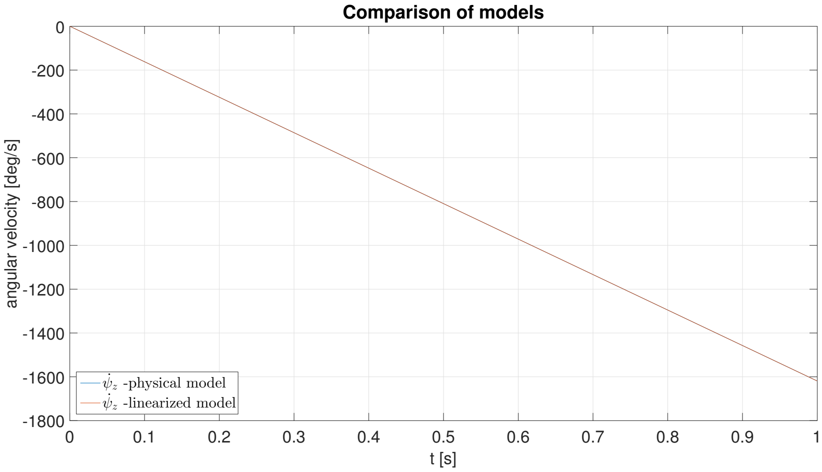


Figure 2.13: Comparison for $\dot{\psi}_z$ - $u = 0.001\text{N/m}$

From these observations a conclusion can be made. The linearized model is a satisfactory replacement for the physical model within the controlling limits.

Chapter 3

3D model

This chapter aims to describe the structure of the system and physical model of the BRB. As before, the dynamic model with equations of motion is derived using Lagrangian mechanics and converted into a linearized model. While linearizing, an equilibrium point in a balancing position is used. However, the 3D model cancels the assumption that the vertical planes are independent. Therefore an approach is difficult and the calculations quite demanding.

3.1 Assumptions

The assumptions for the 3D model are identical to those for the 2D model except the assumption of independent vertical planes. Shortly:

- **Rigid bodies**
- **Rigid floor**
- **Horizontal floor**
- **Friction**
- **No slip**
- **Omni-wheels**
- **Negligible time delay**

Detailed information can be found in Sec. 2.1.

3.2 Model description

As with the 2D model, the 3D model consists of two rigid bodies - the ball and the body. The body is modeled as a solid cuboid and the omni-wheels as solid disks. The COM of the body (together with omni-wheels) is estimated experimentally.

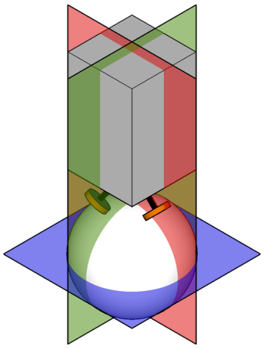


Figure 3.1: 3D illustration of the BRB

3.3 Coordinates

It is also necessary to define the coordinates for the 3D model. The coordinates will be based on the DOFs of the system. To calculate DOFs, a closer look at the system is required. Due to the no slip assumption, only the translation of the ball is relevant. Therefore two DOFs are obtained. For the body, orientations are also observed, which yield three more DOFs.

3.3.1 Minimal coordinates

The minimal coordinates are therefore defined as:

$$\vec{q} = \begin{bmatrix} x_S & y_S & \psi_x & \psi_y & \psi_z \end{bmatrix}^T \tag{3.1}$$

Where x_S and y_S denote the translation of the ball along the x and y axes and ψ_x, ψ_y, and ψ_z indicate the orientation of the body.

3.3.2 Coordinate frames

To describe the position of the ball and the orientation of the body, as in [11], coordinate frames are used. For the orientation of the body, Tait-Bryan

(Yaw-Pitch-Roll) angles, a special case of Euler angles, are used. In total, six coordinate frames are defined:

- **Inertial frame I**
  The basic inertial frame.

- **Coordinate frame 1**
  Translation of the inertial frame with x_S along its x axis.

- **Coordinate frame 2**
  Translation of the coordinate frame 1 with y_S along its y axis. Note that this frame is located in the COM of the ball.

- **Coordinate frame 3**
  Counterclockwise rotation of the coordinate frame 2 with ψ_z around its z axis.

- **Coordinate frame 4**
  Counterclockwise rotation of the coordinate frame 3 with ψ_y around its y axis.

- **Coordinate frame 5**
  Clockwise rotation of the coordinate frame 4 with ψ_x around its x-axis together with translation of value l along its z-axis.

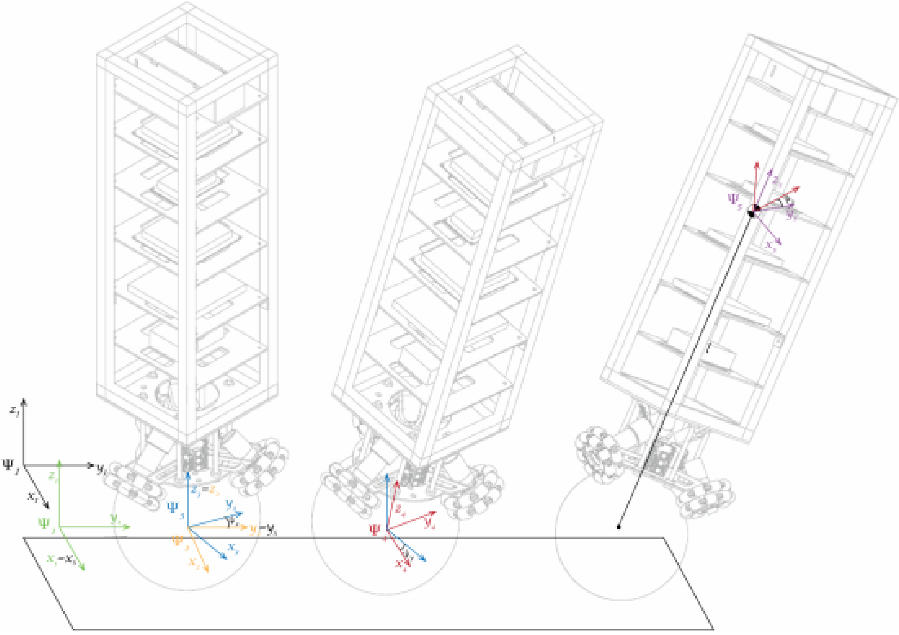


Figure 3.2: Illustration of coordinate frames - taken from [11]

3.3.3 Transformations

In [11], homogeneous matrices are also used to describe transformations between established frames. The homogenous matrix from the frame Ψ_i to the frame Ψ_j is defined as:

$$\mathbf{H}_i^j = \begin{bmatrix} \mathbf{R}_i^j & \vec{o}_i^j \\ \vec{0}_{1\times 3} & 1 \end{bmatrix} \tag{3.2}$$

Where $\mathbf{R}_i^j$ is the rotation matrix from Ψ_i to Ψ_j, $\vec{o}_i^j$ represents the position of origin of Ψ_i expressed in Ψ_j and $\vec{0}_{1\times 3} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$. Therefore:

$$\mathbf{H}_I^1 = \begin{bmatrix} 1 & 0 & 0 & -x_S \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

$$\mathbf{H}_1^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -y_S \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

$$\mathbf{H}_2^3 = \begin{bmatrix} \cos(\psi_z) & \sin(\psi_z) & 0 & 0 \\ -\sin(\psi_z) & \cos(\psi_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

$$\mathbf{H}_3^4 = \begin{bmatrix} \cos(\psi_y) & 0 & \sin(\psi_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\psi_y) & 0 & \cos(\psi_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

$$\mathbf{H}_4^5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\psi_x) & -\sin(\psi_x) & 0 \\ 0 & \sin(\psi_x) & \cos(\psi_x) & -\mathring{\mathrm{u}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

Using these matrices, transformation between any of the frames can be made.

3.4 Variables

The last step in obtaining equations of motion is to derive the variables that will be used. As shown in [11], this is possible with defined coordinate frames and transformations between them. Note that the procedure in this section draws on the mentioned reference. For clarification, an overview of all markings used in this thesis can be found in the following table.

Marking	Description
$\mathbf{H}_i^j$	Homogeneous matrix form Ψ_i to Ψ_j
$\mathbf{R}_i^j$	Rotation matrix form Ψ_i to Ψ_j
$\vec{o}_i^j$	Position of origin of Ψ_i expressed in Ψ_j
$\vec{r}_k^{i,j}$	Position of Ψ_k w.r.t. Ψ_j, expressed in Ψ_i
$\vec{v}_k^{i,j}$	Linear velocity of Ψ_k w.r.t. Ψ_j, expressed in Ψ_i
$\vec{\omega}_k^{i,j}$	Angular velocity of Ψ_k w.r.t. Ψ_j, expressed in Ψ_i
$\vec{T}_k^{i,j}$	Twist of Ψ_k w.r.t. Ψ_j, expressed in Ψ_i
$\mathbf{I}_O^i$	Inertia tensor of an object O around its COM chosen in Ψ_i
$\vec{I}_O$	Moment of inertia of an object O around its COM
$\vec{I}'_O$	Moment of inertia of an object O around its rotation axis
$\mathbf{Ad}_{\mathbf{H}_i^j}$	Adjoint matrix of $\mathbf{H}_i^j$ that maps twist from Ψ_i to Ψ_j

Table 3.1: Table of markings

Where w.r.t. stands for "with relation to".

The twist vector is defined as a conjunction of the angular velocity vector and of the linear velocity vector:

$$\vec{T}_k^{i,j} = \begin{bmatrix} \vec{\omega}_k^{i,j} \\ \vec{v}_k^{i,j} \end{bmatrix} \tag{3.8}$$

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

$$\vec{T}_k^{n,j} = \mathbf{Ad}_{\mathbf{H}_i^n} \cdot \vec{T}_k^{i,j} \tag{3.9}$$

With the adjoint matrix:

$$\mathbf{Ad}_{\mathbf{H}_i^n} = \begin{bmatrix} \mathbf{R}_i^n & \mathbf{0}_{3\times3} \\ \tilde{\mathbf{o}}_i^n \cdot \mathbf{R}_i^n & \mathbf{R}_i^n \end{bmatrix} \tag{3.10}$$

Where $\mathbf{0}_{3\times3}$ is 3x3 zero matrix and:

$$\tilde{\mathbf{o}}_i^n = \begin{bmatrix} 0 & -o_z & o_y \\ o_z & 0 & -o_x \\ -o_y & o_x & 0 \end{bmatrix} \tag{3.11}$$

The elements of this matrix are obtained from:

$$\vec{o}_i^n = \begin{bmatrix} o_x & o_y & o_z \end{bmatrix}^T \tag{3.12}$$

3.4.1 Velocities of the ball

Linear velocity of the ball w.r.t. Ψ_I, expressed in the same frame, is marked as the linear velocity of Ψ_2 w.r.t. Ψ_I, also expressed in the same frame:

$$\vec{v}_2^{I,I} = \begin{bmatrix} \dot{x}_S \\ \dot{y}_S \\ 0 \end{bmatrix} \tag{3.13}$$

Due to the lack of rotation of Ψ_2 w.r.t. Ψ_I, the angular velocity $\vec{\omega}_2^{I,I}$ is equal to zero:

$$\vec{\omega}_2^{I,I} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.14}$$

Note that this angular velocity is not the angular velocity of the ball.
Hence the twist $\vec{T}_2^{I,I}$ w.r.t Ψ_I, expressed in Ψ_I is:

$$\vec{T}_2^{I,I} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{x}_S \\ \dot{y}_S \\ 0 \end{bmatrix} \tag{3.15}$$

Using this twist, the twist Ψ_2 w.r.t. Ψ_I, expressed in Ψ_2 can be introduced as:

$$\vec{T}_2^{2,I} = \mathbf{Ad}_{\mathbf{H}_I^2} \cdot \vec{T}_2^{I,I} \tag{3.16}$$

For later calculations, the twist Ψ_2 w.r.t. Ψ_I, expressed in Ψ_5 is needed. The definition is as follows:

$$\vec{T}_2^{5,I} = \mathbf{Ad}_{\mathbf{H}_I^5} \cdot \vec{T}_2^{I,I} \tag{3.17}$$

Now the angular velocity of the ball remains to be clarified.
With help of the auxiliary vector:

$$\vec{r}_T = \begin{bmatrix} 0 \\ 0 \\ r_S \end{bmatrix} \tag{3.18}$$

And basic formula for the angular velocity:

$$\vec{\omega} = \frac{\vec{r} \times \vec{v}}{|\vec{r}|^2} \tag{3.19}$$

The angular velocity of the ball w.r.t Ψ_I, expressed in Ψ_2 is calculated as:

$$\begin{aligned} \vec{\omega}_S^{2,I} &= \frac{\vec{r}_T \times \vec{v}_2^{2,I}}{|\vec{r}_T|^2} \\ &= \begin{bmatrix} \frac{\dot{y}_S}{r_S} \\ -\frac{\dot{x}_S}{r_S} \\ 0 \end{bmatrix} \end{aligned} \tag{3.20}$$

3.4.2 Velocities of the body

The twist of the body w.r.t. Ψ_I, expressed also in Ψ_I is marked as the twist of the frame Ψ_5 w.r.t. Ψ_I, expressed in Ψ_I. Therefore:

$$\vec{T}_5^{I,I} = \mathbf{J} \cdot \dot{\vec{q}_j} \tag{3.21}$$

Where $\mathbf{J}$ is the Jacobian matrix, which maps the joint velocities to the body of the BRB. Due to Tait-Bryan angles, the velocities are defined as:

$$\dot{\vec{q}_j} = \begin{bmatrix} \dot{x}_S & \dot{y}_S & \dot{\psi}_z & \dot{\psi}_y & \dot{\psi}_x \end{bmatrix}^T \tag{3.22}$$

and the Jacobian as:

$$\begin{aligned} \mathbf{J} &= \begin{bmatrix} \vec{T}_1^{I,I} & \vec{T}_2^{I,1} & \vec{T}_3^{I,2} & \vec{T}_4^{I,3} & \vec{T}_5^{I,4} \end{bmatrix} \\ &= \begin{bmatrix} \vec{T}_1^{I,I} & \mathbf{Ad}_{\mathbf{H}_1^1} \cdot \vec{T}_2^{I,1} & \mathbf{Ad}_{\mathbf{H}_2^I} \cdot \vec{T}_3^{2,2} & \mathbf{Ad}_{\mathbf{H}_3^I} \cdot \vec{T}_4^{3,3} & \mathbf{Ad}_{\mathbf{H}_4^I} \cdot \vec{T}_5^{4,4} \end{bmatrix} \end{aligned} \tag{3.23}$$

The twists for its computation, using Fig. 3.2, are derived as:

$$\begin{aligned} \vec{T}_1^{I,I} &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T \\ \vec{T}_2^{1,1} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T \\ \vec{T}_3^{2,2} &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T \\ \vec{T}_4^{3,3} &= \begin{bmatrix} 0 & 1 & 0 & l & 0 & 0 \end{bmatrix}^T \\ \vec{T}_5^{4,4} &= \begin{bmatrix} -1 & 0 & 0 & 0 & l & 0 \end{bmatrix}^T \end{aligned} \tag{3.24}$$

Where the linear velocity of the twist $\vec{T}_4^{3,3}$ is computed as:

$$\vec{v}_4^{3,3} = \left(\vec{\omega}_4^{3,3} \times \mathbf{R}_4^3 \cdot \vec{l}_T \right)$$

And respectively, the linear velocity of the twist $\vec{T}_5^{4,4}$ as:

$$\vec{v}_5^{4,4} = \left(\vec{\omega}_5^{4,4} \times \mathbf{R}_5^4 \cdot \vec{l}_T \right)$$

Where $\vec{l}_T = \begin{bmatrix} 0 & 0 & l \end{bmatrix}^T$.

Having solved Eq. (3.21), the twist of the body w.r.t. Ψ_I, expressed in Ψ_5 can be calculated:

$$\vec{T}_5^{5,I} = \mathbf{Ad}_{\mathbf{H}_I^5} \cdot \vec{T}_5^{I,I} \tag{3.25}$$

3.4.3 Angular velocities of the omni-wheels

Due to assumption 5, the circumferential velocity of the omni-wheel is identical to the circumferential velocity of the ball in the direction of that omni-wheel. Let us define ϕ_i as the angular velocity of the i-th omni-wheel with a positive counterclockwise rotation. Therefore, with Eq. (2.12) (derived in Sec. 2.4) and setting angle $\beta = 0°$, the vectors from the centre of the ball to each omni-wheel are defined as:

$$\begin{aligned}\vec{r}_{W,1} &= r_S \begin{bmatrix}\sin(\alpha)\\ 0\\ \cos(\alpha)\end{bmatrix}\\ \vec{r}_{W,2} &= r_S \begin{bmatrix}-\frac{1}{2}\sin(\alpha)\\ \frac{\sqrt{3}}{2}\sin(\alpha)\\ \cos(\alpha)\end{bmatrix}\\ \vec{r}_{W,3} &= r_S \begin{bmatrix}-\frac{1}{2}\sin(\alpha)\\ -\frac{\sqrt{3}}{2}\sin(\alpha)\\ \cos(\alpha)\end{bmatrix}\end{aligned} \tag{3.26}$$

Together with unit vectors in the direction of each omni-wheel:

$$\begin{aligned}\vec{u}_{W,1} &= \begin{bmatrix}0\\ 1\\ 0\end{bmatrix}\\ \vec{u}_{W,2} &= \begin{bmatrix}-\frac{\sqrt{3}}{2}\\ -\frac{1}{2}\\ 0\end{bmatrix}\\ \vec{u}_{W,3} &= \begin{bmatrix}\frac{\sqrt{3}}{2}\\ -\frac{1}{2}\\ 0\end{bmatrix}\end{aligned} \tag{3.27}$$

The circumferential velocities can be calculated and the angular velocity of the omni-wheel defined as:

$$\begin{aligned}\dot{\phi}_i r_W &= (\vec{\omega}_S^{5,5} \times \vec{r}_{W,i}) \cdot \vec{u}_{W,i}\\ \dot{\phi}_i &= \frac{1}{r_W}\Big((\vec{\omega}_S^{5,5} \times \vec{r}_{W,i}) \cdot \vec{u}_{W,i}\Big)\end{aligned} \tag{3.28}$$

Where $i = 1, 2, 3$ and:

$$\begin{aligned}\vec{\omega}_S^{5,5} &= \vec{\omega}_S^{5,I} - \vec{\omega}_5^{5,I}\\ &= \mathbf{R}_2^5 \cdot \vec{\omega}_S^{2,I} - \vec{\omega}_5^{5,I}\end{aligned} \tag{3.29}$$

3.5 Equations of motion

As for the 2D model, the equations of motion will be also obtained using Lagrangian mechanics. The kinetic and potential energy for relevant parts of the BRB and external torques are listed below.

3.5.1 Kinetic and potential energy

In the case of the 3D system, the energies were obtained using [11].

Energy of the ball

The kinetic energy of the ball is defined as:

$$\begin{aligned} T_S &= \frac{1}{2}\vec{T}_2^{2,I^T} \cdot \begin{bmatrix} \mathbf{I}_S^2 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & m_S\mathbf{I}_{3\times3} \end{bmatrix} \cdot \vec{T}_2^{2,I} + \frac{1}{2}\vec{\omega}_S^{2,I^T} \cdot \mathbf{I}_S^2 \cdot \vec{\omega}_S^{2,I} \\ &= \frac{1}{2}\vec{\omega}_2^{2,I^T} \cdot \mathbf{I}_S^2 \cdot \vec{\omega}_2^{2,I} + \frac{1}{2}m_S\vec{v}_2^{2,I^T} \cdot \vec{v}_2^{2,I} + \frac{1}{2}\vec{\omega}_S^{2,I^T} \cdot \mathbf{I}_S^2 \cdot \vec{\omega}_S^{2,I} \\ &= \frac{1}{2}m_S\vec{v}_2^{2,I^T} \cdot \vec{v}_2^{2,I} + \frac{1}{2}\left(\mathbf{R}_I^2 \cdot \vec{\omega}_S^{I,I}\right)^T \cdot \mathbf{I}_S^2 \cdot \left(\mathbf{R}_I^2\vec{\omega}_S^{I,I}\right) \end{aligned} \tag{3.30}$$

Where $\mathbf{I}_{3\times3}$ is a 3x3 identity matrix.

Also in the 3D model, Assumption 3 causes the potential energy of the ball to be equal to zero.

$$V_S = 0 \tag{3.31}$$

Energy of the body

For the body, the kinetic energy is defined similarly as:

$$\begin{aligned} T_B &= \frac{1}{2}\vec{T}_5^{5,I^T} \cdot \begin{bmatrix} \mathbf{I}_B^5 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & m_S\mathbf{I}_{3\times3} \end{bmatrix} \cdot \vec{T}_5^{5,I} \\ &= \frac{1}{2}\vec{\omega}_5^{5,I^T} \cdot \mathbf{I}_B^5 \cdot \vec{\omega}_5^{5,I} + \frac{1}{2}m_B\vec{v}_5^{5,I^T} \cdot \vec{v}_5^{5,I} \end{aligned} \tag{3.32}$$

The potential energy has the form:

$$V_B = m_B \begin{bmatrix} 0 & 0 & g \end{bmatrix} \mathbf{R}_5^I \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix} \tag{3.33}$$

Energy of the omni-wheels

The energy of i-th omni-wheel is denoted by its rotational energy as:

$$T_{Wi} = \frac{1}{2}I_{OW}\dot{\phi}_i^2 + \frac{1}{2}I_M(k\dot{\phi}_i)^2 \tag{3.34}$$

And the potential energy is equal to zero.

$$V_{Wi} = 0 \tag{3.35}$$

3.5.2 External torques

For further computation, external torques need to be defined. As shown in Subsec. 2.4.3, in the case of the 2D plane, it is possible to describe the transformation between the external torques and the real torques using the Jacobian matrix. Therefore, as shown in [11], the definition:

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix} = \mathbf{J} \cdot \begin{bmatrix} \dot{x}_S \\ \dot{y}_S \\ \dot{\psi}_x \\ \dot{\psi}_y \\ \dot{\psi}_z \end{bmatrix} \tag{3.36}$$

can be made and and calculated using Eq. (3.28). The result is obtained using an inversion of the Jacobian matrix:

$$\vec{\tau}_{ext} = \mathbf{J}^T \cdot \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{3.37}$$

Due to its complicated form, the transposition is represented only symbolically.

3.5.3 Lagrangian mechanics

With these external torques, energies, minimal coordinates, and the Lagrangian, the Euler-Lagrange equations can be created.

$$L(\vec{q}, \dot{\vec{q}}) = T_S + T_B + T_W - V_S - V_B - V_W \tag{3.38}$$

$$\frac{d}{dt}\Big(\frac{\partial L}{\partial \dot{q}_i}\Big) - \frac{\partial L}{\partial q_i} = \tau_{ext_i} \tag{3.39}$$

Where $i = 1, 2, 3, 4, 5$. For simplification, the same matrix notation as in the 2D case is used here.

$$\mathbf{M}(\vec{q})\ddot{\vec{q}} + \mathbf{C}(\vec{q}, \dot{\vec{q}})\dot{\vec{q}} + \mathbf{G}(\vec{q}) = \vec{\tau}_{ext} \tag{3.40}$$

Where $\mathbf{M}$ is a matrix of inertial forces due to accelerations, $\mathbf{C}$ is a matrix of coriolis and centrifugal forces, and $\mathbf{G}$ is a matrix of gravitational forces. The full forms of the matrices are not listed here, due to their complexity.

3.5.4 Linearization

Let us define the state vector for the 3D case as:

$$\vec{x} = \begin{bmatrix} \vec{q} \\ \dot{\vec{q}} \end{bmatrix} = \begin{bmatrix} x_S & y_S & \psi_x & \psi_y & \psi_z & \dot{x}_S & \dot{y}_S & \dot{\psi}_x & \dot{\psi}_y & \dot{\psi}_z \end{bmatrix}^T \tag{3.41}$$

and the input vector as:

$$\vec{u} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{3.42}$$

As before, the linearization will be done on an unstable equilibrium point on the top of the ball, with all states and inputs equal to zero.

$$\begin{aligned} \bar{\vec{x}} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \\ \bar{\vec{u}} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \tag{3.43}$$

The state-space representation is given by:

$$\begin{aligned} \dot{\vec{x}} &= \mathbf{A} \cdot \vec{x} + \mathbf{C} \cdot \vec{u} \\ \vec{y} &= \mathbf{C} \cdot \vec{x} + \mathbf{D} \cdot \vec{u} \end{aligned} \tag{3.44}$$

Where **C** is a 10x10 identity matrix, **D** is a 10x3 zero matrix, and other matrices are computed in the same way as in Subsec. 2.5.4. Therefore, after substitusions and calculations, their forms are:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1.8433 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.8433 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 14.3425 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 14.3425 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 7.0841 & -7.0841 \\ -8.1800 & 4.0900 & 4.0900 \\ 3.6237 & -1.8119 & -1.8119 \\ 0 & -3.1383 & 3.1383 \\ -684.4921 & -684.4921 & -684.4921 \end{bmatrix} \end{aligned} \tag{3.45}$$

Further calculations show that the system is fully controllable and observable.

3.6 Calculation of parameters

In the previous section, many parameters were used. The majority of them have been listed in Tab. 2.1. Two of them, however, still need to be defined and calculated [11].

3.6.1 Intertia tensor of the ball

Let us define the inertia tensor of the ball as:

$$\mathbf{I}_S^2 = \begin{bmatrix} I_S & 0 & 0 \\ 0 & I_S & 0 \\ 0 & 0 & I_S \end{bmatrix} \tag{3.46}$$

Where I_S is the moment of inertia of the ball (listed in the reference table 2.1).

3.6.2 Intertia tensor of the body

We define the inertia tensor of the body as:

$$\mathbf{I}_B^5 = \begin{bmatrix} I_{B,x} & 0 & 0 \\ 0 & I_{B,y} & 0 \\ 0 & 0 & I_{B,z} \end{bmatrix} \tag{3.47}$$

Where $I_{B,i}$ is the moment of inertia of the body about the i-th axis of the coordinate frame of its COM.

Defined in Subsec. 2.7.2, the moment of inertia of a cuboid of width w and heigth w, about the x or y axis is given by:

$$I_{x,y} = \frac{m(w^2 + h^2}{12} \tag{3.48}$$

And about the z axis as:

$$I_z = \frac{mw^2}{6} \tag{3.49}$$

Therefore, the moments of inertia of the body about the i-th axis of the COM are:

$$\begin{aligned} I_{B,x} &= \frac{1}{12} m_B (w_B^2 + h_B^2) \\ I_{B,y} &= \frac{1}{12} m_B (w_B^2 + h_B^2) \\ I_{B,z} &= \frac{1}{6} m_B w_B^2 \end{aligned} \tag{3.50}$$

Note that the total mass of the body consists of the mass of the frame, motors, shafts, and omni-wheels.

3.7 Simulations

Several simulations determine the accuracy of the linearized model were made. Similar to simulations of the 2D model, two basic tests were done.

The first test is with initial condition of $\psi_x = 0.1°$:

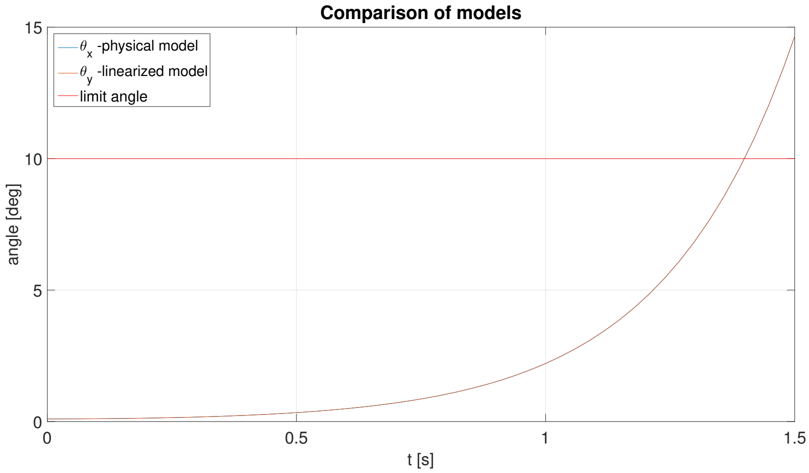


Figure 3.3: Comparison for ψ_x - $\psi_x(t = 0) = 0.1°$

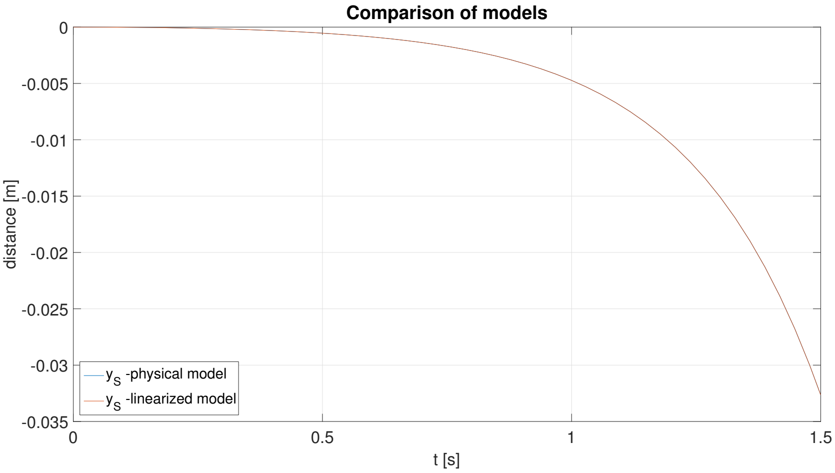


Figure 3.4: Comparison for y_S - $\psi_x(t = 0) = 0.1°$

As can be seen, even after the limit angle is exceeded, the linearized model is a very accurate replacement of the physical model.

The second test is with constant imnput $u_y = 0.01\text{N/m}$ and zero initial conditions:

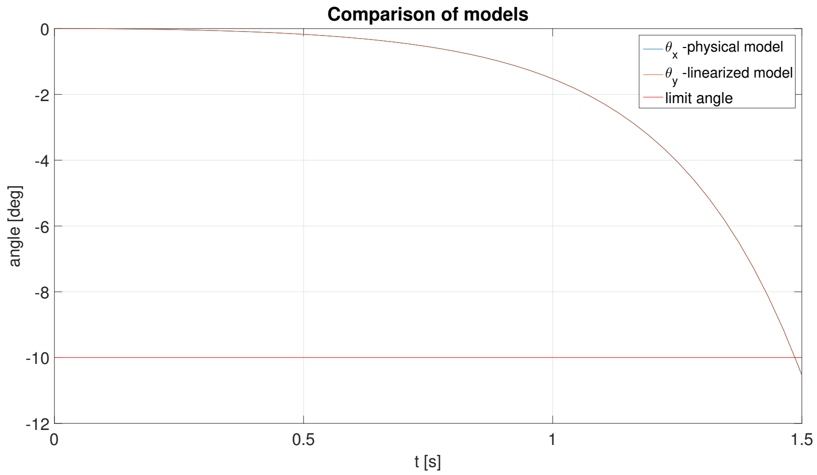


Figure 3.5: Comparison for ψ_x - $u_y = 0.01\text{N/m}$

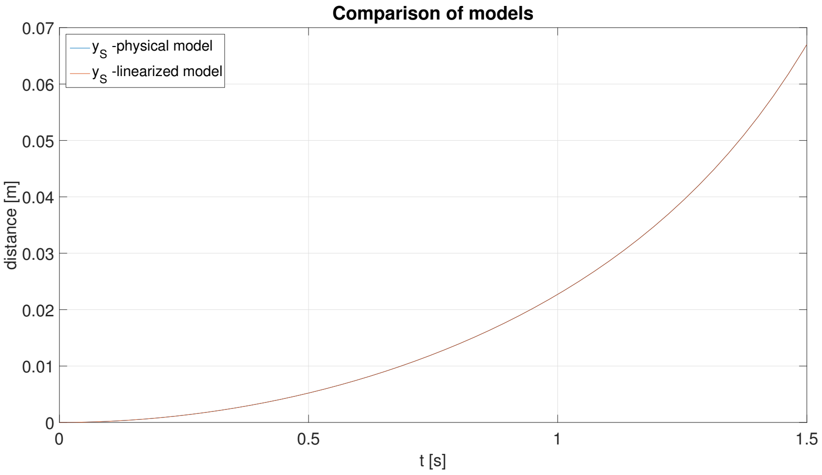


Figure 3.6: Comparison for y_S - $u_y = 0.01\text{N/m}$

Even with constant input, the linearized system acts accurately and can be a satisfactory substitution for the physical model for the limits in which the BRB operates.

Chapter 4

Controller design

In this chapter a controller is designed using the 3D model derived in the previous chapter. The main aims of the controller are to keep the robot stable on the top of the ball and track its trajectory.

In the first section, the methods used in contemporary literature are described. The second section discusses the controller requirements and design approach. Lastly, a Linear-Quadratic Regulator (LQR) controller is designed in the third section.

4.1 Controllers of previously created BRBs

- **CMU**
  Balancing controller: PID controller
  Position controller: PID controller with offline trajectory planning

- **TGU**
  Balancing and position controller: Two PD controllers

- **UA**
  Balancing and position controller: LQR controller with full state feedback and two extra Integral states

- **ETH**
  Balancing controller: Non-linear controller based on LQR theory
  Position controller: Non-linear controller based on LQR theory with feedforward

- **NCHU**
  Balancing and position controller: LQR controller with full state feedback

- **UT**
  Balancing and position controller: LQR controller with full state feedback and later SISO controller

4.2 Requirements and approach

At first, an assumption needs to be made: that the linearized model is approximately equivalent to the nonlinear model in the proximity of the equilibrium point. The proof can be seen in Figures 3.3 - 3.6. Contemporary literature often shows that the angles of the body and position of the ball can be controlled by a single linear controller. Thus this paper also aims to design such a controller, using the same procedure as in [11], and adapting it to the requirements of this project. So firstly, the controller requirements need to be defined.

4.2.1 Design requirements

A well-designed controller is considered to be one that satisfies the following rules:

- the system is internally stable - all the Closed Loop (CL) poles have a negative real part
- the sensitivity function attenuates disturbances up to 2Hz down by at least 80%

$$mag2db(0.2) = -14dB$$

- the CL transfer function attenuates frequencies above 250Hz down by at least 80% (-14 dB), due to Nyquist-Shannon sampling theorem and sampling frequency 500Hz
- the settling time of the step response of pitch and roll angle is between two and four seconds for stabilization
- the torque saturations of motors are minimized.

4.2.2 Design approach

Due to the short project length, simplicity is more important than performance. The LQR design method was therefore chosen due to its short calculating time, despite the fact that finding an optimal controller with LQR theory is a difficult task. However, a suboptimal controller can be found relatively easily.

4.3 LQR controller design

In this section a LQR controller is designed to control the angles of the body and position of the whole robot at the same time.

4.3.1 LQR control theory

LQR is an acronym for Linear-Quadratic Regulator. A special type of proportional derivative controller with feedback and a single gain matrix $\mathbf{K}$, implemented as $\vec{u} = \mathbf{K}\vec{e}$ in the state-space form (3.44), where $\vec{e}$ is an error vector defined as $\vec{e} = \vec{x}_{ref} - \vec{x}$. This results in the CL system shown in Fig. 4.1, with state representation expressed by the following equations:

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

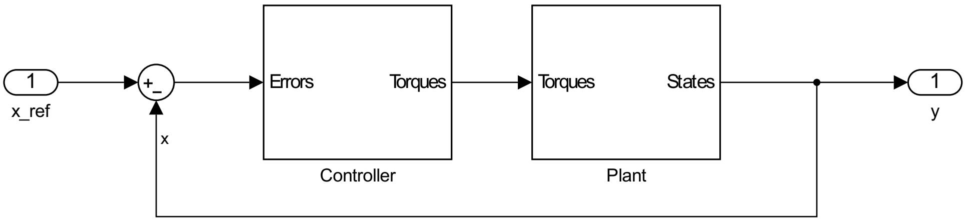


Figure 4.1: The closed loop system

As has been mentioned, the implementation of the control matrix $\mathbf{K}$ is $\vec{u} = \mathbf{K}\vec{e}$. The calculation of this matrix is done by minimizing the quadratic cost function given by:

$$J(\vec{u}) = \int_{t_0}^{t_1} (\vec{e}^T \mathbf{Q}\vec{e} + \vec{u}^T \rho \mathbf{R}\vec{u})dt \tag{4.2}$$

With $\mathbf{Q}$ defined as $\mathbf{N}^T\mathbf{Q}'\mathbf{N}$, where $\mathbf{N}$ defines the plant outputs which need to be controlled and $\mathbf{Q}'$ is the weighting matrix of the controlled outputs. Matrix $\mathbf{R}$ is the weighting matrix of controller outputs, ρ is a positive constant, and the time interval bounds are $t_0 = 0$, $t_1 = \infty$.

As there is no way to find the optimal matrices $\mathbf{Q}$, $\mathbf{R}$, and constant ρ, Bryson's rule [14] is used to estimate the initial values of matrices $\mathbf{Q}'$ and $\mathbf{R}$. Diagonal elements of these matrices are chosen as follows:

$$Q'_{ii} = \frac{1}{e^2_{i,max}} \tag{4.3}$$

$$R_{jj} = \frac{1}{u^2_{j,max}} \tag{4.4}$$

Where $e_{i,max}$ is the maximum possible value of the error with respect to the reference value of the state i, and $u_{j,max}$ is the maximum possible value of the input j. Moreover the parameter ρ also needs to be estimated. When $\rho\mathbf{R} >> \mathbf{Q}$, the control effort cost function is dominant. Similarly when $\rho\mathbf{R} << \mathbf{Q}$, the error cost function is dominant. A mid-point between these two extremes is needed, so the controller focuses on minimizing both extremes. For the beginning state of design, $\rho = 1$ was chosen and further adjusted to satisfy the requirements given in 4.2.1.

4.3.2 Application on the BRB

Due to the linearized model of BRB in Subsec. 3.5.4, all states are controlled. Thus matrix **N** is a 10x10 identity matrix. The maximum possible values of the errors were set approximately to:

$$\begin{aligned} e_{S_x,S_x} = e_{S_y,S_y} &= 0.5 \text{ [m]} \\ e_{\psi_x,\psi_x} = e_{\psi_y,\psi_y} &= 10 \text{ [deg]} \\ e_{\psi_z,\psi_z} &= 360 \text{ [deg]} \end{aligned}$$

and their derivatives to three times the previous values:

$$\begin{aligned} e_{\dot{S}_x,\dot{S}_x} = e_{\dot{S}_y,\dot{S}_y} &= 1.5 \text{ [m/s]} \\ e_{\dot{\psi}_x,\dot{\psi}_x} = e_{\dot{\psi}_y,\dot{\psi}_y} &= 30 \text{ [deg/s]} \\ e_{\dot{\psi}_z,\dot{\psi}_z} &= 1080 \text{ [deg/s]} \end{aligned}$$

Hence the matrix $\mathbf{Q}'$ can be extimated as:

$$\mathbf{Q}' = \begin{bmatrix} \frac{1}{0.5^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{0.5^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{180^2}{(10\pi)^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{180^2}{(10\pi)^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{180^2}{(360\pi)^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{1.5^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{1.5^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180^2}{(30\pi)^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180^2}{(30\pi)^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180^2}{(1080\pi)^2} \end{bmatrix} \tag{4.5}$$

For driving the omni-wheels, three EV3 Large Servo Motors are used, one for each wheel. The maximum torque for a motor of this type is 30 oz/in, which equals roughly 0.21 N/m [21]. The omni-wheels are directly attached to the motors without a transmission. Thus, the maximum torque of a single omni-wheel is identical to the maximum torque of a single motor. Therefore, the matrix **R** is estimated as:

$$\mathbf{R} = \begin{bmatrix} \frac{1}{0.21^2} & 0 & 0 \\ 0 & \frac{1}{0.21^2} & 0 \\ 0 & 0 & \frac{1}{0.21^2} \end{bmatrix} \tag{4.6}$$

The parameter ρ now needs to be tuned. For this purpose, nine values were chosen and nine LQR controllers computed with them. The requirements stated in 4.2.1 have been complied with, and the best controller was chosen.

Figure 4.2 below shows the settling time of the step response of pitch angle for a step signal of size 10°. For $\rho = 0.5$ and above the fourth requirement was not met.

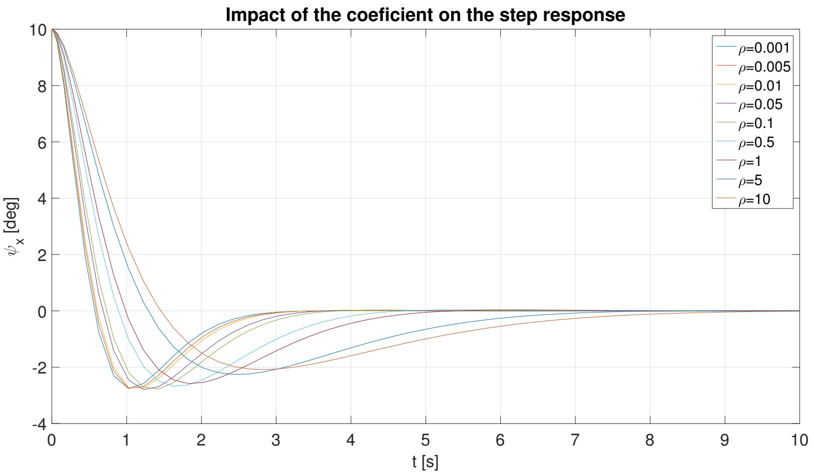


Figure 4.2: Meeting the fourth requirement

4.3.3 Separation of torques

For the next stage in the process, separation of torques is required. This will make distribution of matrix **K** into Proportional Derivative (PD) controllers for each cordinate possible so that these controllers are independent and can be applied in all directions.

This process was already discussed in relation to the 2D model in Sec. 2.4, where the torque τ_z was defined as clockwise positive. In the 3D model, all torques will be defined as counterclockwise positive. Thus the last column of the matrix **J** from the Eq. (2.19) needs to be negative. Substituting $\alpha = 45°$ and $\beta = 0°$ results in:

$$\mathbf{J} = \begin{bmatrix} \frac{2\sqrt{2}}{3} & 0 & -\frac{\sqrt{2}}{3} \\ -\frac{\sqrt{2}}{3} & \frac{\sqrt{6}}{3} & -\frac{\sqrt{2}}{3} \\ -\frac{\sqrt{2}}{3} & -\frac{\sqrt{6}}{3} & -\frac{\sqrt{2}}{3} \end{bmatrix} \tag{4.7}$$

Torques τ_1, τ_2 and τ_3 can now be separated as:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \mathbf{J} \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \tag{4.8}$$

The new linearized system on which the tests for separated coordinates will take place is defined as:

$$\begin{aligned} \dot{\vec{x}} &= \mathbf{A}\vec{x} + \mathbf{BJ}\tilde{u} \\ \vec{y} &= \mathbf{C}\vec{x} \end{aligned} \tag{4.9}$$

Where $\tilde{u} = [\tau_x, \tau_y, \tau_z]^T$.

Note that the coeficients of the controller (matrix **K**) in the fifth and tenth columns are premultiplied by -1 to achieve counterclockwise positive rotation of the torque τ_z.

4.3.4 Frequency domain requirements verification

The transfer function matrix of the state-space model is defined as:

$$\mathbf{P} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{J} \tag{4.10}$$

where $\mathbf{I}$ is a 10x10 identity matrix.

The controller which controls the torques in x, y, and z directions is derived by:

$$\mathbf{K}_{x,y,z} = \mathbf{J}^{-1}\mathbf{K} \tag{4.11}$$

By summing up the combination of gains with relevant coordinates and their derivatives, PD controllers are created. With these controllers and the transfer function matrix, all needed system transfer functions can be calculated. Note that the transfer functions of x_S and y_S are identical. The same is also the case for the transfer functions of ψ_x and ψ_y.

At the beginning, a value of $\rho = 0.01$ was chosen due to its meeting the fourth requirement. However, it can be seen in Figure 4.3 that the sensitivity function for this value does not satisfy the second requirement.

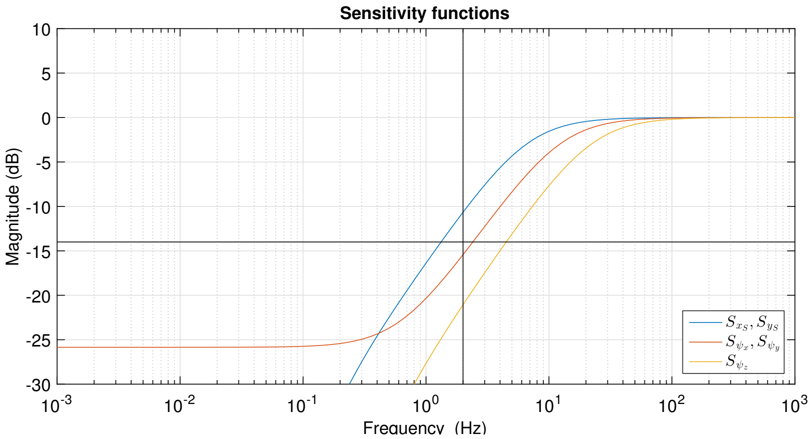


Figure 4.3: Meeting the second requirement ($\rho = 0.01$)

Thus, a new value was chosen to satisfy this requirement. The next two figures show the sensitivity functions for $\rho = 0.005$ and $\rho = 0.001$.

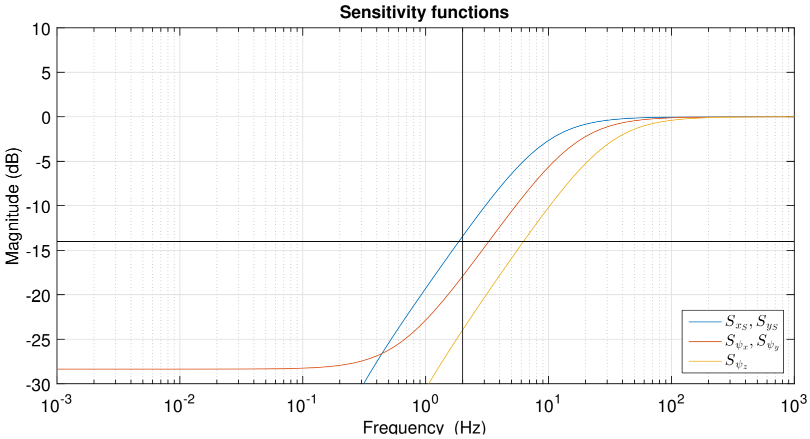


Figure 4.4: Meeting the second requirement ($\rho = 0.005$)

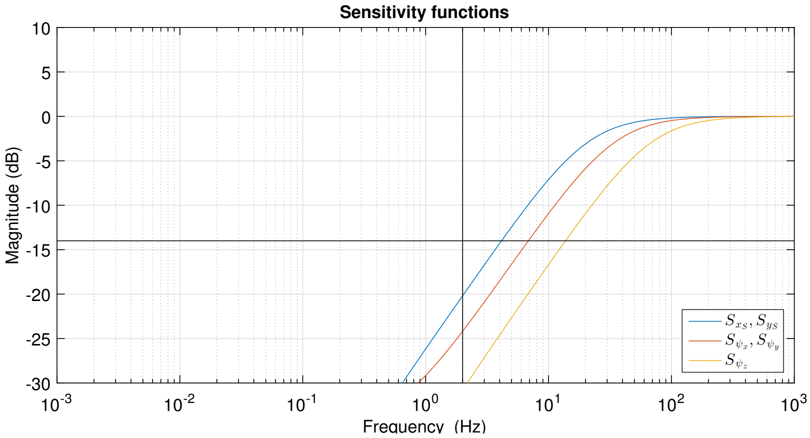


Figure 4.5: Meeting the second requirement ($\rho = 0.001$)

As can be seen, in Figure 4.4 the criterion is almost met. In Figure 4.5 the criterion is met with a large margin. Therefore, the optimal value for ρ will be between these two values. The answer to which value should be chosen is hidden in satisfying the third requirement. Therefore the closed loop transfer functions were obtained and their bode characteristics plotted.

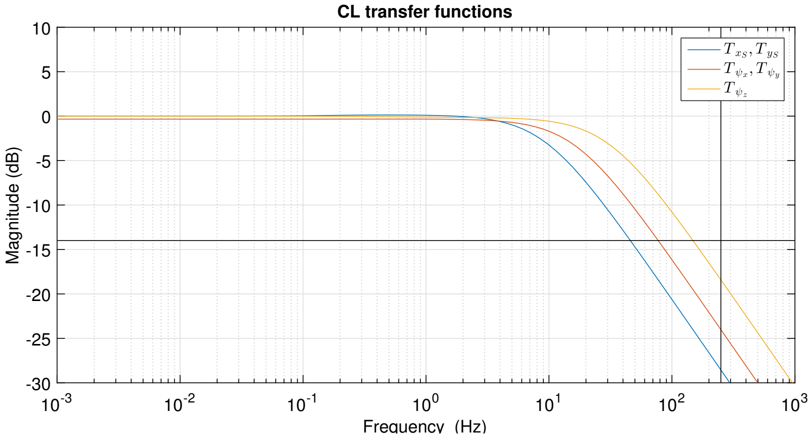


Figure 4.6: Meeting the third requirement ($\rho = 0.005$)

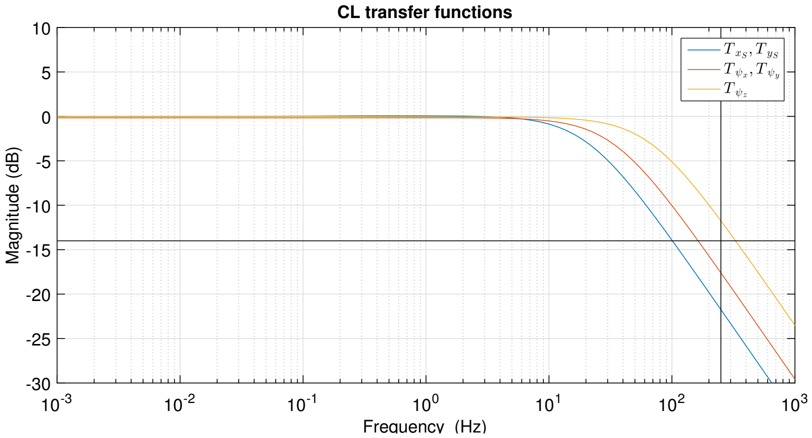


Figure 4.7: Meeting the third requirement ($\rho = 0.001$)

While not meeting the second requirement, $\rho = 0.005$ satisfies the third requirement with a large margin. As can be seen, $\rho = 0.001$ is unsatisfactory. Thus the value of ρ will be a little less than 0.005. From these observations a conclusion can be drawn: $\rho = 0.004$. Below, $\rho = 0.004$ is shown to meet the second, third, and fourth requirements.

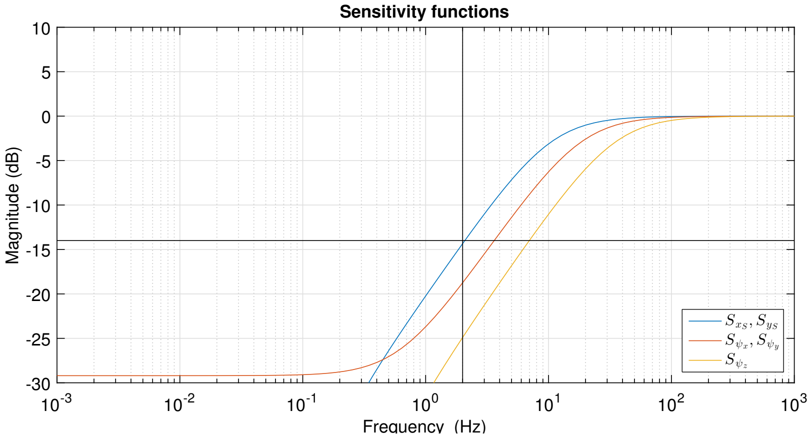


Figure 4.8: Meeting the second requirement ($\rho = 0.004$)

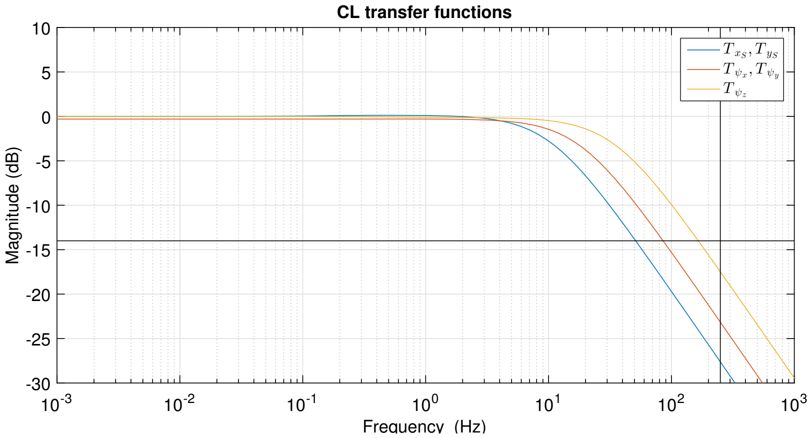


Figure 4.9: Meeting the third requirement ($\rho = 0.004$)

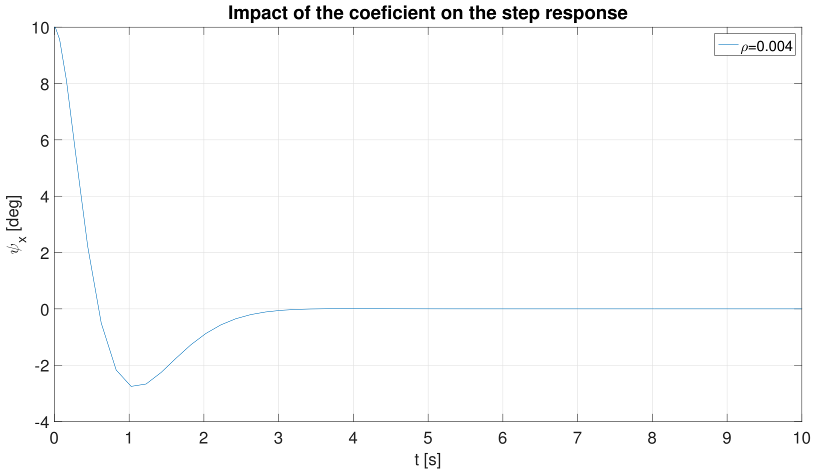


Figure 4.10: Meeting the fourth requirement ($\rho = 0.004$)

Now it remains to verify the first and most important requirement - system stability. To determine if the system is stable, the bode characteristics for all three CL transfer functions need to be observed. For demonstration, only the bode diagram for the first CL transfer function was plotted.

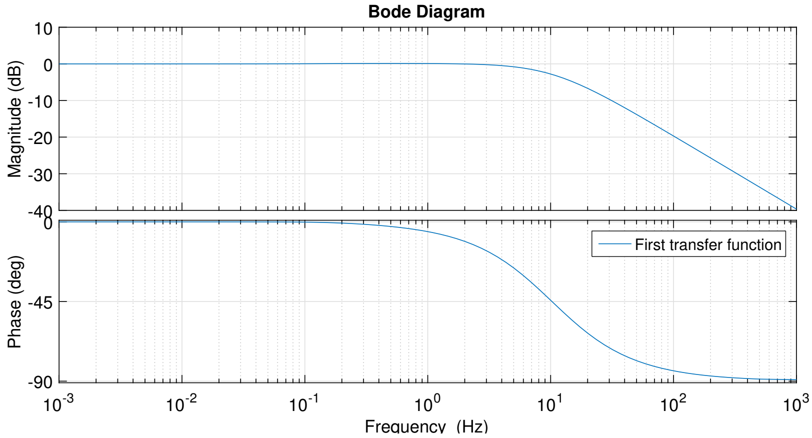


Figure 4.11: Bode plot of the first CL transfer function

The transfer function has, in the Laplace transform, the form of:

$$T_{xy} = \frac{65.49}{s + 64.98} \tag{4.12}$$

Because the transfer functions of coordinate x_S and y_S are the same, the pole is doubled and rewritten in [rad/s] as:

$$p_1 = -1.134$$
$$p_2 = -1.134$$

The transfer function for the second CL transfer function is:

$$T_{\psi_x,\psi_y} = \frac{109.6}{s + 113.3} \tag{4.13}$$

The transfer functions of coordinate ψ_x and ψ_y are the same. Therefore, the pole is doubled and rewritten in [rad/s] as:

$$p_3 = -1.9778$$
$$p_4 = -1.9778$$

The last CL transfer function remains to be estimated. The approximation yields a transfer function:

$$T_{\psi_z} = \frac{212.4}{s + 211.1} \tag{4.14}$$

And a pole which can be rewritten in [rad/s] as:

$$p_5 = -3.684$$

The first requirement of the system stability is met and the final form of controller matrix is:

$$\mathbf{K} = \begin{bmatrix} 0 & 4.6957 & -4.6957 \\ -5.4222 & 2.7111 & 2.7111 \\ -73.4594 & 36.7297 & 36.7297 \\ 0 & 63.6177 & -63.6177 \\ -0.3051 & -0.3051 & -0.3051 \\ 0 & 4.6169 & -4.6169 \\ -5.3311 & 2.6655 & 2.6655 \\ -20.1568 & 10.0784 & 10.0784 \\ 0 & 17.4563 & -17.4563 \\ -0.1032 & -0.1032 & -0.1032 \end{bmatrix}^T . \tag{4.15}$$

4.3.5 Simulations

Further simulations were made to demonstrate controller performance in point stabilization and trajectory tracking. A disturbance signal for coordinates ψ_x, ψ_y of sinus wave of amplitude $5\pi/180$ [rad] and frequency π [rad/s] was created.

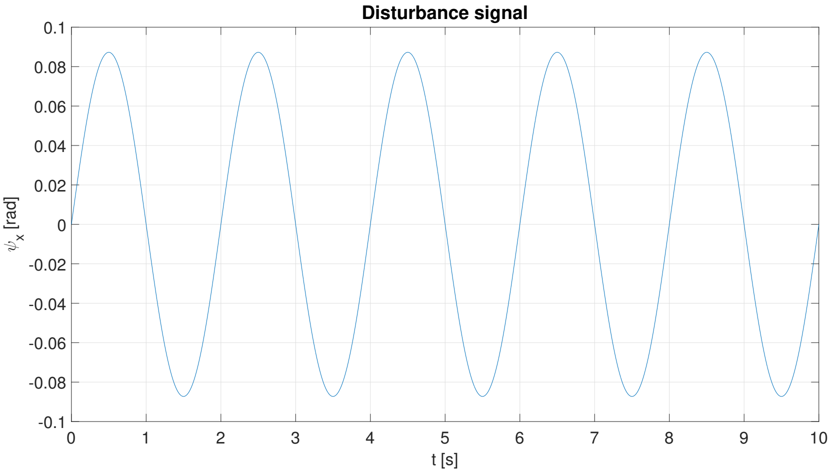


Figure 4.12: Disturbance sinus signal used for angles ψ_x and ψ_y

This signal was gradually applied to the respective angles and the torques were monitored at the output of the controller.

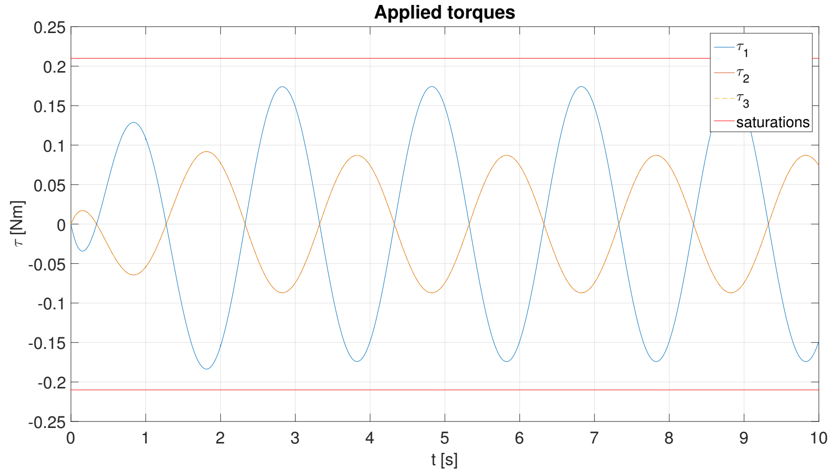


Figure 4.13: Torques for disturbance signal in ψ_x

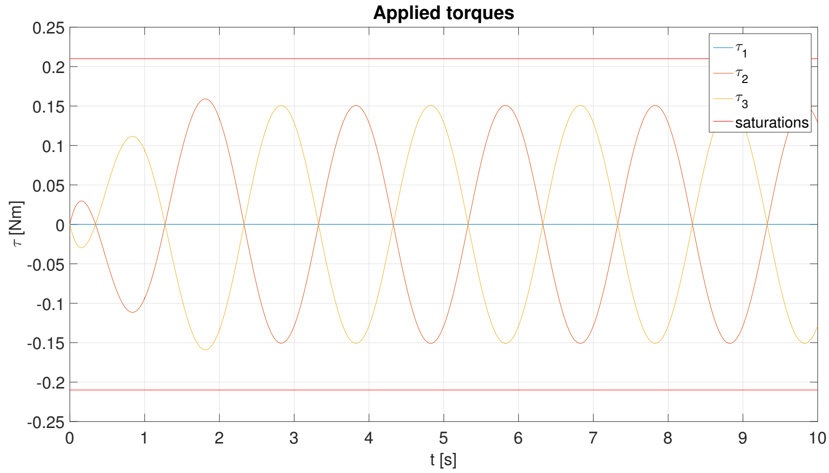


Figure 4.14: Torques for disturbance signal in ψ_y

In both cases, the applied torques did not exceed saturation values.

In the next figures, trajectory tracking related graphs are plotted. The reference signal for both values x_S and y_S was chosen as a ramp with slope 1. For xs a ramp with saturation over 2m was chosen, and for ys with saturation over 1m.

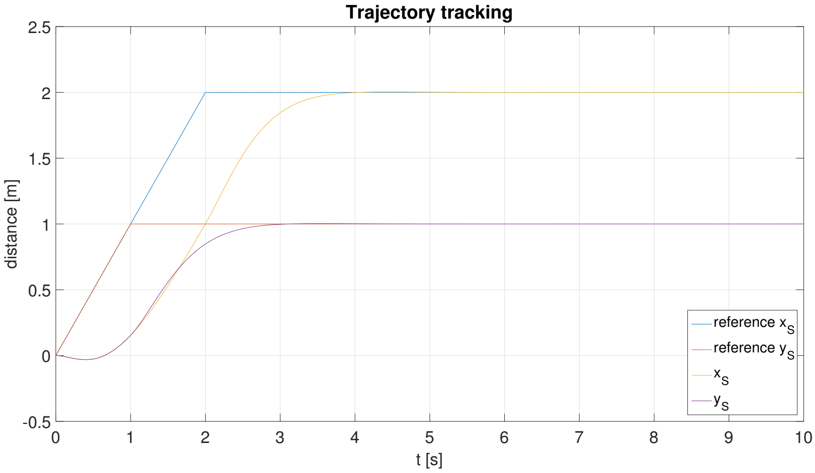


Figure 4.15: Tracking of given trajectory

The reference tracking acted slow but accurate - about half a second is required for a reaction on a reference change. However, the tracking has an approximately identical course with the reference, and is very accurate with time tending to infinity. The applied torques were also observed.

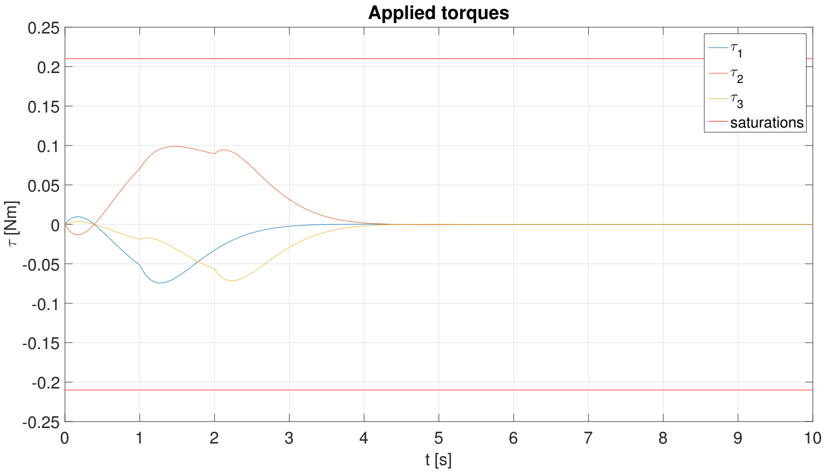


Figure 4.16: Torques for tracking of given trajectory

As in Figures 4.13 and 4.14, the torques were in acceptable range.

Due to this and previous results, the controller was found to be fully satisfactory and implemented into a physical model.

Chapter 5

Robot design

As the title suggests, this chapter is dedicated to the robot design. Firstly the requirements are set. After that, the LEGO Mindstorms EV3 building kit is introduced, followed by a description of the built robot. The building instructions and other necessary steps are also listed.

5.1 Design requirements

Several requirements were stated in previous chapters. Together with requesting LEGO as a building material, the list of requirements is:

- **Position of the COM of the body**
  The COM of the body is located in the centre of horizontal plane, which intersects the body at a reasonable height.

- **Position of gyroscopes**
  Gyroscopes are located near the COM of the body, fixed at the same height and oriented in the direction of x and y axes respectively.

- **Position of omni-wheels**
  Omni-wheels are mounted to the body with a spacing of 120°.

- **Angle of omni-wheels**
  The angle between the body and each omni-wheel is 45°.

- **Building material**
  The whole body is built using the LEGO EV3 building kit.
  The ball is a bowling ball.

5.2 LEGO Mindstorms EV3 building kit

5.2.1 What is LEGO?

LEGO is a well-known Danish construction toy with worldwide popularity. Among its biggest advantages is the compatibility of all sets and clever connection of parts. It is therefore possible to combine sets to create structures of any type or dismantle already assembled creations and reuse bricks without any additional costs. But there are also bad downsides, including: the weight of the bricks, and the complexity of constructing non-rectangular structures.

5.2.2 What is Mindstorms EV3?

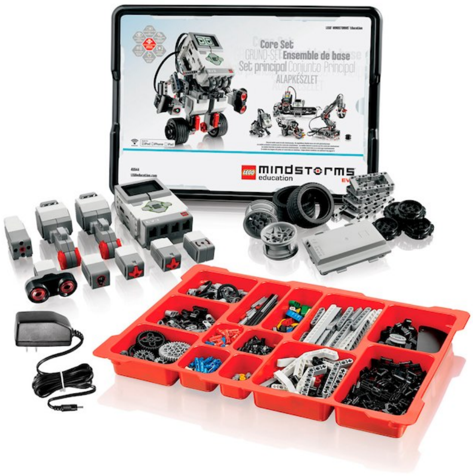


Figure 5.1: LEGO Mindstorms Education EV3 Core Set - Source [16]

Mindstorm is a branch of LEGO products for creation of robots. Both hardware and software are provided for controlling the "intelligent" parts. The EV3 stands for version of the platform. There are in total four platforms [15], sorted from the oldest: RIS, NXT, NXT 2.0, and EV3. The EV3 is the latest and was introduced in 2013. For the purpose of this project, the EV3 was choosen for its modernity and ability to run a selected programming language.

5.2.3 Parts

Due to the robot's complexity, together with the Core Set, the LEGO Mindstorms Education EV3 Expansion Set is also used. A list of all parts contained in these two sets can be found at [17]. Below, the most important parts are described.

Intelligent Brick

Figure 5.2: EV3 Intelligent Brick - Source [18]

The intelligent brick is a programmable controlling part. Here the signals from the sensors are processed, the calculations are made, and the motors are controlled. Communication with other devices is also done here. A Linux-based operating system allows the implementation and running of personal Linux distributions, as well as providing a pre-defined distribution. The brick is powered by six AA batteries or by the rechargable battery pack.

A short overview of the main features:

- ARM 9 processor with Linux-based operating system
- Four input ports for data acquisition of up to 1000 samples per/sec
- Four output ports for execution of commands
- On-board program storage including 16 MB of Flash memory and 64 MB of RAM
- Mini SDHC card reader for 32 GB of expanded memory
- Illuminated, three-color, six-button interface that indicates the brick's active state
- Hi-resolution 178x128 pixel display enabling detailed graph viewing and sensor data observation
- High-quality speaker
- On-brick programming and datalogging that can be uploaded into the EV3 software
- Computer-to-brick communication through on-board USB, or external Wi-Fi or Bluetooth dongles

Reference [18]

Sensors

(a) : EV3 Touch Sensor

(b) : EV3 Gyro Sensor

Figure 5.3: EV3 Sensors - Source [19], [20]

The touch sensor (shown in Fig. 5.3a) is very precise sensor, which can detect multiple presses. The output of this sensor is logical one (pressed) or logical zero (released).

The gyro sensor (shown in Fig. 5.3b) serves to measure rotation and angle changes. The output is either angular velocity or its integration - angle of inclination.

A short overview of the main features:

- Angle mode measures angles with an accuracy of +/- 3 degrees.
- Gyro mode has a maximum output of 440 degrees/second.
- Sample rate of 1 kHz

Reference [20]

LEGO also provides other types of sensors but they are not relevant for this project. Hence their description is not necessary.

Motors

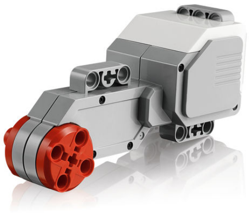


Figure 5.4: EV3 Large Servo Motor - Source [21]

The large servo motor is a very strong motor with a tacho feedback feature. This feature allows precise control, and the signa from the encoder can be also used in brick computations.

A short overview of the main features:

- Tacho feedback to one degree of accuracy
- 160-170 rpm
- Running torque of approximately 21 N/cm
- Stall torque of approximately 40 N/cm

Reference [21]

The set also contains a medium version of this motor (EV3 Medium Servo Motor), which is not specified further as it is not used in the construction of this robot.

5.3 The body

This section describes body design. For clarity the body was divided into 5 parts which combine to create a whole. The design program LEGO Digital Designer was used. Full building instructions can be found in Appendix B - root\building_instructions.

5.3.1 Upper part

The upper part contains the intelligent brick and the touch sensor. They are mounted on the middle part using additional small parts.

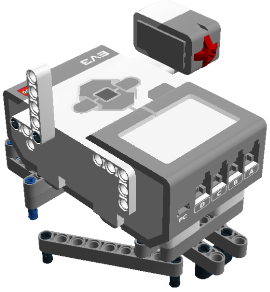


Figure 5.5: Composite upper part

5.3.2 Middle part

As required, the gyroscopes are placed on the central axis, where the COM of the whole body is located. The other requirement, $\beta = 120°$ is met by forming imaginary triangles, on whose vertices the motors will be attached.

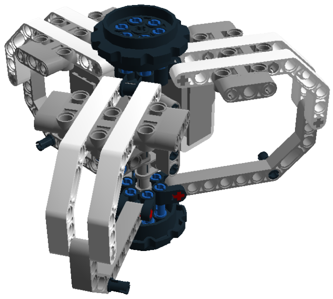


Figure 5.6: Composite middle part

5.3.3 Lower part

The lower part is composed of three identical parts, which are made of motors, omni-wheels, and struts. The angle of the omni-wheels requirement is met, and the mounting of the middle part on the lower part creates a supporting triangle. for driving the ball. Note that due to the lack of a virtual model of the omni-wheel the model uses a normal wheel as a substitute.

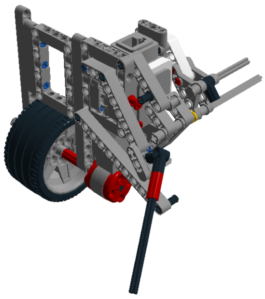
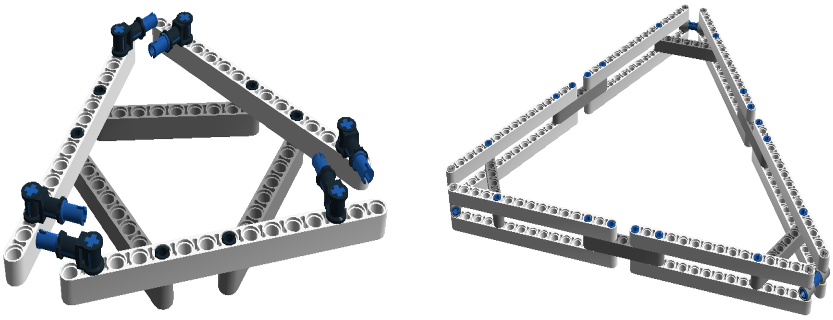


Figure 5.7: Composite one piece of lower part

5.3.4 Frames

For better mass distribution and reinforcment of the joints, the inner and outer frame is made. They are directly attached to the lower part as shown in the 'connection' subfolder.



(a) : Inner frame

(b) : Outer frame

Figure 5.8: Composite frames

5.3.5 Omi-wheels

The Hitechnic Rotacaster Multi-directional wheel was chosen as a suitable omni-wheel. Its diameter is 48mm, weight 39g, and maximum load 18kg.

Figure 5.9: Hitechnic Rotacaster Multi-directional wheel - Source [22]

5.4 The ball

As has been mentioned, the ball on which the robot balances is a bowling ball. Experimental simulations with different inflatable balls show that the robot bounces into the ball and has difficulty manipulating it. Therefore, the bowling ball was chosen because it is a ball of evenly distributed weight with a rigid surface. The mass of the ball was chosen to be 6Lbs and the brand as shown in the following figure:

Figure 5.10: EBONITE: Maxim - Night Sky Bowling Ball - Source [23]

To minimize slip, the whole ball was rubberized. Firstly a base layer to improve adhesion (Body Plasto Fix 340) was applied. Secondly, (Body 950 White) rubber spray was applied multiple times. Lastly, the blue CTU Logo was created at the side of the ball using a conventional blue spray.

Chapter 6

Controller implementation

After completing the assembly of the body, the previously derived LQR controller needs to be implemented. For this, a programming language needs to be chosen, and communication between the computer and the intelligent brick established. The last step before implementing the controller is obtaining the state vector $\vec{x}$.

6.1 Programming language

After several considerations, the Matlab Simulink Visual Programming Language (VPL) was chosen due to its support library for the LEGO Mindstorms EV3, its low computational demands, and its well-arranged design. Like other VPL's, the Simulink lets users create programs by connecting pre-defined or custom blocks. Therefore, the program is more clear for a wider audience. For more information about Simulink, see reference [24].

6.1.1 List of functions

The LEGO Mindstorms EV3 Support from Simulink [25] provides multiple blocks which implement many features. Full description of these features can be found in Mathworks Documentation. For our purpose the following are relevant:

Touch Sensor and Gyro Sensor blocks

The Touch Sensor block (shown in Fig. 6.1a) detects whether a selected EV3 Touch Sensor is being pressed. As stated before, logical one means ’pressed’ and logical zero ’released’.

The Gyro Sensor block (shown in Fig. 6.1b) measures the rate of rotation in degrees per second of a selected EV3 Gyro Sensor.

Note that both sensors are working with a user defined sample time. This time can be set to any number but no lower than the inverted value of the limit frequency listed in Subsec. 5.2.3.

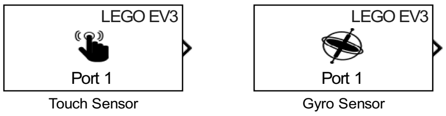


Figure 6.1: Used Simulink sensor blocks

Motor Control and Motor Encoder blocks

The Motor block (shown in Fig. 6.2a) controls the power and direction of a selected EV3 Motor. The block input is a Pulse Width Modulation (PWM) shown as an integer number from 100 to -100. Where 100 denotes full power forward, 0 no power, and -100 full power reverse.

The Encoder block (shown in Fig. 6.2b) is actually also a sensor block. It measures the cumulative rotation of a selected EV3 Motor in degrees. This value does not wrap around after reaching 360 degrees and can be reset at each sample time or by a user defined signal.

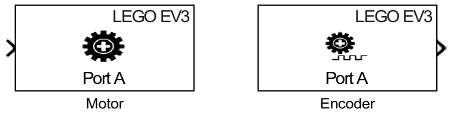


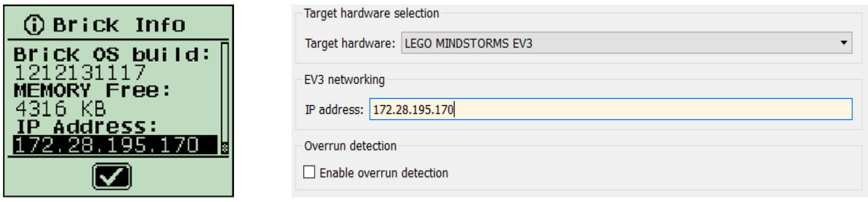
Figure 6.2: Used Simulink motor blocks

6.1.2 Establishment of communication

To establish a communication between the computer and the intelligent brick a WiFi dongle is used. However the brick only supports a few models. The supported dongles are usually old versions with large bodies and low transmission speeds. Therefore, a fan-made articles describing possible extensions of supported devices were made. One of them [26] uses a custom firmware to support the Edimax EW-7811Un WiFi Dongle. This small dongle more than satisfies our speed and size requirements, and is therefore used for this project. The full tutorial, including instalation of the firmware and connection of the adapter is described in the above-mentioned reference.

Figure 6.3: Edimax EW-7811Un WiFi Dongle - Source [27]

The next step is setting the IP address of the brick in a Simulink model. After connecting the brick into a local WiFi network, the IP address can be obtained from the brick (Settings Menu > Brick Info > IP Address). This address needs to be entered into the field (Tools > Run on Target Hardware > Prepare to Run... 'or' Options... > Target hardware: 'LEGO MINDSTORMS EV3' > IP address: 'IP address of the brick' > Apply) in the Simulink model. When set, the programs can run in the brick directly.



Brick Info

Brick OS build:
1212131117
MEMORY Free:
4316 KB
IP Address:
172.28.195.170

(a) : The brick

Target hardware selection

Target hardware: LEGO MINDSTORMS EV3

EV3 networking

IP address: 172.28.195.170

Overrun detection

☐ Enable overrun detection

(b) : The Simulink model

Figure 6.4: Obtaining and setting the IP address

6.2 State vector $\vec{x}$

6.2.1 Angles and angular velocities of the body

The state vector $\vec{x}$ consists of many elements. Two of the most important are angles and angular velocities (rates) of the body. With the EV3 Gyro Sensor there is an option to measure the angular velocity directly. However, the gyro sensor has an offset and suffers from a gradual shift of this offset. Therefore, a more complicated measurement method is used. The gyro sensor is calibrated for one second and the base offset value is determined (shown in Fig. 6.6). The robot needs to be at the equilibrium point and not move. This process is indicated by a red LED light. After this, another block runs with the angle and angular velocity computing, and the offset of the gyro sensor is updated (shown in Fig. 6.7). This process is indicated by an orange LED light and is executed for an indefinite period of time. With this measurement method and a simple conversion between degrees and radians, the four elements of the state vector (ψ_x, ψ_y, $\dot{\psi}_x$, $\dot{\psi}_y$) are computed.

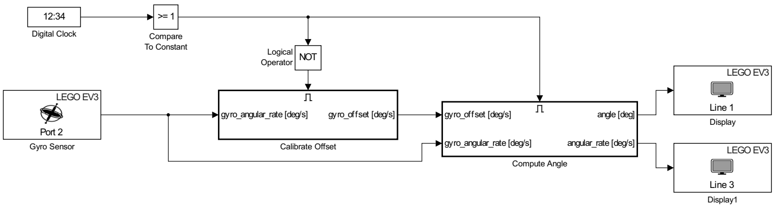


Figure 6.5: Measurement of the value of the gyro sensor

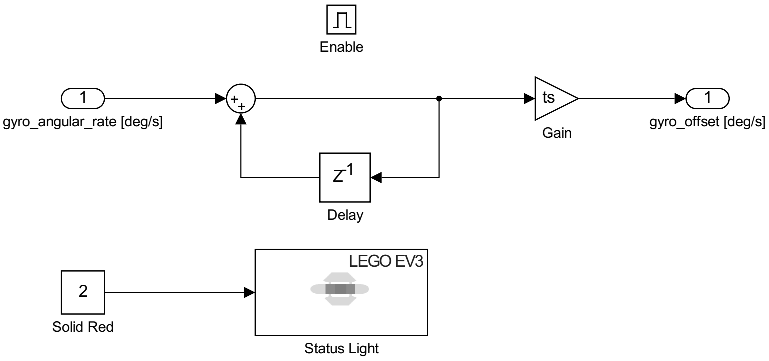


Figure 6.6: Content of the Calibrate Offset block

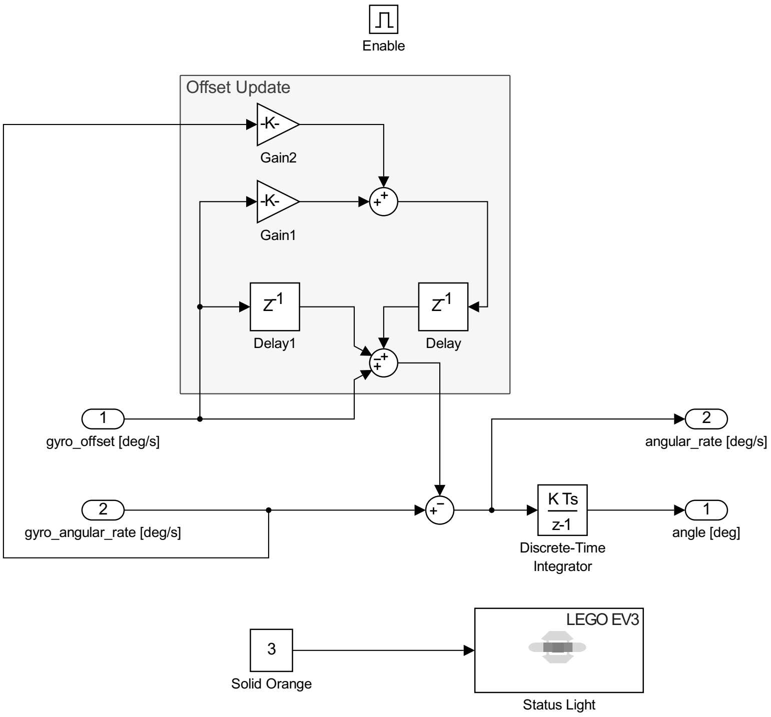


Figure 6.7: Content of the Compute Angle block

Due to the lack of the third gyro sensor, the angle ψ_z and the angular velocity $\dot{\psi}_z$ are computed directly from the angles of the omni-wheels. As shown in [11], the possible way to compute these elements is to use the previously derived angles and angular velocities of the body with the angles and angular velocities of the motors.

In Subsec. 4.3.3 the relation between the angular velocities of the omni-wheels and the decoupled angular velocities of the omni-wheels were defined and computed. Similarly, a reverse relationship can be defined as:

$$\begin{bmatrix}\dot{\phi}_x\\ \dot{\phi}_y\\ \dot{\phi}_z\end{bmatrix} = \mathbf{J}^T \cdot \begin{bmatrix}\dot{\phi}_1\\ \dot{\phi}_2\\ \dot{\phi}_3\end{bmatrix} \tag{6.1}$$

Where:

$$\mathbf{J}^T = \begin{bmatrix}\frac{2\sqrt{2}}{3} & -\frac{\sqrt{2}}{3} & -\frac{\sqrt{2}}{3}\\ 0 & \frac{\sqrt{6}}{3} & -\frac{\sqrt{6}}{3}\\ -\frac{\sqrt{2}}{3} & -\frac{\sqrt{2}}{3} & -\frac{\sqrt{2}}{3}\end{bmatrix} \tag{6.2}$$

The integration of Eq. 6.1 yields:

$$\begin{bmatrix}\phi_x\\ \phi_y\\ \phi_z\end{bmatrix} = \mathbf{J}^T \cdot \begin{bmatrix}\phi_1\\ \phi_2\\ \phi_3\end{bmatrix} \tag{6.3}$$

With the fact that the traveled distance of the ball is opposite to the decoupled traveled distance of the omni-wheels, the relation between them can be made as:

$$r_S \begin{bmatrix}\theta_x\\ \theta_y\\ \theta_z\end{bmatrix} = -r_W \begin{bmatrix}\phi_x\\ \phi_y\\ \phi_z\end{bmatrix} \tag{6.4}$$

It is important to state that the angle θ_z is due to the No slip assumption, equal to zero and in fact, only the body rotates around the z axis in the same direction as the omni wheels. Therefore:

$$\begin{aligned}\psi_z &= -\theta_z\\ &= \frac{r_W}{r_W}\phi_z\end{aligned} \tag{6.5}$$

The angular velocity $\dot{\psi}_z$ is the numerical derivation of this angle.

6.2.2 Positions and linear velocities of the ball

The other vector elements that remain to be determined are positions and linear velocities of the ball. Similar to deriving ψ_z, the positions will be derived using the inverted Jacobian.

Using Eq. (6.4), the position of the ball can be derived from the angles as:

$$\begin{bmatrix}x_S\\ y_S\end{bmatrix} = \begin{bmatrix}r_S\theta_y\\ -r_S\theta_x\end{bmatrix} \tag{6.6}$$

After substituting Eq. (6.3) into Eq. (6.6) the final form is:

$$\begin{bmatrix}x_S\\ y_S\end{bmatrix} = \begin{bmatrix}0 & -r_W & 0\\ r_W & 0 & 0\end{bmatrix} \cdot \mathbf{J}^T \cdot \begin{bmatrix}\phi_1\\ \phi_2\\ \phi_3\end{bmatrix} \tag{6.7}$$

Note that the x_S and y_S are defined in the coordinate frame Ψ_I. However, the decoupled angles ϕ_x, ϕ_y, and ϕ_z are defined in the coordinate frame Ψ_5. Therefore, a conversion needs to be made. Due to the fact that the body is balancing, the influence of ψ_x and ψ_y is negligible. But ψ_z determines the orientation of the robot. Thus the equation is premultipled by the reduced rotational matrix $\mathbf{R_z}$ which ensures a rotation around the z axis:

$$\mathbf{R}_z = \begin{bmatrix} \cos(\psi_z) & \sin(\psi_z) \\ -\sin(\psi_z) & \cos(\psi_z) \end{bmatrix} \tag{6.8}$$

Therefore, the positions of the ball are obtained as:

$$\begin{bmatrix} x_S \\ y_S \end{bmatrix} = \mathbf{R}_z \cdot \begin{bmatrix} 0 & r_W & 0 \\ -r_W & 0 & 0 \end{bmatrix} \cdot \mathbf{J}^T \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} \tag{6.9}$$

The linear velocities are the numerical derivations of these values.

6.3 Driving a single motor

The output of the controller are the torques, which need to be applied to the motors. As shown in Subsec. 3.5.2 the torques acting on the model are in fact the angular rates of the omni-wheels. However, the EV3 Large Servo Motors are driven by the PWM signals. Therefore, a regulator for conversion between the angular rate and the PWM is created.

PWM regulator

The design of such a regulator has been the subject of several articles. The most common way was to identify motor parameters and then use these to calculate the current, which was then used as the input to the system. With the computed current a conversion to the desired voltage was made. The final PWM was calculated using the desired voltage and the voltage of the battery (Resources [29] and [**?**]). However, in our case, the main goal is to drive the motor by the torque, which can actually be converted to the angular velocity. Experiments show that a simple Proportional (P) controller with a gain of size one and feedback can ensure system stability. Therefore the algorithm for conversion between the angular velocity and the PWM is the following:

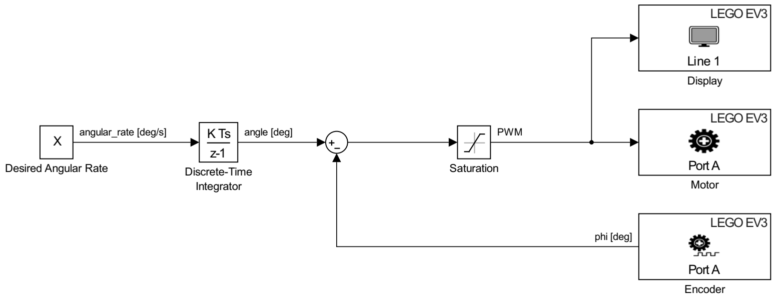


Figure 6.8: PWM regulator

6.4 Touch sensor holder

As was mentioned before, the EV3 Touch Sensor returns the current state (1 or 0). However, for the purposes of this project, the logical one is needed after an odd number of presses, and the logical zero after an even number. With this functionality, a balancing loop can be started and stopped. Therefore, a Sample and Hold block with negative feedback is added to provide the desired functionality.

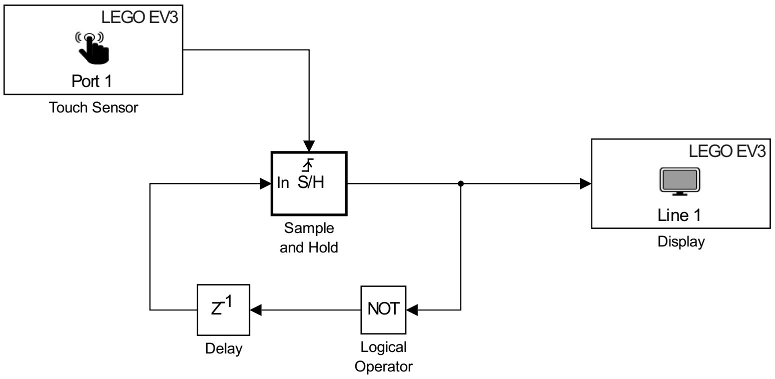


Figure 6.9: Touch sensor holder block

6.5 Final program

The final program can be found in Appendix B and includes some minor changes to the program diagrams above. Firstly, the calibration time of the Gyro Sensors was extended to a duration of 3 seconds to provide better results. Secondly, due to the smaller drift, the filter constant in the Offset Update area in the Calibrate Offset block was changed to 0.001. Lastly, several Stop blocks to stop simulation were added: one for the second press of the touch Sensor and two for the pitch and roll angles to monitor the 10° limit.

6.6 Real-time simulations

Finally, the real time simulations can be done. Three tests are created and the robot reaction to them is observed. The first test was balancing without any disturbance. Note that all tests were done with the stationary ball. See the conclusion for more information.

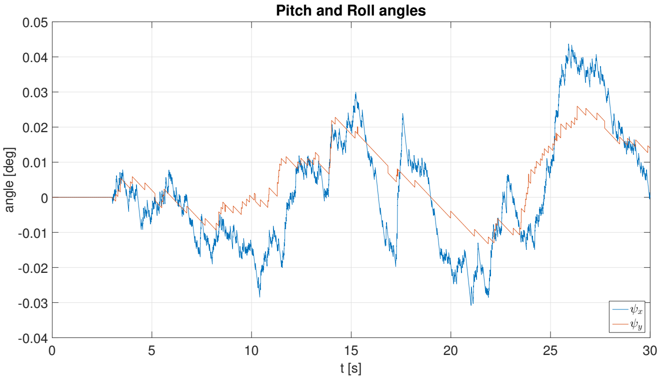


Figure 6.10: Balancing - angles ψ_x, ψ_y

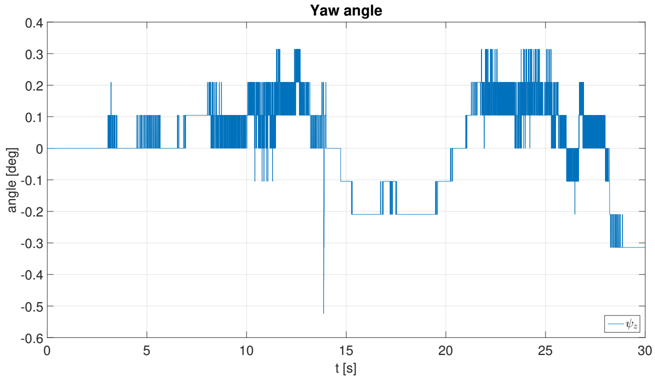


Figure 6.11: Balancing - angle ψ_z

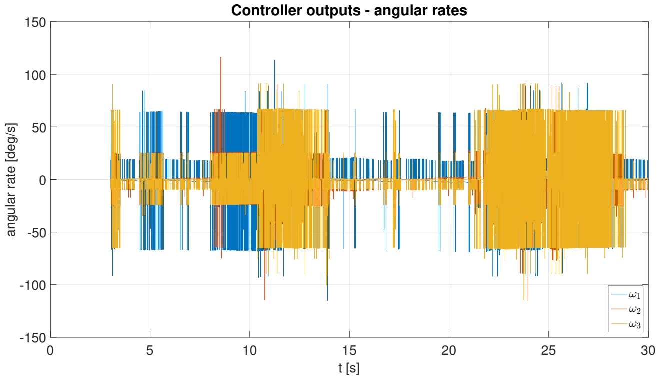


Figure 6.12: Balancing - angular rates ω_1, ω_2, ω_3

As can be seen, the balancing is successful. The Pitch, Roll and, Yaw angles were minimally deflected. In Fig. 6.12 the trembling of the omni-wheels is visible. However, on the real model this disorder does not appear. The trembling is further discussed in the conclusion.

The second test was balancing with disturbances. Two disturbances, first in each of the axes (x and y) and second in the axis x were created by pushing into the body of the robot.

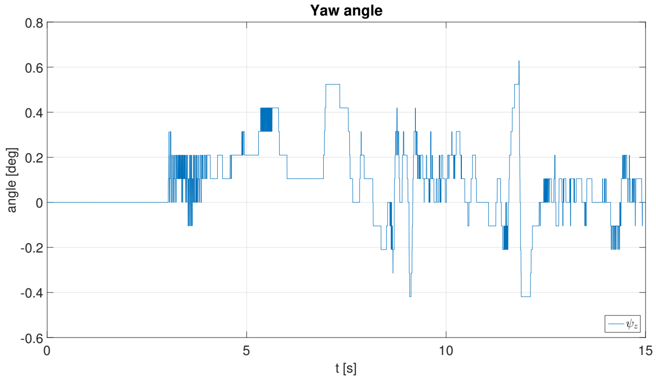


Figure 6.13: Balancing with disturbances - angle ψ_z

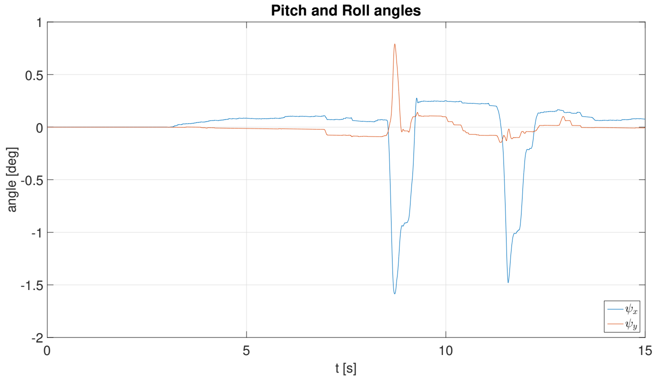


Figure 6.14: Balancing with disturbances - angles ψ_x, ψ_y

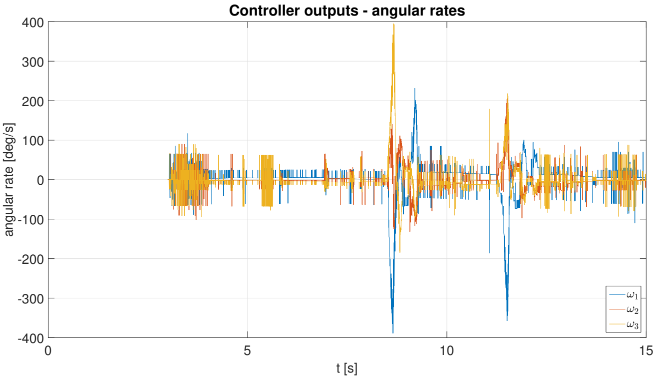


Figure 6.15: Balancing with disturbances - angular rates ω_1, ω_2, ω_3

Also in this test the angles were minimal deflected. The disturbances are visible in Fig. 6.14. In Fig. 6.15, it can be seen that the controller tries to compensate for the disturbances and keep the system in equilibrium.

In the third test the reference tracking was observed. Due to the braked ball, only a value of reference ψ_z was set to a non-zero value of 45°.

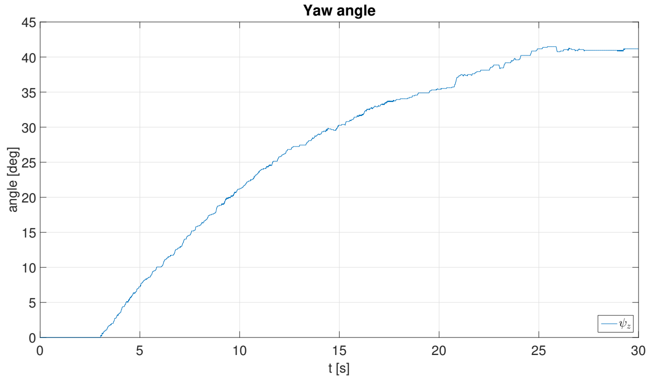


Figure 6.16: Pivoting - angles ψ_x, ψ_y

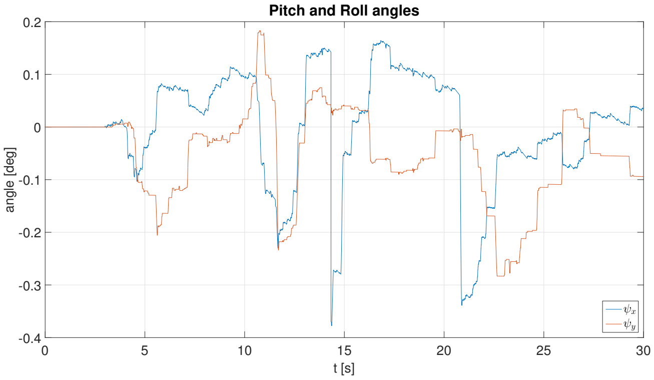


Figure 6.17: Pivoting - angle ψ_z

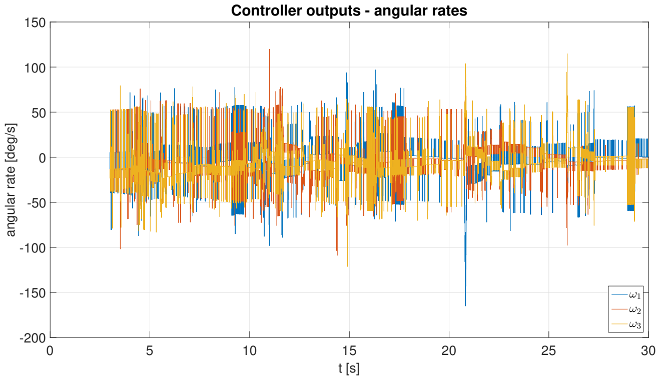


Figure 6.18: Pivoting - angular rates ω_1, ω_2, ω_3

In Fig. 6.16 the reference tracking can be seen. The controller attempts to achieve the angle of 45°. However, due to the computation of ψ_z and the missing Integral (I) regulator term, the tracking is only approximate but satisfactory.

Chapter 7

Conclusion

The first goal of this bachelor thesis was to describe the system dynamics: found in chapter 2 for the 2D model and chapter 3 for the 3D model. While the 2D model is insufficient as an overall control solution, it does allow for a better understanding of dynamic problems, and serves as a basis for the 3D model. The 3D model is a full-featured solution to the problem.

The next goal was to derive a linearized model. A linearization was successfully done on both models. Both linearized models were observed and recognized as sufficient compensation of the physical models at the equilibrium point on the top of the ball.

The third goal was to develop a controller for point stabilization and trajectory tracking. A LQR controller was designed and needed to meet several requirements. The controller fulfilled the requirements with great success and was stated as a well-designed controller. Further simulations on the linearized 3D model show that the controller can provide satisfactory results.

The fourth goal, the creation of a LEGO robot, was also fulfilled. A robot (the body of the BRB) was successfully created and satisfied all design requirements. Together with the ajusted bowling ball (the ball of the BRB), the whole BRB was created.

The last goal was to implement the developed controller. This task was very difficult and required a number of complicated steps. Firstly, reliable conversions between the hardware and software were needed. Therefore, the Measurement of the value of the gyro sensor, PWM regulator, and Touch sensor holder diagrams were created. Together with the derived positions of the ball, linear velocities of the ball, angle ψ_z, and angular velocity $\dot{\psi}_z$, the state vector $\vec{x}$ was created. Using this state vector, the LQR controller with full state feedback of the real model could be formed and real-time simulations done. As was mentioned before, the simulations were completed using a stationary ball. The reason for such a move was the incapability of the regulator to stabilize the body on the moving ball. The cause of this behaviour is not known, but could be due to the limits of the LEGO: the gyroscopes and encoders generate big errors which are brought into the controller and which make a quick and smooth reaction impossible. Another reason could be the weight of the robot. Many of the studies referenced used robots which are heavier than the ball. However, the robot designed in this thesis is lighter than the ball, and therefore the robustness of the dynamics could be disturbed. One solution common to both problems could be to use more accurate sensors and to create the robot from stronger parts to improve load capacity. The design of the controller could also be a factor in this behaviour. However, the simulations in the chapter on controller design suggest that this is not the case. Therefore, at this point in time, the question remains unanswered.

The trembling of the omni-wheels can be observed in any real-time simulation. However, as mentioned before, the effect on the real model is minimal. To reduce the trembling, more simulations would be needed, noise identified, and filters (low pass and high pass) designed to filter the noise. Due to the lack of time and a small consequence on the real model, this task remains to be looked at in a future project.

Due to the stationary ball, the tracking of references (different to zero) could be only tested on angle ψ_z and angular velocity $\dot{\psi}_z$. The simulation with non-zero ψ_z reference was done with success. However, for a better result, the Integral (I) regulator term for tracking the angle could be added in future. The simulation with a non-zero $\dot{\psi}_z$ reference failed, mainly due to the angular velocity $\dot{\psi}_z$ being derived from the angular rates of the omni wheel. One solution to this problem would be to add a third gyro sensor to the robot in the rotation of the axis z and obtaining the angle via flawless discrete integrator and the angular velocity directly. Even with this failed simulation, the pivoting at the stable angular rate can be achieved by removing the feedback of the angle ψ_z and settings its reference to the non-zero value. Future projects - having solved the stationary ball problem - could look at changing the references wireless which would allow the robot to be easily controlled to desired positions.

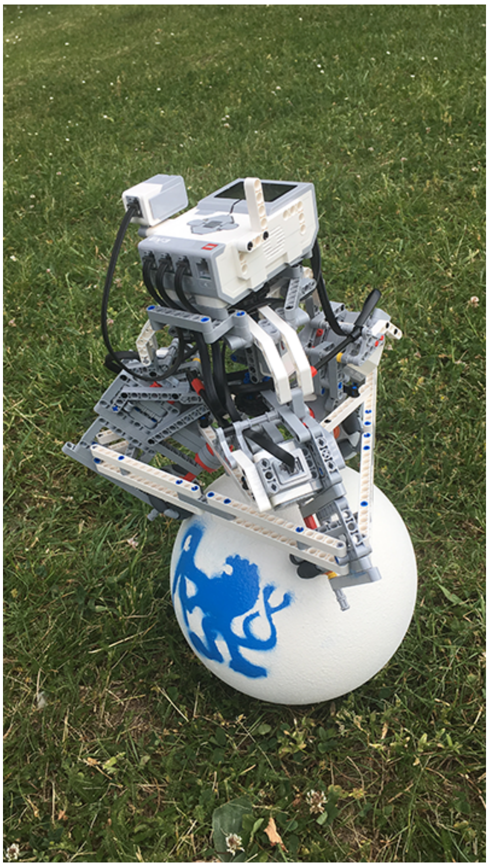


Figure 7.1: BRB CTU

Appendix A

Derivations

Derivation of $\vec{v_{B,yz}^T} \cdot \vec{v_{B,yz}}$

$$\begin{aligned}
\vec{v}_{B,yz}^T \cdot \vec{v}_{B,yz} &= |v_{B,yz}|^2 \\
&= \dot{y}_B^2 + \dot{z}_B^2 \\
&= [\frac{d}{dt}(r_S\theta_x + l\sin(\psi_x))]^2 + [\frac{d}{dt}(l\cos(\psi_x))]^2 \\
&= [r_s\dot{\theta}_x + l\dot{\psi}_x\cos(\psi_x)]^2 + [-l\dot{\psi}_x\sin(\psi_x)]^2 \\
&= r_S^2\dot{\theta}_x^2 + 2r_Sl\dot{\theta}_x\dot{\psi}_x\cos(\psi_x) + l^2\dot{\psi}_x^2\cos^2(\psi_x) + l^2\dot{\psi}_x^2\sin^2(\psi_x) \\
&= r_S^2\dot{\theta}_x^2 + 2r_Sl\dot{\theta}_x\dot{\psi}_x\cos(\psi_x) + l^2\dot{\psi}_x^2
\end{aligned} \tag{A.1}$$

Derivation of $\vec{v_{W,yz}^T} \cdot \vec{v_{W,yz}}$

$$\begin{aligned}
\vec{v}_{W,yz}^T \cdot \vec{v}_{W,yz} &= |v_{W,yz}|^2 \\
&= \dot{y}_W^2 + \dot{z}_W^2 \\
&= [\frac{d}{dt}(r_S\theta_x + (r_S + r_W)\sin(\psi_x))]^2 + [\frac{d}{dt}((r_S + r_W)\cos(\psi_x))]^2 \\
&= [r_s\dot{\theta}_x + (r_S + r_W)\dot{\psi}_x\cos(\psi_x)]^2 + [-(r_S + r_W)\dot{\psi}_x\sin(\psi_x)]^2 \\
&= r_S^2\dot{\theta}_x^2 + 2r_S(r_S + r_W)\dot{\theta}_x\dot{\psi}_x\cos(\psi_x) + (r_S + r_W)^2\dot{\psi}_x^2\cos^2(\psi_x) \\
&+ (r_S + r_W)^2\dot{\psi}_x^2\sin^2(\psi_x) \\
&= r_S^2\dot{\theta}_x^2 + 2r_S(r_S + r_W)\dot{\theta}_x\dot{\psi}_x\cos(\psi_x) + (r_S + r_W)^2\dot{\psi}_x^2
\end{aligned} \tag{A.2}$$

Derivation of ϕ_x

The virtual actuating wheel rotates in a positive direction when the ball rotates in a positive direction. Equating the traveled distance yields:

$$r_W\phi_x = r_S\theta_x \tag{A.3}$$

Also, the virtual actuating wheel rotates in a positive direction when the body rotates in a negative direction. Equating the traveled distance yields:

$$r_W \phi_x = -r_S \psi_x \tag{A.4}$$

Combining Eq. (A.3) and Eq. (A.4) yields:

$$\begin{aligned} r_W \phi_x &= r_S \theta_x - r_S \psi_x \\ &= r_S(\theta_x - \psi_x) \end{aligned} \tag{A.5}$$

Therefore, the derivation of ϕ_x is:

$$\dot{\phi}_x = \frac{r_S}{r_W}(\dot{\theta}_x - \dot{\psi}_x) \tag{A.6}$$

Derivation of $v_{W,xy}^{\vec{T}} \cdot v_{\vec{W},xy}$

$$\begin{aligned} \vec{v}_{W,xy}^T \cdot \vec{v}_{W,xy} &= |v_{W,xy}|^2 \\ &= \dot{x}_{W,xy}^2 + \dot{y}_{W,xy}^2 \\ &= [\frac{d}{dt}((r_S + r_W)\cos(\psi_z))]^2 + [\frac{d}{dt}((r_S + r_W)\sin(\psi_z))]^2 \\ &= [-(r_S + r_W)\dot{\psi}_z \sin(\psi_z)]^2 + [(r_S + r_W)\dot{\psi}_z \cos(\psi_z)]^2 \\ &= (r_S + r_W)^2 \dot{\psi}_z^2 \sin^2(\psi_z) + (r_S + r_W)^2 \dot{\psi}_z^2 \cos^2(\psi_z) \\ &= (r_S + r_W)^2 \dot{\psi}_z^2 \end{aligned} \tag{A.7}$$

Derivation of ϕ_z

Due to the No slip assumption, the virtual actuating wheel only rotates in a positive direction when the body rotates in a negative direction. Equating the traveled distance yields:

$$r_W \phi_z = -r_S \psi_z \tag{A.8}$$

Therefore, the derivation of ϕ_z is:

$$\dot{\phi}_z = -\frac{r_S}{r_W}\dot{\psi}_z \tag{A.9}$$

Derivation of moments of inertia of virtual wheel

Energy equilibrium of rotational energies around the x axis:

$$\begin{aligned} \frac{1}{2} I_{W,x} \dot{\phi}_x^2 &= \frac{1}{2} I_{OW}(\dot{\phi}_x^2 \cos(\alpha))^2 + \frac{1}{2} I_M (k\dot{\phi}_x \cos(\alpha))^2 \\ &+ 2[\frac{1}{2} I_{OW}(-\frac{1}{2}\dot{\phi}_x \cos(\alpha))^2 + \frac{1}{2} I_M(-k\frac{1}{2}\dot{\phi}_x \cos(\alpha))^2] \\ &= \frac{1}{2}[I_{OW}\cos(\alpha)^2]\dot{\phi}_x^2 + \frac{1}{2}[I_M k^2 \cos(\alpha)^2]\dot{\phi}_x^2 \\ &+ \frac{1}{2}[2I_{OW}\frac{1}{4}\cos(\alpha)^2]\dot{\phi}_x^2 + \frac{1}{2}[2I_M \frac{1}{4}k^2 \cos(\alpha)^2]\dot{\phi}_x^2 \end{aligned} \tag{A.10}$$

Therefore:

$$\begin{aligned} I_{W,x} &= \cos(\alpha)^2(I_{OW} + k^2 I_M + \frac{1}{2} I_{OW} + \frac{1}{2} k^2 I_M) \\ &= \frac{3}{2}\cos(\alpha)^2(I_{OW} + k^2 I_M) \end{aligned} \tag{A.11}$$

Energy equilibrium of rotational energies around the y axis:

$$\begin{aligned} \frac{1}{2} I_{W,y}\dot{\phi}_y^2 = & \frac{1}{2} I_{OW}(\frac{1}{2}\sqrt{3}\dot{\phi}_y \cos(\alpha))^2 + \frac{1}{2} I_M(k\frac{1}{2}\sqrt{3}\dot{\phi}_y \cos(\alpha))^2 \\ & + \frac{1}{2} I_{OW}(-\frac{1}{2}\sqrt{3}\dot{\phi}_y \cos(\alpha))^2 + \frac{1}{2} I_M(-k\frac{1}{2}\sqrt{3}\dot{\phi}_y \cos(\alpha))^2 \\ & = \frac{1}{2}[2I_{OW}\frac{3}{4}\cos(\alpha)^2]\dot{\phi}_y^2 + \frac{1}{2}[2I_M k^2 \frac{3}{4}\cos(\alpha)^2]\dot{\phi}_y^2 \end{aligned} \tag{A.12}$$

Therefore:

$$\begin{aligned} I_{W,y} &= \frac{3}{2} I_{OW}\cos(\alpha)^2 + \frac{3}{2} I_M k^2 \cos(\alpha)^2 \\ &= \frac{3}{2}\cos(\alpha)^2(I_{OW} + k^2 I_M) \end{aligned} \tag{A.13}$$

Energy equilibrium of rotational energies around the z axis:

$$\begin{aligned} \frac{1}{2} I_{W,z}\dot{\phi}_z^2 &= 3[\frac{1}{2} I_{OW}(\sin(\alpha)\dot{\phi}_z)^2 + \frac{1}{2} I_M(k\sin(\alpha)\dot{\phi}_z)^2] \\ &= \frac{1}{2}[3I_{OW}\sin(\alpha)^2]\dot{\phi}_z + \frac{1}{2}[3I_M k^2 \sin(\alpha)^2]\dot{\phi}_z \end{aligned} \tag{A.14}$$

Therefore:

$$I_{W,z} = 3\sin(\alpha)^2(I_{OW} + k^2 I_M) \tag{A.15}$$

The derivations were taken from [11].

Appendix B

Content of the enclosed CD

Below the folders and the most important files are listed:

- root
  - building_instructions
    - upper_part
    - middle_part
    - lower_part
    - inner_frame
    - outer_frame
    - connection
  - programs
    - 2d_model
    - 3d_model
    - lqr
    - gyro_program
    - motor_program
    - touch_program
    - final_program
      - BRB.slx
  - website_czech
    - index.html

Extra folder 'website_czech' contains construction and programming guide for the robot in Czech language at the request of thesis supervisor.

Appendix C

Bibliography

[1] T. Lauwers, G. A. Kantor, and R. L. Hollis, *A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive*, Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, pages 2884-2889. IEEE, 2006.

[2] U. Nagarajan, *Dynamic constraint-based optimal shape trajectory planner for shapeaccelerated underactuated balancing systems*, Robotics: Science and Systems. IEEE, 2010

[3] 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*, Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pages 998-1003. IEEE, 2009

[4] U. Nagarajan, G. Kantor, and R. Hollis, *Integrated planning and control for graceful navigation of shape-accelerated underactuated balancing mobile robots*, Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 136-141. IEEE, 2012

[5] U. Nagarajan, B. Kim, and R. Hollis, *Planning in high-dimensional shape space for a singlewheeled balancing mobile robot with arms*, Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 130-135. IEEE, 2012

[6] M. Kumaga and T. Ochiai, *Development of a robot balanced on a ball: Application of passive motion to transport*, 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*, Control, Automation and Systems, 2008. ICCAS 2008. International Conference on, pages 433-438. IEEE, 2008

[8] J. Fong and S. Uppill, *Ballbot: Preliminary report*, Mechatronics honours project, University of Adelaide, 2009

[9] P. Fankhauser and C. Gwerder, *Modeling and control of a ballbot*, Bachelor thesis, Swiss Federal Institute of Technology Zürich, 2010

[10] C.-C. Tsai, C.-K. Chan, and L.-C. Kuo, *Lqr motion control of a ball-riding robot*, Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on, pages 861-866. IEEE, 2012

[11] Koos van der Blonk, *Modeling and Control of a Ball-Balancing Robot*, Internship & Master thesis at ALTEN Mechatronics, 2014

[12] Péter Fankhauser, *Meet Rezero, the dancing ballbot*, Available: `https://www.youtube.com/watch?v=ACohrH64YKs`

[13] J. Meriam and L. Kraige, *Parallel axis theorem*, Engineering Mechanics: Dynamics, volume 2, pages 651-655. John Wiley & Sons Incorporated, 2012

[14] G. Franklin, J. Powell, and A. Emami-Naeini, *Bryson's rule*, Feedback Control of Dynamic Systems, volume 5, page 493. Pearson, 2006

[15] *History of LEGO Robotics*, Available: `https://www.lego.com/cs-cz/mindstorms/history`

[16] *LEGO MINDSTORMS Education EV3 Core Set product description*, Available: `https://education.lego.com/en-us/products/lego-mindstorms-education-ev3-core-set-/5003400`

[17] *What's in the Box? LEGO MINDSTORMS Education EV3*, Available: `https://education.lego.com/en-us/support/mindstorms-ev3/whats-in-the-box`

[18] *EV3 Intelligent Brick product description*, Available: `https://education.lego.com/en-us/products/ev3-intelligent-brick/45500`

[19] *EV3 Touch Sensor product descriptio*, Available: `https://education.lego.com/en-us/products/ev3-touch-sensor/45507`

[20] *EV3 EV3 Gyro Sensor product description*, Available: `https://education.lego.com/en-us/products/ev3-gyro-sensor-/45505`

[21] *EV3 Large Servo Motor product description*, Available: `https://education.lego.com/en-us/products/ev3-large-servo-motor/45502`

[22] *A Catalog of Omni-directional or Holonomic Wheels for Lego Robots*, Available: `http://www.brickengineer.com/pages/2011/08/23/a-catalog-of-omni-directional-or-holonomic-wheels-for-lego-robots/`

[23] *EBONITE: Maxim - Night Sky overview*, Available: `http://www.ebonite.com/products/balls/retired-balls/maxim-night-sky`

[24] *Simulink - Simulation and Model-Based Desig*, Available: `https://www.mathworks.com/products/simulink.html`

[25] *LEGO MINDSTORMS EV3 Support from Simulink*, Available: `https://www.mathworks.com/hardware-support/lego-mindstorms-ev3-simulink.html`

[26] *HOWTO Configure WiFi EDIMAX EW-7811Un on Mindstorms EV3*, Available: `http://thetechnicgear.com/2014/04/howto-configure-wifi-edimax-ew-7811un-mindstorms-ev3/`

[27] *EDIMAX EW-7811Un 150 Mbps Wireless IEEE 802.11b/g/n Nano USB Adapter*, Available: `https://www.newegg.com/Product/Product.aspx?Item=N82E16833315091`

[28] Ashley C. Mitchell, B.S.A.E, *MODELING AND CONTROL OF A MOTOR SYSTEM USING THE LEGO EV3 ROBOT*, Thesis Prepared for the Degree of MASTER OF SCIENCE, UNIVERSITY OF NORTH TEXAS, August 2015

[29] *Mathematical Model of Lego EV3 Motor*, Available: `http://nxt-unroller.blogspot.cz/2015/03/mathematical-model-of-lego-ev3-motor.html`