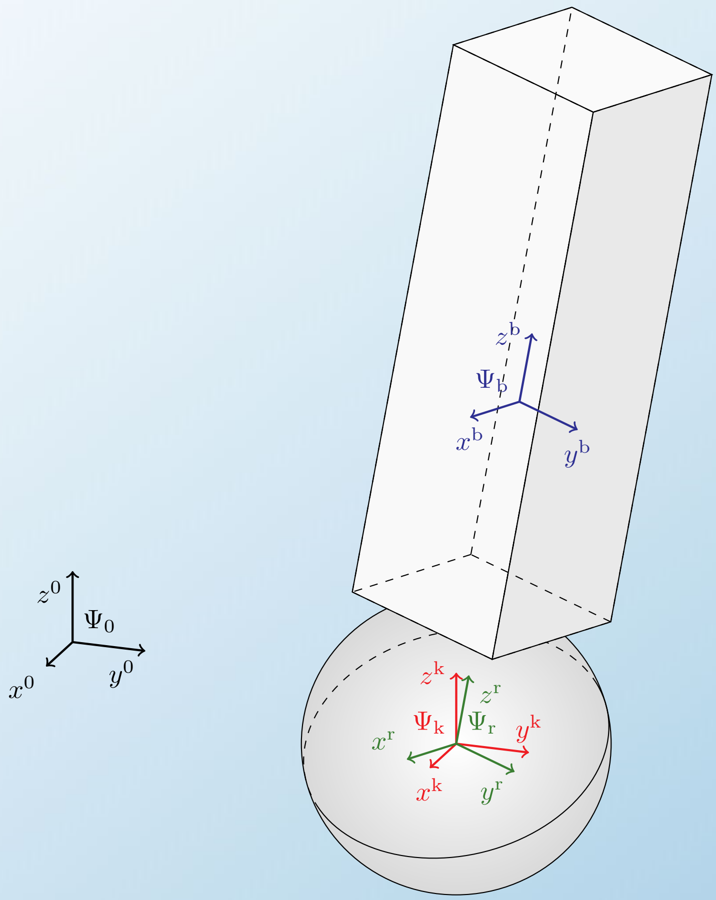
Geometric control of an under-actuated balancing robot

Cees Verdier

Master of Science Thesis



Delft University of Technology

Delft Center for Systems and Control

Geometric control of an under-actuated balancing robot

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Cees Verdier

22nd July 2015

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was conducted in collaboration with Alten Mechatronics. Their cooperation is hereby gratefully acknowledged.

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Geometric control of an under-actuated balancing robot

by

Cees Verdier

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: 22nd July 2015

Supervisor(s): ____________________
Dr.ir. G. Delgado Lopes

____
Ir. G-J. Heldens

Reader(s): ____________________
Dr.ir. G. Delgado Lopes

____
Prof.dr.ir. M. Wisse

____
Dr.ir. D. Jeltsema

____
Ir. G-J. Heldens

Abstract

Ball-balancing robots, or Ballbots, are under-actuated omni-directional mobile robots that balance on top of a single ball. The under-actuated nature arises from the fact that both position and attitude of the robot are actuated by the same actuators. This thesis introduces a geometric approach to the control of ball-balancing robots.

In this approach, a new full 3D model is derived using screw theory. Based on this model, a geometric observer and geometric controller are proposed. Two methodologies are implemented, a computed torque controller and a sliding mode controller, that can track the attitude of the robot on either the special orthogonal group $SO(3)$ or the 2-sphere S^2. Position control is achieved through the use of the relation between the desired linear acceleration and the attitude of the robot. The resulting desired attitude is tracked by one of the geometric attitude controllers.

Simulation results show the effectiveness of the proposed controllers. The proposed sliding mode controller is shown to be more robust to model uncertainties. The position controller is shown to be able to control the position and follow trajectories, but overall global stability is not guaranteed. Recommendations are made to improve the performance. For the geometric observer, stability is shown under a set of assumptions. The work is concluded with a set of experiments on a real platform.

Contents

List of Figures

Acknowledgements

I would like to thank my supervisors Gabriel A. D. Lopes and Geert-Jan Heldens for their guidance and feedback during this MSc thesis.

I would like to thank Alten Mechatronics, who made this project possible. They provided a positive and personal environment in which it was a pleasure to work. I would like to thank everybody at Alten Mechatronics who contributed to this project. I especially would like to thank the temporary members of the ball-balancing robot team, including Koos van der Blonk, Joris van de Weem, John Gardenier and Jan Swinkels.

I would also like to thank Arjan Verboord, Rubén Varela and Rens Louwerens for their help with the implementation of the software on the robot.

Finally I would like to thank the business manager Chris Kalis for his personal management style and conversations that helped me broaden my vision.

Delft, University of Technology — Cees Verdier
22nd July 2015

Chapter 1

Introduction

1-1 Motivation

Ball-balancing robots, or 'ballbots' are under-actuated omni-directional mobile balancing robots that balance on top of a single ball. The ballbot can be argued to be a three dimensional version of a two-wheeled inverted pendulum robot, such as for example a Segway. While the Segway has to balance in only one degree of freedom, the ball-balancing robot has to balance in two degrees of freedom. The ballbot has as main advantage over a Segway robot that it can move instantaneously in every direction, while the Segway has to rotate around its own axis to get a certain heading direction. Additionally, the footprint of the robot is smaller, making such a robot useful in tight environments.

Well-known examples of these ball-balancing robot are ballbot of the Carnegie Mellon University [1], BallIp of the Tohoku Gakuin University [2] and Rezero of the Eidgenössische Technische Hochshule Zürich [3], see Figure 2-2.

Controlling the ball-balancing robot is challenging. First of all, the robot is under-actuated. Under-actuated systems have a lower number of actuators than degrees of freedom [4]. As a result the same actuators have to be used for multiple degrees of freedom. The under-actuation of the ballbot arises from the property that the robot should control the position of the ball $(x, y) \in \mathbb{R}^2$ and the rotation of the body $R \in SO(3)$, thus in total 5 degrees of freedom, by using only 2 to 3 actuators.

Secondly, the dynamics of the robot are highly three-dimensional and are primarily defined by three dimensional rotations. A three dimensional rotation of a rigid body can be represented by a rotation matrix on the special orthogonal group $SO(3)$, which is a *differential manifold*. In previous work the rotations are parametrised by either Euler angles, e.g. [1,5], or quaternions, e.g. [6]. These parametrisations are *local charts* on $SO(3)$. There are multiple combinations to parametrise the rotation matrix with Euler angles, i.e. the parametrisation is not unique.

To illustrate the concepts of a differential manifold and a local coordinate chart, consider for example the earth as a manifold and a map as a local coordinate chart. The earth is a three dimensional sphere, while the map is a two dimensional surface.

Although the two objects are geometrically very different, it is possible to represent the sphere locally by the map, i.e. points on the earth map to points on the map. However, the choice of a map is not unique. For example, the Pacific Ocean can be depicted in the middle of the map, at the side or it could even be split up into two parts.

Given that a certain local coordinate chart is not unique, e.g. a certain choice of Euler angles, also the model and control law using this local chart will not be unique and will only be valid for that specific choice of the coordinate map.

Furthermore, as the word *local* suggests, the choice of Euler angles does not cover $SO(3)$ globally. As a result, a choice of Euler angles can be subjected to singularities for certain angles or some angles result in a gimbal lock. Alternatively, the rotation matrix can be parametrised by quaternions. This parametrisation is not subjected to singularities, but instead double covers $SO(3)$, i.e. two quaternions give the same rotation.

In this work, it is chosen to take a 'geometric' approach to the modelling and control of an under-actuated balancing robot. Here the term geometric is used to indicate a coordinate-free approach, i.e. the dynamics and control of the system are not parametrised using e.g. Euler angles or quaternions. Instead the model and control are directly derived for the underlying differential manifold, such as for example $SO(3)$.

The benefit of such an approach that the dynamics and control laws are directly represented on the underlying manifolds, instead of (often local) approximations of these manifolds. The resulting model and controller are coordinate invariant and give a global representation. This is especially useful for a system such as the ball-balancing robot, as the dynamics are primarily given by three dimensional rotations.

The direct use of the differential manifolds comes with the need of alternative error functions. A common way to express an error between two points is the Euclidean error: $e = x_1 - x_2$. This error makes sense for flat spaces. However, for manifolds such as the special orthogonal group, the subtraction of two rotation matrices loses its geometric interpretation. Instead, nonlinear error functions for the underlying manifolds will be used. Using these error functions, control techniques such as computed torque and sliding mode control will be extended to control the ball-balancing robot directly on differential manifolds, such as $SO(3)$.

Throughout this work multiple the system will be evaluated in different spaces. An overview of these spaces are shown in Figure 1-1. First of all, the system consists of two bodies. The workspace of these bodies can be expressed by $SE(3) \times SE(3)$. In the modelling chapter, it will be found that the dynamics can be fully described by two angular acceleration and the configuration space is reduced to $SO(3) \times SO(3) \times \mathbb{R}^2$. Note that $\mathbb{R}^2$ is still included in the configuration space, as it holds the information about the position of the robot in space, even though the translation does not influence the dynamics. The desired goal of the robot will be to control the attitude on either S^2 or $SO(3)$ and the position on $\mathbb{R}^2$. Hence the goal space differs from the workspace and configuration space and is given by either $SO(3) \times \mathbb{R}^2$ or $S^2 \times \mathbb{R}^2$.

Workspace	Reduced Configuration space	Goal space
$SE(3) \times SE(3)$ $\longrightarrow$	$SO(3) \times SO(3) \times \mathbb{R}^2$	$SO(3) \times \mathbb{R}^2$ $S^2 \times \mathbb{R}^2$
12 DOF	8 DOF	5 or 4 DOF

Figure 1-1: Different spaces of the system

1-2 Alten Mechatronics

The thesis is done in collaboration with Alten Mechatronics in Eindhoven. At Alten a ball-balancing robot is currently in development with the purpose to be an eye-catcher and demonstrator at technical events and gatherings. This section will give a brief introduction to the current state of the robot.

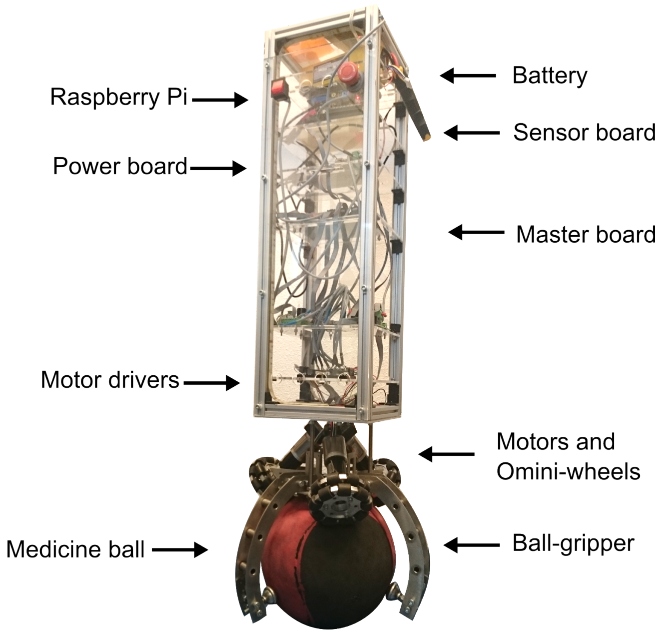


Figure 1-2: The ball balancing robot of Alten Mechatronics

In Figure 1-2 the current robot is shown, along with its primary components. The battery supplies the power to the robot, which is distributed via the power board. Measurements on the states of the body are provided by the sensor board, which is equipped with an Inertia Measurement Unit (IMU). The IMU consists of a accelerometer, magnetometer and gyroscope that measure the linear acceleration, magnetic field and angular velocity respectively at a sample rate of 200 Hz. Furthermore, the motors are equipped with encoders to measure the angles of the omni-wheels.

The desired control input is computed on the master board, which sends either a desired torque to the motor drivers. These drives subsequently compute and track a desired current.

The ball is actuated by the means of three double-row omni-wheels. The ball is a 3kg medicine ball and is chosen for its relative low price and relative smooth surface and stiffness as compared to other commercially available balls.

The current implemented controller was part of a preceding thesis [7]. This thesis will be a continuation and extension of previous work.

Both a LQR controller and a SISO controller were proposed to control both position and balancing of the body. For initial tests, the LQR controller was implemented for balancing only, but the system was not able to stabilise. Instead a manually tuned PD controller was implemented for balancing. Position control was left as future work. The PD controller is able to balance the robot for a couple of seconds, but does not robustly stabilise the body. In [7] the cause of the unstable balancing was argued to be caused by mechanical vibrations due to the double row omni-wheels and the lack of proper filtering.

1-3 Problem statement

The thesis consists of a theoretical part and practical part.

The theoretical part can be split into the derivation of a 3D model, the design of a controller and the design of an observer. The overall goal is to use a coordinate-free, i.e. geometric approach. To the best knowledge of the author a geometric approach is novel for ball-balancing robots.

For the modelling of the robot, a coordinate-free model will be derived using screw-theory. This results in a model expressed in rotation matrices, angular velocities, translations and linear velocities.

The traditional Euclidean error will not suffice to control these quantities. Hence a geometric nonlinear control will be derived using error functions for the underlying differential manifolds. The designed geometric algorithm should be able to control both the attitude (rotations) and position of the robot. More specifically, the designed controller should be able to:

1. Stabilise the attitude robot
2. Stabilise the robot at a certain position
3. Track predefined references

Furthermore, the use case of the robot will be events and demonstrations. Here it will be likely that the robot is accidentally pushed. Moreover the floor type will be unknown, resulting in unknown friction. Moreover, optional attachments to the robot might result in a change of inertial properties of the robot such as its mass. Hence it is desired if the control algorithm is:

4. Capable of disturbance rejection
5. Robust to model uncertainties

For the observer design again the error functions for the differential manifolds will be used to propose an almost global geometric observer.

The practical part of this work consists of the implementation of the proposed algorithms. Since the robot in current state is not able to balance yet, first a more traditional LQR controller and Kalman filter will be implemented in order to stabilise the attitude of the robot.

1-4 Structure of the MSc thesis

The thesis is organised as follows:

Chapter 2: Background This introductory chapter will briefly review previous work on ball-balancing robots.

Chapter 3: Mathematical preliminaries This chapter covers the basic mathematical preliminaries used throughout the thesis

Chapter 4: Modelling of the robot This chapter describes the derivation of a full 3D model using screw-theory.

Chapter 5: Geometric nonlinear control In this chapter a nonlinear geometric approach is proposed to control the attitude and position of the robot. The effectiveness of the proposed controllers is shown in simulation.

Chapter 6: Geometric observer In this chapter a geometric nonlinear observer is designed to estimate the states of the system. The effectiveness will again be shown using simulation results.

Chapter 7: Implementation In this chapter the implementation of the linear algorithm is described.

Chapter 8: Conclusion and recommendations This MSc thesis is concluded with conclusions and recommendations

Chapter 2

Background

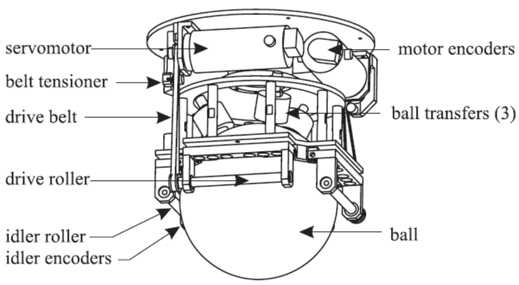
This chapter covers the ballbots as found in literature. Firstly an overview of the most well-known ballbots will be given, followed by the modelling and control of these robots. If the reader is solely interested in the contributions of this work, it is advised to skip this chapter.

2-1 Brief history of ball-balancing robots

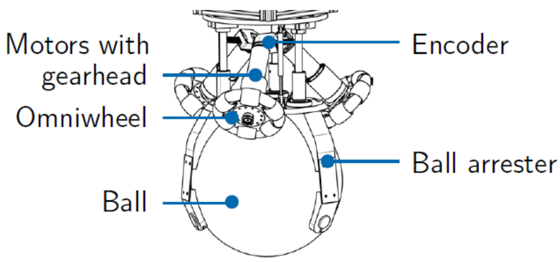
In this section the most well-known ball-balancing robots in literature are introduced. Mainly two ballbot variants are known with respect the actuation mechanism; the robot is actuated by using either an inverted mouse-ball drive or three omni-wheels, see Figure 2-1.

The four best publishing and/or well-known institutes in the field of ballbots are Carnegie Mellon University (CMU) with ‘ballbot’, Tohoku Gakuin University (TGU) with BallIp, Eidgenössische Technische Hochshule Zürich (ETH) with Rezero and finally the National Chung Hsing University (NCHU) with two unnamed ballbots.

Carnegie Mellon University One of the first successful ball-balancing robots was developed by the Carnegie Mellon University [8], see Figure 2-2a. This robot uses an

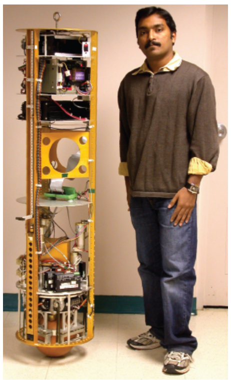


(a) Inverted mouse-ball drive [8]

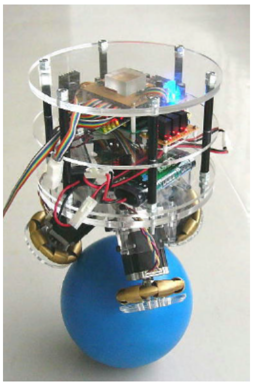


(b) 3 omni-wheels actuation [5]

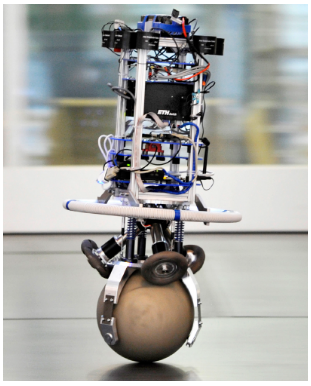
Figure 2-1: Actuation variants



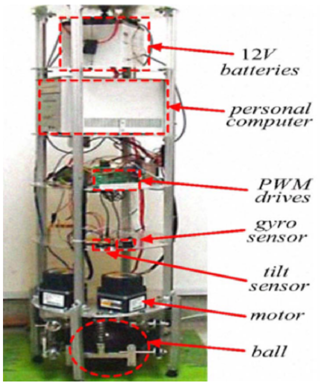
(a) Ballbot from CMU [1]



(b) BallIP from TGU [2]



(c) Rezero from ETH [3]



(d) Ballbot from NCHU [9]

(e) Second ballbot from NCHU [10]

Figure 2-2: Overview of ball balancing robot

inverted mouse-ball drive to balance on top of a ball, see Figure 2-1a. An evolved version was later introduced which included a yaw drive mechanism that can reorient the body with respect to the driving mechanism [1]. Furthermore, the robot is also equipped with three retractable legs, making it able to be statically stable. In later iterations also a variant with two arms was developed [11]. An IMU provides Kalman-filtered body angles and rates (w.r.t. gravity). Finally, the four motors that drive the ball are equipped with encoders to measure the rotation of the ball.

Tohoku Gakuin University In 2008 the Tohoku Gakuin University presented BallIP [12], shown in Figure 2-2b. As opposed to the ballbot of CMU, the ball is driven by three omni-wheels, each one actuated by stepping motors without gearboxes. Furthermore the robot is equipped with two sets of gyroscopes and accelerometers, which provide the pitch angles by combining both signals using a first-order digital filter. The angular velocities are directly obtained from the gyroscopes.

Eidgenössische Technische Hochshule Zürich Similar to BallIP, Rezero from Eidgenöss-ische Technische Hochshule Zürich also has three omni-wheels, driven by DC brushless motors with gear heads, each having their own encoder, see Figure 2-2c. Furthermore the Rezero is equipped with an IMU, which measures the three Euler angles and angular rates of the body. The ball is a hollow aluminium sphere coated with high stiction synthetic. Ball arresters are used to push the ball against the omni-wheels, in order to increase the contact force and thus minimising slip [5].

National Chung Hsing University At the National Chung Hsing University both variant of the ballbots were developed. The first robot has an inverted mouse-ball drive [9, 13–15], similar to the ballbot of CMU, as shown in Figure 2-2d. The second robot is actuated by three omni-wheels [10,16], similar to Rezero and BallIP, as shown in Figure 2-2e

The robots are equipped with a tilt sensor, gyroscope and the motors are equipped with an encoder.

2-2 Modelling

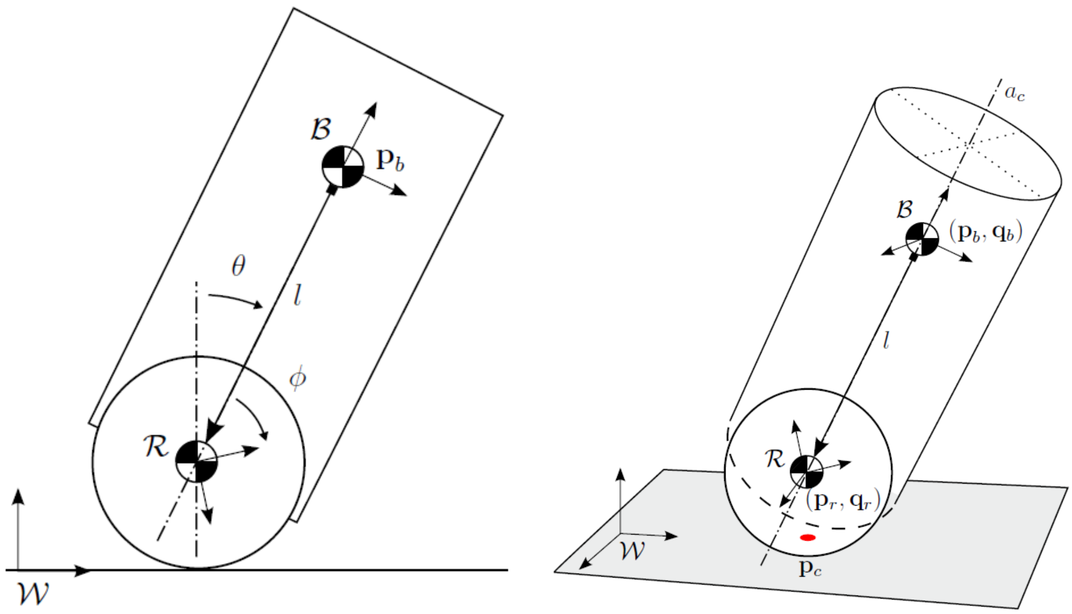
In this section the different approaches to model a ballbot tried in literature is reviewed. The vertical xz plane is also referred to as the sagittal plane and the vertical yz plane as the coronal plane. The inclination angle (for both the sagittal and coronal planes) is defined as the angle between the body and the z-axis. The inclination angles are often referred to as the pitch and roll angles. The yaw angle is defined as the rotation about the z-axis in body coordinates.

Common assumptions for the modelling of the ballbot are:

- The robot is simplified to a rigid cylinder and the ball to a rigid sphere.
- There is no slip between the ball and floor and the ball and drive mechanism.
- The dynamics in the sagittal and coronal plane are identical.
- The ball does not rotate around the z-axis.
- The ball does not deform.
- The actuator dynamics are fast, i.e. the input has no delay.
- The floor is flat and level.
- The wheels are always in contact with the ball.

For 2D models also the assumption is made that the dynamics of the two planes are independent and therefore the system can be decoupled as two independent planar models; CMU: [1,8], TGU: [2,12], ETH: [5], NCHU: [9,10,15,16]. This assumption is based on the fact that the coupling terms between the vertical planes contain sine products. Therefore, if the system is only subjected to small pitch angles, the coupling terms are negligible and therefore the planar dynamics can be decoupled [1].

Besides the two decoupled 2D models, also 3D models have been proposed; CMU: [11], ETH: [5] and NCHU: [13,14]. A 3D model is also proposed in [6] and [17].



(a) 2D model [6] **(b)** 3D model [6]

Figure 2-3: Models of a ballbot

A visualisation of 2D and 3D models for a ballbot are shown in Figure 2-3a and 2-3b respectively [6]. The main advantage of two identical decoupled 2D models is its relative simplicity. However, it has the following limitations and disadvantages. First of all, coupling terms are neglected. These effects become especially significant at higher velocities. [5, 7]. Secondly, the natural yaw dynamics are completely neglected [6]. Finally, addition of arms or asymmetric loads would make a 2D model even less accurate [6]. However, the cost of a full 3D model is the added complexity and the no-slip condition results in a nonholonomic constraint, as opposed to a holonomic constraint for the 2D model.

In general, the Euler-Lagrange equations are used to derive the equations of motion. However, the choice of general coordinates often varies between the different ballbots. This section is devoted to those differences. Unless mentioned otherwise, Lagrangian mechanics are used for the dynamic equations.

The model of the ballbot of CMU has the tilt angle ϕ and the angular ball configuration θ as the generalised coordinates. The angular ball configuration is chosen such that the horizontal position of the ball w.r.t. the world frame is given by $x = r(\theta + \phi)$, where r is the radius of the ball. The advantage of choosing the ball configuration coordinate θ like this, is that it directly corresponds to the encoder readings of the ball drive. Euler-Lagrange equations are used to derive the dynamic equations. Friction was initially modelled as only viscous friction, in order to prevent discontinuous dynamics caused by Coulomb friction [8]. However, in the improved work also Coulomb friction was introduced [1].

For the modelling of Rezero both a planar 2D model and a 3D model were proposed [5]. As an additional assumption friction is neglected, as non-continuous (Coulomb) friction would result in more complex equations. The 3D model consists of the ball, 3 omni-wheels and the body including the motors and gear heads. The generalised coordinates are the integral of the angular velocity of the ball, i.e. the travelled angular position, and three Euler angles the spatial orientation angles of the body. These coordinates are chosen in such a way that they are directly measurable by the systems sensors.

The modelling for the two different ballbots of NCHU is similar. However, there are some differences. For the robot with 3 omni-wheels, the general coordinates are the tilt angle and the motors' angular positions for both vertical planes [10, 16]. For the inverted mouse-ball drive the general coordinates are the tilt angle of the robot and the rotating angles of of the ball for both vertical planes [9, 15].

For the 3D models of [13] and [14], the orientation of the cylinder is described with a pitch and yaw angle and its planar position.

As an exception, in [14] Newtonian mechanics are used to derive a model. The final equations of motion are expressed in the nutation and procession angles and the longitudinal and lateral displacement in the moving frame.

For all work of NCHU, except [13], friction is modelled as a sum of Coulomb and viscous friction between the ball and ground.

Inal et all. [6] proposed a full 3D model, as they reasoned that a 2D model does not capture yaw motion arising from the wheel rolling motion and coupled inertial effects. Furthermore it is found that for a 2D model the no-slip condition results in holonomic constraint, while for the 3D model this constraint is nonholonomic.

Three coordinate frames are used: the inertial world frame, a body frame and ball frame. Instead of Euler-angles, quaternions where used to represent the rotation. It is assumed that the relative yaw rotation and the ball is prevented by a constraint friction torque that is not directly controlled. Furthermore it is assumed that friction between the ball and the ground prevent it from slipping in horizontal and yaw direction. As opposed to other models, quaternions are used to represent the rotations of the ball and robot. The equations of motion are derived using force balancing and constraints on the dynamics.

In Lotfiani et all. [17] a full 3D model was proposed as well. Again three frames are used: one for the inertial frame, one for the body and one the sphere. The generalised coordinates are two sets of three consecutive rotations, one set for both the body and sphere, resulting in 6 generalised coordinates in total. An additional assumption is introduced that the resultant moment around the longitudinal axis of the cylinder is always zero. This yields two nonholonomic constraints. Furthermore viscous friction is modelled between the idle rollers and the sphere. The system is then modelled using the Euler-Lagrange equations and Lagrange multipliers. The Lagrange multipliers are eliminated using the null space of the constraint matrix A.

2-3 Control

In this section the proposed control methods for ballbots are reviewed

CMU The initially proposed control consisted a two-loop controller for each vertical plane, consisting of a PI controller as inner loop and LQR controller as outer loop [8]. These planar controllers are decoupled and independent of each other. The output of the outer loop is a desired velocity of the ball, which is then tracked by the inner loop. The purpose of the inner loop is to reduce the effect of the unmodelled static and dynamic friction, while the outer loop is used to track a given reference.

In [1] an improved control scheme is presented, again consisting of two independent two-loop controllers for each plane. The inner loop is used for balancing and the outer loop for position control. However, using this controller for trajectory tracking results in a jerky motion. It was reasoned to be because the inner and outer loop fight each other [8], as the system is under-actuated. Therefore an offline trajectory planner is proposed [1]. This controller uses the natural dynamics for desired point-to-point motion.

To improve tracking performance the shape-space planner is proposed in [11]. The shape space indicates the variables influencing the mass matrix of the system and hence have a large influence on the dynamics of the system, as opposed to position variables. As opposed to the trajectory planner described in [1], this trajectory planning algorithm is fast enough to run in real-time on the robot. This is because this shape-space planner only uses a subset of the equation of motion, namely the dynamic constraint equations. Furthermore, this also results in tracking that is more robust to actuator and friction uncertainties. The shape-space planner returns desired body angles, given a certain desired acceleration of the position variables. The resulting desired body angles are tracked using the PID balancing controller.

The shape trajectory planner was found to be faster than collocation methods, as the optimisation is performed on a smaller parameter space and only the dynamic constraints are used. On the other hand, it was found that direct collocation methods perform better when the state trajectories have good initial guesses, as they improved the configuration trajectories and achieved lower tracking errors. The shape space planner can be used to obtain such an initial guess. Furthermore, direct collocation methods can be applied for dynamics systems in general, whereas the shape trajectory planner is limited to shape-accelerated balancing systems.

TGU The control of the BallIP robot consists of two independent controllers for the two vertical planes. Each controller is formed by a summation of two PD controllers, one for the pitch angle and the one for the position [2, 12].

Rezero For the control of Rezero a linear full state LQR controller was used. To improve the tracking performance of non-zero setpoints a feed-forward term was used [5].

A nonlinear velocity controller was proposed using gain scheduling, i.e. designing multiple linear controllers at different operating points and interpolating between them.

Due to the high dimensionality of the system and the limited memory and computational power, only the non-linearities caused by velocity are considered, i.e. linearising around setpoints with different velocities. In contrast to the linear controller, this controller is able to stably control the system at higher velocities (over 2 m/s), as at these velocities also the coupling terms have a more significant influence. As a result the nonlinear controller is better able to track complicated sequences as compared to the linear controller.

In 2013, an extended Kalman filter (EKF) to estimate the states of the Rezero [3] was proposed. It uses the data from the accelerators, gyroscope and encoders to estimate the complete state of the robots, i.e. attitude, position and velocity. This is in contrast with the robots BallIP, ballbot of CMU and the robots of the National Chung Hsing University, where the attitude and position states are estimated separately.

The states of the EKF are the position of the IMU, its velocity and the rotation matrix of the body, all expressed in the body frame. Discrete prediction equations are derived, which relate the sensor outputs to the states of the EKF. For the update step of the EKF, a virtual measurement of the ball velocity is used for the innovation, since for a simple point contact model between the ball and omni-wheel. The corresponding innovation is hence defined as the difference between the equation relying on the filtered states and the equation relying on the readings from the omni-wheel sensors.

It was proven that both the absolute position and the rotation around the gravity axis are unobservable. However, drift in these states does not influence the other predicted states.

NCHU Throughout the work at the NCHU, the proposed controllers are backstepping sliding-mode control without [13] and with parametric uncertainties [9, 15, 16], LQR control [10] and a multi-loop control approach [14].

In [14] a double two-loop controller was proposed for self-balancing and position control, one two-loop controlling the pitch and forward position, the other two-loop the jaw velocity and the latitudinal movement. Each controller has an inner-loop consisting of a PI controller and an outer-loop with an LQR controller, similar to [8]. An additional feed-forward term was used to overcome friction.

In [10] a single LQR controller was proposed. The control for position keeping and balancing is combined into one regulator problem with as states the tilt angle, its velocity, the error between current position and desired position and its derivative.

In [13] a backstepping hierarchical sliding-mode controller is proposed, consisting of an inner and outer layer. The inner layer has a sliding function for the pitch angle and for the heading direction. The outer layer is defined as a convex combination of the two inner sliding surfaces.

In [9, 15, 16] the backstepping sliding-mode controller was augmented with a recurrent interval type-2 neural network (RIT2FNN) in order to be able to deal with uncertainties.

To make the auxiliary backstepping errors converge to zero, the procedure of a hierarchical aggregated sliding-mode controller is used, as introduced by [18].

Other In [19] a fuzzy controller was proposed for a ballbot. Here a two-loop is used, similar to the two-loop in [1, 8]: an outer loop puts out an desired angle to the inner loop, which returns the input for the system. Here each loop controller consists of a fuzzy controller with 25 rules each.

In [17] both a computed torque and sliding mode controller were proposed for the balancing control and verified by the means of simulation. The under-actuation of the system was dealt with by the means of a fuzzy trajectory planner. This fuzzy controller uses the angular angles of the sphere and its derivatives as input and returns the desired angles for the cylinder. These desired angles are used as the control input for the computed torque and sliding mode controller.

Chapter 3

Mathematical preliminaries

In this chapter the mathematical preliminaries will be treated.

3-1 Screw theory

In this section basic concepts of screw theory and their notation used in this work are introduced. Readers unfamiliar with these concepts are referred to [20] for a more in depth treatment of this material.

Tilde operator First let us define the tilde operator. The vector product of two vectors $a, b \in \mathbb{R}^3$, can also be expressed by a multiplication with a skew symmetric matrix:

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \quad \tilde{a} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{3-1}$$

$$a \times b = \tilde{a} b \tag{3-2}$$

Here $\tilde{a}$ is referred to as the tilde form of a. The inverse of the tilde operator is given by the vex operator $\cdot^{\vee}$:

$$\tilde{a}^{\vee} = a \tag{3-3}$$

Furthermore, the following useful properties will be used throughout this work:

$$\tilde{a}^T = -\tilde{a} \tag{3-4}$$

$$\tilde{a} b = -\tilde{b} a \tag{3-5}$$

$$-\frac{1}{2}\mathrm{tr}(\tilde{a}\tilde{b}) = \frac{1}{2} a^T b \tag{3-6}$$

Here tr() denotes the trace function:

$$\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}$$

for $A \in R^{n \times n}$

Frames, rotation matrices and homogeneous matrices To express the orientation and position of bodies in space, different coordinate frames will be used. A *frame* is denoted by Ψ_i and consists of three orthonormal vectors $x, y, z \in \mathbb{R}^3$. Frames are related to each other by a rotation and translation.

The rotation between frame Ψ_a and Ψ_b is denoted by the *rotation matrix* R_a^b. The rotation matrix belongs the the special orthogonal group $SO(3)$

$$SO(3) := \left\{R \in \mathbb{R}^{3\times3}; RR^T = I, \det R = +1\right\} \tag{3-7}$$

Note that the inverse of the rotation matrix is equal to its transpose:

$$\left(R_\mathrm{a}^\mathrm{b}\right)^{-1} = \left(R_\mathrm{a}^\mathrm{b}\right)^T = R_\mathrm{b}^\mathrm{a}$$

A rotation matrix R_1^n can be found by using subsequent matrix multiplication of rotation matrices of intermediate frames:

$$R_1^\mathrm{n} = R_{\mathrm{n}-1}^\mathrm{n} \dots R_2^3 R_1^2 \tag{3-8}$$

This matrix multiplication gives the rotation from frame 1 to 2, followed by the rotation from frame 2 to 3 and so on.

The Lie algebra of the rotation matrix $R \in SO(3)$ is given by $\dot{R}R^T$ and $R^T\dot{R}$, which are skew-symmetric matrices and belong to $so(3)$.

$$so(3) := \left\{\tilde{\omega} \in \mathbb{R}^{3\times3}; \tilde{\omega}^T = -\tilde{\omega}\right\} \tag{3-9}$$

Here ω is the instantaneous angular velocity. The angular velocity $\omega_\mathrm{a}^{\mathrm{c},\mathrm{b}} \in \mathbb{R}^3$ represents the angular velocity of Ψ_a with respect to Ψ_b expressed in frame Ψ_c and is defined by

$$\tilde{\omega}_\mathrm{b}^{\mathrm{b},\mathrm{a}} = R_\mathrm{a}^\mathrm{b}\dot{R}_\mathrm{b}^\mathrm{a} \tag{3-10}$$
$$\tilde{\omega}_\mathrm{b}^{\mathrm{a},\mathrm{a}} = \dot{R}_\mathrm{b}^\mathrm{a}R_\mathrm{a}^\mathrm{b} \tag{3-11}$$

The angular velocity can be rewritten in a different frame by using the relation

$$\omega_\mathrm{j}^{\mathrm{b},\mathrm{k}} = R_\mathrm{a}^\mathrm{b}\omega_\mathrm{j}^{\mathrm{a},\mathrm{k}} \tag{3-12}$$

Furthermore, let us define the following identity:

$$R\tilde{a}R^T = (Ra)^\sim \tag{3-13}$$

The *translation* between frame Ψ_a and Ψ_b is given by the translation vector $p_\mathrm{a}^\mathrm{b} \in \mathbb{R}^3$. Using the concepts of rotation and translation, it is possible to define a *homogeneous matrix*, which is used to translate between two frames. The homogeneous matrix is denoted by H_a^b and denotes the transformation from frame Ψ_a to Ψ_b and is given by

$$H_\mathrm{a}^\mathrm{b} = \begin{bmatrix} R_\mathrm{a}^\mathrm{b} & p_\mathrm{a}^\mathrm{b} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} \tag{3-14}$$

and its inverse is given by

$$\left(H_{\mathrm{a}}^{\mathrm{b}}\right)^{-1} = H_{\mathrm{b}}^{\mathrm{a}} = \begin{bmatrix} \left(R_{\mathrm{a}}^{\mathrm{b}}\right)^{T} & -\left(R_{\mathrm{a}}^{\mathrm{b}}\right)^{T} p_{\mathrm{a}}^{\mathrm{b}} \\ 0 & 1 \end{bmatrix} \tag{3-15}$$

Note that the inverse of the homogeneous matrix can be expressed using only transposes.

Homogeneous matrices are part of the special Euclidean group $SE(3)$:

$$SE(3) := \left\{ \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \text{ s.t. } R \in SO(3),\ p \in \mathbb{R}^3 \right\} \tag{3-16}$$

A homogeneous matrix H_1^{n} can be found by using subsequent matrix multiplication of homogeneous matrices of intermediate frames:

$$H_1^{\mathrm{n}} = H_{\mathrm{n}-1}^{\mathrm{n}} \dots H_2^3 H_1^2 \tag{3-17}$$

This matrix multiplication gives the transformation from frame 1 to 2, followed by the transformation from frame 2 to 3 and so on.

A point $p^{\mathrm{a}} \in \mathbb{R}^3$ expressed in fame Ψ_{a} can be expressed in frame Ψ_{b} by using the following relation

$$\begin{bmatrix} p^b \\ 1 \end{bmatrix} = H_{\mathrm{a}}^{\mathrm{b}} \begin{bmatrix} p^a \\ 1 \end{bmatrix} \tag{3-18}$$

Note that the points p^{i}, with $\mathrm{i} = \{a, b\}$ are augmented with 1. In the remainder of this work this augmented vector will be denoted by P^{i}, where the superscript denotes the frame. The homogeneous matrix forms the transformation matrix between frames and can be seen as the kinematic relations between different frames.

The Lie algebra of $H \in SE(3)$ is given by $\dot{H}H^{-1}$ and $H^{-1}\dot{H}$ belong to $se(3)$

$$se(3) := \left\{ \begin{bmatrix} \tilde{\omega} & v \\ 0 & 0 \end{bmatrix} \text{ s.t. } \tilde{\omega} \in so(3),\ v \in \mathbb{R}^3 \right\} \tag{3-19}$$

Here $v_{\mathrm{a}}^{\mathrm{c,b}}$ is the *linear velocity* between velocity of Ψ_{a} with respect to Ψ_{b} expressed in frame Ψ_{c}. The Lie algebra is defined as a *twist*, which are the generalisation of velocities for a rigid body. The twist $T_{\mathrm{a}}^{\mathrm{c,b}}$ expresses linear and angular velocity of Ψ_{a} with respect to Ψ_{b} expressed in frame Ψ_{c} and is defined by

$$\tilde{T}_{\mathrm{b}}^{\mathrm{b,a}} = H_{\mathrm{a}}^{\mathrm{b}} \dot{H}_{\mathrm{b}}^{\mathrm{a}} \tag{3-20}$$

$$\tilde{T}_{\mathrm{b}}^{\mathrm{a,a}} = \dot{H}_{\mathrm{b}}^{\mathrm{a}} H_{\mathrm{a}}^{\mathrm{b}} \tag{3-21}$$

and has the following structure

$$T = \begin{bmatrix} \omega \\ v \end{bmatrix} \in \mathbb{R}^6, \qquad \tilde{T} = \begin{bmatrix} \tilde{\omega} & v \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

Twists can be rewritten in different frames using the adjoint matrix defined by

$$Ad_{H_{\mathrm{a}}^{\mathrm{b}}} = \begin{bmatrix} R_{\mathrm{a}}^{\mathrm{b}} & 0 \\ \tilde{p}_{\mathrm{a}}^{\mathrm{b}} R_{\mathrm{a}}^{\mathrm{b}} & R_{\mathrm{a}}^{\mathrm{b}} \end{bmatrix} \tag{3-22}$$

using

$$T_{\mathrm{j}}^{\mathrm{b,k}} = Ad_{H_{\mathrm{a}}^{\mathrm{b}}} T_{\mathrm{j}}^{\mathrm{a,k}} \tag{3-23}$$

Wrenches Previously twists were defined, which are the generalisations of velocities. Now *wrenches* will be defined, which can be seen as the generalisation of force. The wrench $W^{\mathrm{b,a}} \in \mathbb{R}^6$ denotes a wrench acting on frame Ψ_{a} expressed in frame Ψ_{b} and is given by

$$W^{\mathrm{b,a}} = \begin{bmatrix} \tau^{\mathrm{b,a}} & f^{\mathrm{b,a}} \end{bmatrix} \tag{3-24}$$

where $\tau^{\mathrm{b,a}} \in \mathbb{R}^3$ denotes the torque and $f^{\mathrm{b,a}} \in \mathbb{R}^3$ the force, both acting on frame Ψ_{a} expressed in frame Ψ_{b}. Note that wrenches are covectors, hence are represented using row vectors. The wrench can be expressed in a different frame by using the relation

$$\left(W^{\mathrm{b,j}}\right)^T = Ad^T_{H^{\mathrm{a}}_{\mathrm{b}}} \left(W^{\mathrm{a,j}}\right)^T \tag{3-25}$$

The transposed adjoint matrix is given by

$$Ad^T_{H^{\mathrm{a}}_{\mathrm{b}}} = \begin{bmatrix} R^{\mathrm{b}}_{\mathrm{a}} & -R^{\mathrm{b}}_{\mathrm{a}}\tilde{p}^{\mathrm{a}}_{\mathrm{b}} \\ 0 & R^{\mathrm{b}}_{\mathrm{a}} \end{bmatrix} \tag{3-26}$$

Flattened rotation matrix The flattened rotation matrix is denoted by $\bar{R}^{\mathrm{b}}_{\mathrm{a}}$ and has the following structure:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{3-27}$$

$$\bar{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{21} & r_{22} & r_{23} & r_{31} & r_{32} & r_{33} \end{bmatrix}^T \tag{3-28}$$

The derivative of the flattened matrix $\dot{\bar{R}}^{\mathrm{a}}_{\mathrm{b}}$ is related to the angular velocity $\omega^{\mathrm{b,a}}_{\mathrm{b}} \in \mathbb{R}^3$ through the map $\Phi(R^{\mathrm{a}}_{\mathrm{b}})$:

$$\dot{\bar{R}}^{\mathrm{a}}_{\mathrm{b}} = \Phi(R^{\mathrm{a}}_{\mathrm{b}})\omega^{\mathrm{b,a}}_{\mathrm{b}} \tag{3-29}$$

where the map $\Phi(R^{\mathrm{a}}_{\mathrm{b}})$ is defined as

$$\Phi(R^{\mathrm{a}}_{\mathrm{b}}) = \left[\begin{array}{ccc|ccc|ccc} 0 & r_{13} & -r_{12} & 0 & r_{23} & -r_{22} & 0 & r_{33} & -r_{32} \\ -r_{13} & 0 & r_{11} & -r_{23} & 0 & r_{21} & -r_{33} & 0 & r_{31} \\ r_{12} & -r_{11} & 0 & r_{22} & -r_{21} & 0 & r_{32} & -r_{31} & 0 \end{array}\right]^T \tag{3-30}$$

Proof. Consider the first row of $\dot{R}^{\mathrm{a}}_{\mathrm{b}}$, i.e. $\dot{R}_1 = \begin{bmatrix} \dot{r}_{11} & \dot{r}_{12} & \dot{r}_{13} \end{bmatrix}$. This vector is computed by

$$\dot{R}^T_1 = R_1\tilde{\omega}^{\mathrm{b,a}}_{\mathrm{b}} \tag{3-31}$$

where R_1 denotes the first row of $R^{\mathrm{a}}_{\mathrm{b}}$. This can be rewritten using the cross product property $\tilde{a}b = -\tilde{b}a$ as:

$$\dot{R}^T_1 = -\tilde{R}_1\omega^{\mathrm{b,a}}_{\mathrm{b}} \tag{3-32}$$

Repeating the same steps for the second and third row of $\dot{R}^{\mathrm{a}}_{\mathrm{b}}$ and then concatenating the results in a vector yields:

$$\underbrace{\begin{bmatrix} \dot{R}^T_1 \\ \dot{R}^T_2 \\ \dot{R}^T_3 \end{bmatrix}}_{\dot{\bar{R}}^0_{\mathrm{s}}} = -\underbrace{\begin{bmatrix} \tilde{R}_1 \\ \tilde{R}_2 \\ \tilde{R}_3 \end{bmatrix}}_{\Phi(R^0_{\mathrm{s}})} \omega^{\mathrm{s,0}}_{\mathrm{s}} \tag{3-33}$$

$\square$

3-2 Theorems

Now let us define stability theorems that will be used throughout this work. These theorems originate from [21].

Theorem 1 (Lyapunov stability theorem)**.** *Let $x = 0$ be an equilibrium point for $\dot{x} = f(x)$ and $\mathcal{Q} \subset \mathbb{R}^n$ be a domain containing $x = 0$. Let $V : \mathcal{Q} \to \mathbb{R}$ be a continuously differentiable function, such that*

$$\begin{aligned} &V(0) = 0 \\ &V(x) > 0 \textit{ in } \mathcal{Q}/\{0\} \end{aligned}$$

Then, $x = 0$ is stable. Moreover, if

$$\dot{V}(x) < 0 \textit{ in } \mathcal{Q}/\{0\}$$

then $x = 0$ is locally asymptotically stable. Furthermore,if $\mathcal{Q} = \mathbb{R}^n$ and

$$\begin{aligned} &V(0) = 0 \\ &V(x) > 0 \textit{ in } \mathcal{Q}/\{0\} \\ &\dot{V}(x) < 0 \textit{ in } \mathcal{Q}/\{0\} \\ &V(x) \to \infty \textit{ if } \|x\| \to \infty \end{aligned}$$

the equilibrium $x = 0$ is globally asymptotically stable

In order to prove stability, sometimes LaSalle's invariant principle will be needed to finalise the proof:

Theorem 2 (LaSalle's invariant principle)**.** *Let $\Omega \subset \mathcal{Q}$ be a compact set that is positively invariant with respect to $\dot{x} = f(x)$. Let $V : \mathcal{Q} \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(x) \leq 0$ in Ω. Let $\mathcal{E}$ be a set of all points in Ω where $\dot{V} = 0$. Let $\mathcal{M}$ be the largest invariant set in $\mathcal{E}$. Then every solution starting in Ω approaches $\mathcal{M}$ as $t \to \infty$.*

3-3 Error functions

In this section error functions on $SO(3)$ will be defined that will be used throughout this work. These error functions originate from [21, 22]

First of all, the rotation matrix R_e denotes the relative rotation between a frame Ψ_{a} and the desired frame Ψ_{d}:

$$R_e = R_{\text{a}}^{\text{d}} = R_0^{\text{d}} R_{\text{a}}^0$$

For ease of notation and readability, let us use R to indicate R_{a}^0 and R_d to indicate R_{d}^0. Hence:

$$R_e = R_d^T R \tag{3-34}$$

Now define the error function as

$$\phi(R_e) = \frac{1}{2}\mathrm{tr}\left(I - R_e\right) \tag{3-35}$$

This function has a minimum for $R_e = I$, which corresponds to $R_\mathrm{i}^0 = R_\mathrm{d}^0$, and a maximum for $R_e = -I$. The function $\phi(R_e)$ is therefore bounded by $0 \leq \phi(R_e) \leq 3$.

The derivative of $\phi(R_e)$ is given by

$$\begin{aligned}\frac{d}{dt}\phi(R_e) &= -\frac{1}{2}\mathrm{tr}\left(\dot{R}_e\right)\\ &= -\frac{1}{2}\mathrm{tr}\left(R_e\tilde{\omega}_e\right)\end{aligned}$$

Now using the property $-\frac{1}{2}\mathrm{tr}(\tilde{a}\tilde{b}) = \frac{1}{2}a^Tb$:

$$= \frac{1}{2}\left(\left(R_e - R_e^T\right)^\vee\right)^T\omega_e$$

From this the attitude tracking error $e_R \in \mathbb{R}^3$ chosen to be :

$$\begin{aligned}e_R &= \frac{1}{2}\left(R_e - R_e^T\right)^\vee\\ e_R &= \frac{1}{2}\left(R_d^TR - R^TR_d\right)^\vee\end{aligned} \tag{3-36}$$

Now let us derive the error function for the angular velocity e_ω corresponding to the error rotation matrix, expressed in frame Ψ_a.

$$\tilde{e}_\omega = \tilde{\omega}_\mathrm{a}^{\mathrm{a,d}} = R_e^T\dot{R}_e \tag{3-37}$$

Let us denote the shorthand of the desired angular velocity $\omega_\mathrm{d}^{\mathrm{d},0}$ as ω_d. Using the relation $\dot{R}_e = R_e\tilde{e}_\omega$:

$$\begin{aligned}\frac{d}{dt}\left(R_d^TR\right) &= R_d^TR\tilde{e}_\omega\\ \dot{R}_d^TR + R_d^T\dot{R} &= R_d^TR\tilde{e}_\omega\\ \tilde{\omega}_d^TR_d^TR + R_d^TR\tilde{\omega} &= R_d^TR\tilde{e}_\omega\end{aligned}$$

Pre-multiplying with $R_e^T = \left(R_d^TR\right)^T$ and using the identity $R\tilde{\omega}R^T = (R\omega)^\sim$:

$$\begin{aligned}\tilde{e}_\omega &= \left(R_e^T\omega_d^T\right)^\sim + \tilde{\omega}\\ e_\omega &= \omega - R_e^T\omega_d\\ e_\omega &= \omega - R^TR_d\omega_d\end{aligned} \tag{3-38}$$

The above angular velocity error is expressed in frame Ψ_a:

$$e_\omega = \omega_\mathrm{a}^{\mathrm{a,d}} = \omega_\mathrm{a}^{\mathrm{a},0} - R_0^\mathrm{a}R_\mathrm{d}^0\omega_\mathrm{d}^{\mathrm{d},0}$$

To express the angular velocity error in inertia frame, the expression is pre-multiplied with R_a^0:

$$\omega_\mathrm{a}^{0,\mathrm{d}} = \omega_\mathrm{a}^{0,0} - \omega_\mathrm{d}^{0,0} \tag{3-39}$$

Summarising the above results, the error functions for $SO(3)$ are defined as:

$$e_R = \frac{1}{2}\left(R_d^T R - R^T R_d\right)^\vee$$
$$e_\omega^{\mathrm{a}} = \omega_{\mathrm{a}}^{\mathrm{a},0} - R_0^{\mathrm{a}} R_{\mathrm{d}}^0 \omega_{\mathrm{d}}^{\mathrm{d},0}$$
$$e_\omega^0 = \omega_{\mathrm{a}}^{0,0} - \omega_{\mathrm{d}}^{0,0}$$

where e_ω^{a} and e_ω^0 are expressed in frame Ψ_{a} or inertia frame Ψ_0 respectively.

Chapter 4

Modelling of the robot

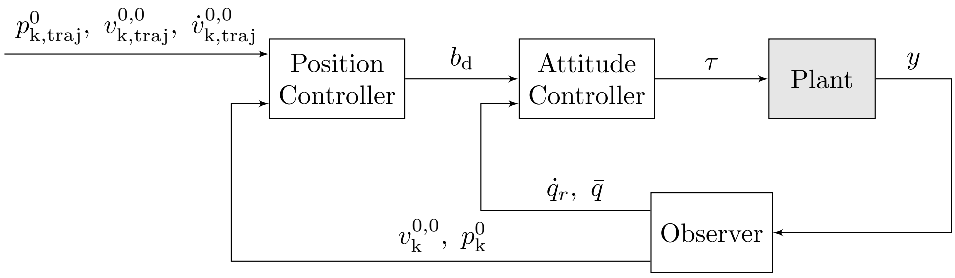


Figure 4-1: Controller structure

In this chapter the full 3D model of the robot will be derived with the help of screw theory and the Euler-Lagrange equations. First the used frames and assumptions will be defined. Secondly the kinematics of the robot will be discussed and finally also the full dynamic equations will be derived.

Frames In Figure 4-2 the different frames that will be used are shown. In Figure 4-2a the sphere frames Ψ_{k} and Ψ_{s} are shown. Here the origin of frame Ψ_{k} coincides with the center of mass of the ball and is only translated with respect to the inertial frame Ψ_0. The origin of frame Ψ_{s} also coincides with the center of mass of the ball and the frame is rigidly connected to the ball. This frame will also be referred to as the *sphere frame*.

In Figure 4-2 the two body frames Ψ_{r} and Ψ_{b} are shown. Here the origin of Ψ_{r} is coincides with the center of mass of the ball and is rigidly connected to the body, i.e. the frame has the same rotation as the body. The origin of Ψ_{b} coincides with the center of mass of the body and is rigidly connected to the body. This frame will also be referred to as the *body frame*.

4-1 Assumptions

In this section all assumptions are listed

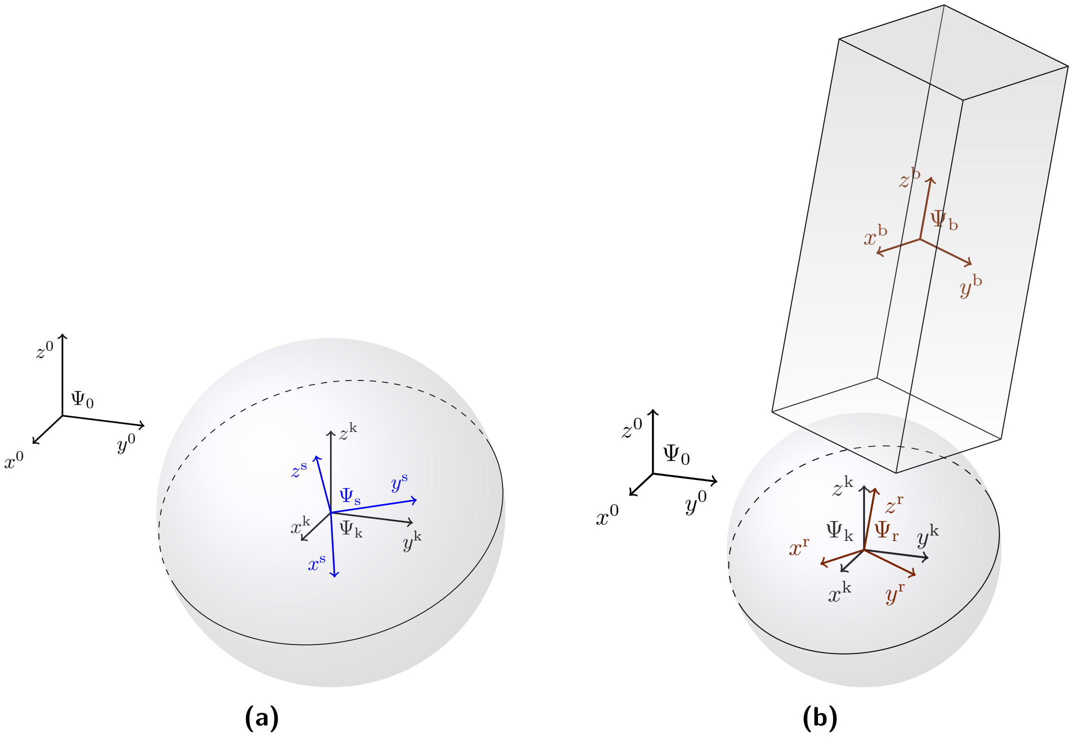


Figure 4-2: Frames of the ballbot. Left: ball frames, right: Body frames

- Zero slip between the sphere and the ground, i.e. the instantaneous velocity at the contact point is always zero.
- There is no slip between the actuation wheels and the ball.
- The sphere is homogeneous, i.e. in the the center of mass of the sphere $j_\mathrm{s} = j_{\mathrm{s},x} = j_{\mathrm{s},y} = j_{\mathrm{s},z}$
- The surface on which the ball rolls is flat and horizontal.
- The ball is rigid and therefore does not deform, i.e. the radius of the ball r_s is constant
- The body can be modelled as a pendulum that rotates in a spherical joint in the center of the ball.
- The distance between the center of mass of the ball and body is constant.

4-2 Kinematics

In this section the kinematic relations of the robot are derived. First of all the homogeneous matrices relating the different frames are defined. Subsequently configuration space and twists are defined and the geometric Jacobian will be constructed.

Homogeneous matrices The relation between the defined frames can be expressed using homogeneous matrices. The homogeneous matrices relating a certain frame to the inertial frame will later be used to find the twists between the inertial frame and a certain frame.

Frame Ψ_k is related to the inertial frame Ψ_0 by a pure translation p_k^0:

$$H_\mathrm{k}^0 = \begin{bmatrix} I & p_\mathrm{k}^0 \\ 0 & 1 \end{bmatrix} \tag{4-1}$$

Frame Ψ_s is related to frame Ψ_k through a rotation R_s^k and no translation.

$$H_\mathrm{s}^\mathrm{k} = \begin{bmatrix} R_\mathrm{s}^\mathrm{k} & 0 \\ 0 & 1 \end{bmatrix}$$

The homogeneous matrix relating the sphere frame Ψ_s and the inertial frame Ψ_0 can be found using the following matrix multiplication:

$$\begin{aligned} H_\mathrm{s}^0 &= H_\mathrm{k}^0 H_\mathrm{s}^\mathrm{k} \\ &= \begin{bmatrix} I & p_\mathrm{k}^0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_\mathrm{s}^\mathrm{k} & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_\mathrm{s}^\mathrm{k} & p_\mathrm{k}^0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

From this homogeneous matrix, it is obvious that $R_\mathrm{s}^0 = R_\mathrm{s}^\mathrm{k}$. Therefore, the homogeneous matrix relating the sphere frame to the inertial frame is given by:

$$H_\mathrm{s}^0 = \begin{bmatrix} R_\mathrm{s}^0 & p_\mathrm{k}^0 \\ 0 & 1 \end{bmatrix} \tag{4-2}$$

Similar to the sphere frame Ψ_s, the homogeneous matrix for frame Ψ_r is given by

$$H_\mathrm{r}^0 = \begin{bmatrix} R_\mathrm{r}^0 & p_\mathrm{k}^0 \\ 0 & 1 \end{bmatrix} \tag{4-3}$$

with $R_\mathrm{r}^\mathrm{k} = R_\mathrm{r}^0$.

Finally the homogeneous matrix between the body frame Ψ_b and the inertial frame is again found using the following matrix multiplication:

$$\begin{aligned} H_\mathrm{b}^0 &= H_\mathrm{r}^0 H_\mathrm{b}^\mathrm{r} \\ &= \begin{bmatrix} R_\mathrm{r}^0 & p_\mathrm{k}^0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & p_\mathrm{b}^\mathrm{r} \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_\mathrm{r}^0 & R_\mathrm{r}^0 p_\mathrm{b}^\mathrm{r} + p_\mathrm{k}^0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Again, it is obvious that $R_\mathrm{b}^0 = R_\mathrm{r}^0$. Therefore the homogeneous matrix between the body frame and the inertial frame is given by:

$$H_\mathrm{b}^0 = \begin{bmatrix} R_\mathrm{b}^0 & R_\mathrm{b}^0 p_\mathrm{b}^\mathrm{r} + p_\mathrm{k}^0 \\ 0 & 1 \end{bmatrix} \tag{4-4}$$

Summarising, the following useful homogeneous matrices are defined:

$$\boxed{H_\mathrm{k}^0 = \begin{bmatrix} I & p_\mathrm{k}^0 \\ 0 & 1 \end{bmatrix} \qquad H_\mathrm{s}^0 = \begin{bmatrix} R_\mathrm{s}^0 & p_\mathrm{k}^0 \\ 0 & 1 \end{bmatrix} \qquad H_\mathrm{b}^0 = \begin{bmatrix} R_\mathrm{b}^0 & R_\mathrm{b}^0 p_\mathrm{b}^\mathrm{r} + p_\mathrm{k}^0 \\ 0 & 1 \end{bmatrix}}$$

Furthermore, the following properties hold:

$$R_\mathrm{s}^\mathrm{k} = R_\mathrm{s}^0, \qquad R_\mathrm{r}^\mathrm{k} = R_\mathrm{b}^\mathrm{k} = R_\mathrm{b}^0, \qquad p_\mathrm{r}^\mathrm{b} = \text{constant}$$

Configuration space The *workspace* of the robot is given by the homogeneous matrices of the two bodies: $H_\mathrm{s}^0(R_\mathrm{s}^0, p_\mathrm{s}^0)$ and $H_\mathrm{b}^0(R_\mathrm{b}^0, p_\mathrm{b}^0)$. In other words, the workspace of the system is given by $SE(3) \times SE(3)$. However, as can be seen from the expressions of these homogeneous matrices in equations (4-1) to (4-4), the homogeneous matrices can all be expressed as a function of the rotation matrices R_s^0 and R_b^0 and the translation p_k^0, which form the *configuration space* $SO(3) \times SO(3) \times \mathbb{R}^3$. Using the flattened form of the rotation matrices, the flattened configuration space is given by:

$$\bar{q} = \begin{bmatrix} \bar{R}_\mathrm{s}^0 \\ \bar{R}_\mathrm{b}^0 \\ p_\mathrm{k}^0 \end{bmatrix} \tag{4-5}$$

Configuration twist The configuration twists corresponding to the configuration space are the angular velocities $\omega_\mathrm{s}^{0,0}$ and $\omega_\mathrm{b}^{0,0}$ and linear velocity $v_\mathrm{k}^{0,0}$:

$$\dot{q} = \begin{bmatrix} \omega_\mathrm{s}^{0,0} \\ \omega_\mathrm{b}^{0,0} \\ v_\mathrm{k}^{0,0} \end{bmatrix} \tag{4-6}$$

Recall from the preliminaries in Section 3-1 the relation between the tilde form of the twists $\tilde{T}_\mathrm{i}^{0,0}$ and the homogeneous matrices:

$$\tilde{T}_\mathrm{i}^{0,0} = \begin{bmatrix} \tilde{\omega}_\mathrm{i}^{0,0} & v_\mathrm{i}^{0,0} \\ 0 & 0 \end{bmatrix} = \dot{H}_\mathrm{i}^0 H_0^\mathrm{i}$$

where i denotes a certain frame Ψ_i. Using this definition a relation between the configuration space and the configuration twists is found.

For frame Ψ_k:

$$\begin{aligned} \tilde{T}_\mathrm{k}^{0,0} = \dot{H}_\mathrm{k}^0 H_0^\mathrm{k} &= \begin{bmatrix} 0 & \dot{p}_\mathrm{k}^0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I & -p_\mathrm{k}^0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \dot{p}_\mathrm{k}^0 \\ 0 & 0 \end{bmatrix} \end{aligned}$$

Therefore, the following relations hold:

$$\boxed{\tilde{\omega}_\mathrm{k}^{0,0} = 0, \qquad \omega_\mathrm{k}^{0,0} = 0, \qquad v_\mathrm{k}^{0,0} = \dot{p}_\mathrm{k}^0 \qquad (4\text{-}7)}$$

For frame Ψ_s:

$$\begin{aligned}\tilde{T}_\mathrm{s}^{0,0} = \dot{H}_\mathrm{s}^0 H_0^\mathrm{s} &= \begin{bmatrix}\dot{R}_\mathrm{s}^0 & \dot{p}_\mathrm{k}^0\\ 0 & 0\end{bmatrix}\begin{bmatrix}R_0^\mathrm{s} & -R_0^\mathrm{s}p_\mathrm{k}^0\\ 0 & 1\end{bmatrix}\\ &= \begin{bmatrix}\dot{R}_\mathrm{s}^0 R_0^\mathrm{s} & -\dot{R}_\mathrm{s}^0 R_0^\mathrm{s} p_\mathrm{k}^0 + \dot{p}_\mathrm{k}^0\\ 0 & 0\end{bmatrix}\end{aligned}$$

Using the anti-commutative property of the cross product: $-\tilde{\omega}_\mathrm{s}^{0,0}p_\mathrm{k}^0 = \tilde{p}_\mathrm{k}^0\omega_\mathrm{s}^{0,0}$, the following relations hold:

$$\boxed{\tilde{\omega}_\mathrm{s}^{0,0} = \dot{R}_\mathrm{s}^0 R_0^\mathrm{s}, \qquad v_\mathrm{s}^{0,0} = \tilde{p}_\mathrm{k}^0\omega_\mathrm{s}^{0,0} + v_\mathrm{k}^{0,0}} \tag{4-8}$$

Finally, for frame Ψ_b:

$$\begin{aligned}\tilde{T}_\mathrm{b}^{0,0} = \dot{H}_\mathrm{b}^0 H_0^\mathrm{b} &= \begin{bmatrix}\dot{R}_\mathrm{b}^0 & \dot{R}_\mathrm{b}^0 p_\mathrm{b}^\mathrm{r} + \dot{p}_\mathrm{k}^0\\ 0 & 0\end{bmatrix}\begin{bmatrix}R_0^\mathrm{b} & -R_0^\mathrm{b}\left(R_\mathrm{b}^0 p_\mathrm{b}^\mathrm{r} + p_\mathrm{k}^0\right)\\ 0 & 1\end{bmatrix}\\ &= \begin{bmatrix}\dot{R}_\mathrm{b}^0 R_0^\mathrm{b} & -\dot{R}_\mathrm{b}^0 R_0^\mathrm{b}\left(R_\mathrm{b}^0 p_\mathrm{b}^\mathrm{r} + p_\mathrm{k}^0\right) + \dot{R}_\mathrm{b}^0 p_\mathrm{b}^\mathrm{r} + \dot{p}_\mathrm{k}^0\\ 0 & 0\end{bmatrix}\end{aligned}$$

Using $\dot{R}_\mathrm{b}^0 = \tilde{\omega}_\mathrm{b}^{0,0}R_\mathrm{b}^0$, the following simplification can be made:

$$\begin{aligned}\tilde{T}_\mathrm{b}^{0,0} &= \begin{bmatrix}\dot{R}_\mathrm{b}^0 R_0^\mathrm{b} & \left(-\tilde{\omega}_\mathrm{b}^{0,0}R_\mathrm{b}^0 p_\mathrm{b}^\mathrm{r} + \tilde{\omega}_\mathrm{b}^{0,0}R_\mathrm{b}^0 p_\mathrm{b}^\mathrm{r}\right) - \tilde{\omega}_\mathrm{b}^{0,0}p_\mathrm{k}^0 + \dot{p}_\mathrm{k}^0\\ 0 & 0\end{bmatrix}\\ &= \begin{bmatrix}\dot{R}_\mathrm{b}^0 R_0^\mathrm{b} & -\tilde{\omega}_\mathrm{b}^{0,0}p_\mathrm{k}^0 + \dot{p}_\mathrm{k}^0\\ 0 & 0\end{bmatrix}\end{aligned}$$

Therefore, the following relations hold:

$$\boxed{\tilde{\omega}_\mathrm{b}^{0,0} = \dot{R}_\mathrm{b}^0 R_0^\mathrm{b}, \qquad v_\mathrm{b}^{0,0} = \tilde{p}_\mathrm{k}^0\omega_\mathrm{b}^{0,0} + v_\mathrm{k}^{0,0}} \tag{4-9}$$

Using these results, the configuration twist are related to the configuration space by:

$$\tilde{\omega}_\mathrm{s}^{0,0} = \dot{R}_\mathrm{s}^0 R_0^\mathrm{s} \tag{4-10}$$

$$\tilde{\omega}_\mathrm{b}^{0,0} = \dot{R}_\mathrm{b}^0 R_0^\mathrm{b} \tag{4-11}$$

$$v_\mathrm{k}^{0,0} = \dot{p}_\mathrm{k}^0 \tag{4-12}$$

However, it is also desired to find the relation between the derivative of the flattened configuration space and the normal form of the angular velocity. To find this relation the previously found relation between the angular velocity expressed body frame and the flattened rotation matrix is used: $\dot{\bar{R}}_\mathrm{i}^0 = \Phi(R_\mathrm{i}^0)\omega_\mathrm{i}^{\mathrm{i},0}$. This angular velocity can be translated to the inertial frame by $\omega_\mathrm{i}^{\mathrm{i},0} = R_0^\mathrm{i}\ \omega_\mathrm{i}^{0,0}$. It is now possible to define a map $\Theta(R_\mathrm{s}^0, R_\mathrm{b}^0)$ that relates the configuration twists and the derivatives of the flattened configuration space:

$$\underbrace{\begin{bmatrix}\dot{\bar{R}}_\mathrm{s}^0\\ \dot{\bar{R}}_\mathrm{b}^0\\ \dot{p}_\mathrm{k}^0\end{bmatrix}}_{\dot{\bar{q}}} = \underbrace{\begin{bmatrix}\Phi(R_\mathrm{s}^0)(R_\mathrm{s}^0)^T & 0 & 0\\ 0 & \Phi(R_\mathrm{b}^0)(R_\mathrm{b}^0)^T & 0\\ 0 & 0 & I\end{bmatrix}}_{\Theta(R_\mathrm{s}^0, R_\mathrm{b}^0)}\underbrace{\begin{bmatrix}\omega_\mathrm{s}^{0,0}\\ \omega_\mathrm{b}^{0,0}\\ v_\mathrm{k}^{0,0}\end{bmatrix}}_{\dot{q}} \tag{4-13}$$

Geometric Jacobian Using the relations between the body twists and the configuration velocities given by (4-8) and (4-9), the geometric Jacobian can be constructed.

$$\begin{bmatrix} T_{\mathrm{s}}^{0,0} \\ T_{\mathrm{b}}^{0,0} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ \tilde{p}_{\mathrm{k}}^{0} & 0 & I \\ 0 & I & 0 \\ 0 & \tilde{p}_{\mathrm{k}}^{0} & I \end{bmatrix} \begin{bmatrix} \omega_{\mathrm{s}}^{0,0} \\ \omega_{\mathrm{b}}^{0,0} \\ v_{\mathrm{k}}^{0,0} \end{bmatrix} \tag{4-14}$$

Thus the geometric Jacobian is given by:

$$J = \begin{bmatrix} I & 0 & 0 \\ \tilde{p}_{\mathrm{k}}^{0} & 0 & I \\ 0 & I & 0 \\ 0 & \tilde{p}_{\mathrm{k}}^{0} & I \end{bmatrix} \tag{4-15}$$

4-3 Dynamics

In this section the dynamics of the robot are derived using the Euler-Lagrange equations. First the kinetic and potential energy will be derived, after which the equations of motion can be computed. This section will be concluded with the inclusion of the rolling constraint in the equations of motion.

Inertia tensor Before it is possible to compute the kinetic energy, the inertia tensor needs to be defined. The frames Ψ_{s} and Ψ_{b} are rigidly connected to in the center of mass of the ball and body of the robot respectively, and therefore correspond to the principal inertia frames of the rigid bodies, i.e. the inertia tensor takes the form

$$\mathcal{I}^{\mathrm{i,i}} = \begin{bmatrix} J_{\mathrm{i}} & 0 \\ 0 & m_{\mathrm{i}} I \end{bmatrix} \tag{4-16}$$

with

$$J_{\mathrm{i}} = \begin{bmatrix} j_{\mathrm{i},x} & 0 & 0 \\ 0 & j_{\mathrm{i},y} & 0 \\ 0 & 0 & j_{\mathrm{i},z} \end{bmatrix} \tag{4-17}$$

where $j_{\mathrm{i,j}}$ denotes the moment of inertia of i around axis j and m_{i} the mass of i. The inertia tensors of the sphere frame and body frame are given by:

$$\mathcal{I}^{\mathrm{s,s}} = \begin{bmatrix} J_{\mathrm{s}} & 0 \\ 0 & m_{\mathrm{s}} I \end{bmatrix} \qquad \mathcal{I}^{\mathrm{b,b}} = \begin{bmatrix} J_{\mathrm{b}} & 0 \\ 0 & m_{\mathrm{b}} I \end{bmatrix}$$

For a homogeneous ball, it holds that $j_{\mathrm{s}} = j_{\mathrm{s},x} = j_{\mathrm{s},y} = j_{\mathrm{s},z}$, $J_{\mathrm{s}} = j_{\mathrm{s}} I$.

The defined inertia tensors are defined in the corresponding body frames. However, the dynamics will be expressed in the inertial frame. Therefore it is required that the inertia tensors are transformed to the inertia frame as well. The transformation of a inertia tensor expressed in body frame Ψ_{i} to the inertia frame is given by the following relation:

$$\mathcal{I}^{0,\mathrm{i}} = \left(Ad_{H_0^{\mathrm{i}}} \right)^T \mathcal{I}^{\mathrm{i,i}} Ad_{H_0^{\mathrm{i}}} \tag{4-18}$$

with

$$Ad_{H_0^{\mathrm{i}}} = \begin{bmatrix} \left(R_{\mathrm{i}}^{0}\right)^{T} & 0 \\ -\left(R_{\mathrm{i}}^{0}\right)^{T} \tilde{p}_{\mathrm{i}}^{0} & \left(R_{\mathrm{i}}^{0}\right)^{T} \end{bmatrix} \tag{4-19}$$

Kinetic energy and inertia matrix The kinetic energy of the total system is given by

$$T^{*} = \frac{1}{2} \begin{bmatrix} T_{\mathrm{s}}^{0,0} & T_{\mathrm{b}}^{0,0} \end{bmatrix} \begin{bmatrix} \mathcal{I}^{0,\mathrm{s}} & 0 \\ 0 & \mathcal{I}^{0,\mathrm{b}} \end{bmatrix} \begin{bmatrix} T_{\mathrm{s}}^{0,0} \\ T_{\mathrm{b}}^{0,0} \end{bmatrix} \tag{4-20}$$

Using equation (4-14) the kinetic energy can be expressed in the configuration velocities:

$$T^{*} = \frac{1}{2} \dot{q}^{T} \underbrace{J^{T} \begin{bmatrix} \mathcal{I}^{0,\mathrm{s}} & 0 \\ 0 & \mathcal{I}^{0,\mathrm{b}} \end{bmatrix} J}_{M(q)} \dot{q} \tag{4-21}$$

Here $M(q)$ denotes the inertia matrix. The resulting kinetic energy is given by:

$$T^{*}(q) = \frac{1}{2} \dot{q}^{T} M(q) \dot{q} \tag{4-22}$$

Potential energy Now the potential energy of the robot will be derived.

The center of mass of the sphere is at constant height. Therefore the potential energy of the sphere remains constant and will have no influence on the dynamics. Furthermore, if the inertia frame is chosen level with frame Ψ_{k}, the potential energy is zero and will remain zero. Therefore only the potential energy of the body of the robot is considered.

The body frame Ψ_{b} is fixed to the center of mass, therefore the point of the center of mass expressed in frame Ψ_{b} is given by $P^{\mathrm{b}} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^{T}$. This point can easily be written in the inertia frame by using the homogeneous matrix H_{b}^{0}. The potential energy can then be expressed as

$$V_{\mathrm{b}} = m_{\mathrm{b}}\, g \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} H_{\mathrm{b}}^{0} P^{\mathrm{b}} \tag{4-23}$$

Given the homogeneous matrix expressed in the configuration space, the potential energy is given by

$$V_{\mathrm{b}} = m_{\mathrm{b}}\, g \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{\mathrm{b}}^{0} & R_{\mathrm{b}}^{0} p_{\mathrm{b}}^{\mathrm{r}} + p_{\mathrm{k}}^{0} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

This results in a final expression for the potential energy in the configuration space:

$$V_{\mathrm{b}}(q) = m_{\mathrm{b}}\, g \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \left(R_{\mathrm{b}}^{0} p_{\mathrm{b}}^{\mathrm{r}} + p_{\mathrm{k}}^{0} \right) \tag{4-24}$$

Euler-Lagrange equation Given the kinetic and potential energy, it is possible to compute the Lagrangian

$$\mathcal{L} = T^*(\dot{q}, q) - V_\mathrm{b}(q)$$

and compute the equations of motion using the Euler-Lagrange equations

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = F_\mathrm{ext}^T \tag{4-25}$$

Given

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial q} &= \frac{\partial \bar{q}}{\partial q}\frac{\partial \mathcal{L}}{\partial \bar{q}} \\ &= \Theta^T(R_\mathrm{s}^0, R_\mathrm{b}^0)\frac{\partial \mathcal{L}}{\partial \bar{q}}\end{aligned} \tag{4-26}$$

the Euler-Lagrange equations can be easily computed and can be rewritten in the form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = F_\mathrm{ext}^T \tag{4-27}$$

Where $M(q)$ denotes the inertia matrix, $C(q,\dot{q})$ the Coriolis matrix, $G(q)$ the gravity matrix and F_ext denotes all external forces.

Rolling constraint Up to so far the dynamics of the robot with no implicit interaction with the floor are derived and given in (4-27). However, in reality the sphere is in contact with the floor and subjected to pure rolling, i.e. rolling without slip. The following part will describe how to include the rolling constraint into the equations of motion.

The pure rolling constraint entails that the velocity at the contact point is equal to zero. Let us consider an additional frame Ψ_c at the contact point between the sphere and the floor. This frame is related to frame Ψ_k by $p_\mathrm{c}^\mathrm{k} = r = \begin{bmatrix}0 & 0 & -r_s\end{bmatrix}^T$, where r_s denotes the radius of the ball:

$$H_\mathrm{k}^\mathrm{c} = \begin{bmatrix} I & -r \\ 0 & 1\end{bmatrix} \tag{4-28}$$

Given frame Ψ_c at the contact point, it is known that the linear velocity of the sphere relative to the inertia frame as seen from frame Ψ_c should be equal to zero, i.e. $v_\mathrm{s}^{\mathrm{c},0} = 0$. Relating this velocity to the twists of frame Ψ_k results in:

$$v_\mathrm{s}^{\mathrm{c},0} = -\tilde{r}\omega_\mathrm{s}^{\mathrm{k},0} + v_\mathrm{s}^{\mathrm{k},0} = 0 \tag{4-29}$$

The next step is to express $\omega_\mathrm{s}^{\mathrm{k},0}$ and $v_\mathrm{s}^{\mathrm{k},0}$ in the configuration velocities. The angular velocity $\omega_\mathrm{s}^{\mathrm{k},0}$ can be expressed as:

$$\omega_\mathrm{s}^{\mathrm{k},0} = R_0^\mathrm{k}\omega_\mathrm{s}^{0,0} = \omega_\mathrm{s}^{0,0} \tag{4-30}$$

The linear velocity $v_\mathrm{s}^{\mathrm{k},0}$ is slightly more involved. First recall the relation for the linear velocity $v_\mathrm{s}^{0,0}$ from equation (4-8):

$$v_\mathrm{s}^{0,0} = \tilde{p}_\mathrm{k}^0\omega_\mathrm{s}^{0,0} + v_\mathrm{k}^{0,0}$$

Rewriting $v_{\mathrm{s}}^{\mathrm{k},0}$ to the inertia frame and substituting $v_{\mathrm{s}}^{0,0}$:

$$\begin{aligned} v_{\mathrm{s}}^{\mathrm{k},0} &= \tilde{p}_0^{\mathrm{k}} \omega_{\mathrm{s}}^{0,0} + v_{\mathrm{s}}^{0,0} \\ &= \tilde{p}_0^{\mathrm{k}} \omega_{\mathrm{s}}^{0,0} + \tilde{p}_{\mathrm{k}}^{0} \omega_{\mathrm{s}}^{0,0} + v_{\mathrm{k}}^{0,0} \end{aligned}$$

Thus

$$v_{\mathrm{s}}^{\mathrm{k},0} = v_{\mathrm{k}}^{0,0} \tag{4-31}$$

Now the rolling constraint expressed as a function of the configuration velocities is given by:

$$-\tilde{r}\omega_{\mathrm{s}}^{0,0} + v_{\mathrm{k}}^{0,0} = 0 \tag{4-32}$$

And written as Pfaffian constraint:

$$\begin{aligned} \begin{bmatrix} -\tilde{r} & 0 & I \end{bmatrix} \begin{bmatrix} \omega_{\mathrm{s}}^{0,0} \\ \omega_{\mathrm{b}}^{0,0} \\ v_{\mathrm{k}}^{0,0} \end{bmatrix} &= 0 \\ A\dot{q} &= 0 \end{aligned} \tag{4-33}$$

Reduced equations of motion The Pfaffian constraint given in equation (4-33) can be included through the means of Lagrangian multipliers, which impose a virtual force on the equations of motions:

$$\begin{cases} \dfrac{d}{dt}\left(\dfrac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \dfrac{\partial \mathcal{L}}{\partial q} = A^T\lambda + F_{\mathrm{ext}}^T \\ A\dot{q} = 0 \end{cases} \tag{4-34}$$

The Lagrangian multipliers can be explicitly solved. However, for the relative larger system matrices this might be computational intensive. Instead the null space of the constraint matrix is used to eliminate the Lagrangian multipliers from the equations of motion. The vector $\dot{q}$ should lie in this null space.

Consider a set of reduced generalised velocities $\dot{q}_r$, with

$$\dot{q} = S\dot{q}_r \tag{4-35}$$

By pre-multiplying with S^T, the following reduced of motion are obtained:

$$\begin{aligned} S^T M(q)\ddot{q} + S^T C(q,\dot{q})\dot{q} + S^T G(q) &= S^T F_{\mathrm{ext}}^T \\ S^T M(q)S\ddot{q}_r + \left(S^T M\dot{S} + S^T C(q,\dot{q})S\right)\dot{q}_r + S^T G(q) &= S^T F_{\mathrm{ext}}^T \\ \bar{M}\ddot{q}_r + \bar{C}\dot{q}_r + \bar{G} &= S^T F_{\mathrm{ext}}^T \end{aligned}$$

with

$$\begin{aligned} \bar{M} &= S^T M(q) S \\ \bar{C} &= S^T M\dot{S} + S^T C(q,\dot{q})S \\ \bar{G} &= S^T G(q) \end{aligned}$$

Now the following reduced configuration space of the dynamics is proposed:

$$\dot{q}_r = \begin{bmatrix} \omega_\mathrm{s}^{0,0} \\ \omega_\mathrm{b}^{0,0} \end{bmatrix} \tag{4-36}$$

with

$$\dot{q} = \underbrace{\begin{bmatrix} I & 0 \\ 0 & I \\ \tilde{r} & 0 \end{bmatrix}}_{S} \dot{q}_r \tag{4-37}$$

Using this transformation, the *configuration space of the dynamics* is reduced from $SO(3) \times SO(3) \times \mathbb{R}^3$ to $SO(3) \times SO(3)$. However, the configuration space of the system is still $SO(3) \times SO(3) \times \mathbb{R}^2$, as it is desired to track the position of the robot. It can be verified that this matrix S lies indeed in the null space of the constraint matrix A. Furthermore, since S is a constant, the matrix $\dot{S} = 0$.

The final equations of motion are given by

$$\begin{cases} \ddot{q}_r & = \bar{M}^{-1}\left(S^T F_\mathrm{ext}^T - \bar{C}\dot{q}_r - \bar{G}\right) \\ \dot{\bar{q}} & = \Theta(R_\mathrm{s}^0, R_\mathrm{b}^0)\; S\; \dot{q}_r \end{cases} \tag{4-38}$$

here the dynamics $\ddot{q}_r$ are given by two angular accelerations and the kinematics $\dot{\bar{q}}$ by the derivatives of two rotation matrices and one linear translation.

4-4 External forces

In the previous section the equations of motion are derived, which were subjected to external forces. In this section these external forces are defined.

4-4-1 Actuation

The robot is actuated by a specific number wheels/rollers actuated by motors, which apply a wrench to both the ball and body. To find the relationship between the applied wrench $W^{0,\mathrm{i}}$ and the external forces F_ext, the supplied power is used:

$$P = F_\mathrm{ext}\dot{q} = \begin{bmatrix} W^{0,\mathrm{s}} & W^{0,\mathrm{b}} \end{bmatrix} \begin{bmatrix} T_\mathrm{s}^{0,0} \\ T_\mathrm{b}^{0,0} \end{bmatrix} \tag{4-39}$$

Now recall the following relation

$$\begin{bmatrix} T_\mathrm{s}^{0,0} \\ T_\mathrm{b}^{0,0} \end{bmatrix} = J\dot{q} \tag{4-40}$$

Therefore

$$F_\mathrm{ext}\dot{q} = \begin{bmatrix} W^{0,\mathrm{s}} & W^{0,\mathrm{b}} \end{bmatrix} J\dot{q}$$

Now it is possible to express the external forces as a function of the applied wrenches:

$$F_{\text{ext}}^T = J^T \begin{bmatrix} (W^{0,\text{s}})^T \\ (W^{0,\text{b}})^T \end{bmatrix} \tag{4-41}$$

In the remainder of this section the two wrenches applied to both sphere and body are derived.

Actuation of the sphere First consider the actuation of the sphere. Two sets of vectors are specified. The first set are the direction unit vectors of the actuators $\hat{u}_j^{\text{r}}$, which give the direction in which the applied force works. The second set are the position unit vectors $\hat{r}_j^{\text{r}}$, pointing from the center of the sphere to the contact points with the wheels. Here j denotes a specific actuator and the superscript the frame in which the vector is expressed. Note that it is chosen to express the vectors in frame Ψ_{r}, which is fixed in the center of the sphere and rigidly connected to the body. As this frame is fixed with respect to the body, these sets of vectors are constant.

For wheel j, the magnitude of the force applied at the contact point f_j and the magnitude of the input torque τ_j are related by

$$f_j = \frac{\tau_j}{r_w} \tag{4-42}$$

Here r_w denotes the radius of the wheel. The direction of this force on the ball expressed in frame Ψ_{r} is given by the unit vector $\hat{u}_j^{\text{r}}$, thus the force applied to the sphere is given by

$$\begin{aligned} \left(f_{\text{j}}^{\text{r,s}}\right)^T &= f_j \hat{u}_j^{\text{r}} \\ &= \frac{1}{r_w} \tau_j \hat{u}_j^{\text{r}} \end{aligned}$$

The torque applied to the ball is then

$$\begin{aligned} \left(\tau_{\text{j}}^{\text{r,s}}\right)^T &= r_s \hat{r}_j^{\text{r}} \times f^{\text{r,s}} \\ &= \frac{r_s}{r_w} \tau_j \left(\hat{r}_j^{\text{r}} \times \hat{u}_j^{\text{r}}\right) \end{aligned}$$

Now, using the tilde form to express the cross product and given $W^{\text{r,s}} = \begin{bmatrix} \tau^{\text{r,s}} & f^{\text{r,s}} \end{bmatrix}$, the wrench on the sphere expressed in frame Ψ_{r} as a function of the input torques for n actuators is given by:

$$(W^{\text{r,s}})^T = \begin{bmatrix} \frac{r_s}{r_w} I & 0 \\ 0 & \frac{1}{r_w} I \end{bmatrix} \begin{bmatrix} \tilde{\hat{r}}_1^{\text{r}} \hat{u}_1^{\text{r}} & \tilde{\hat{r}}_2^{\text{r}} \hat{u}_2^{\text{r}} & \dots & \tilde{\hat{r}}_n^{\text{r}} \hat{u}_n^{\text{r}} \\ \hat{u}_1^{\text{r}} & \hat{u}_2^{\text{r}} & \dots & \hat{u}_n^{\text{r}} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{bmatrix} \tag{4-43}$$

In equation (4-43) the wrench is expressed in frame Ψ_r, but the equations of motion of the sphere are expressed in the inertia frame Ψ_0. However, these can easily be translated into frame Ψ_k with:

$$\left(W^{0,\mathrm{s}}\right)^T = \left(Ad_{H_0^\mathrm{r}}\right)^T (W^{\mathrm{r,s}})^T \tag{4-44}$$

This transposed adjoint matrix is given by:

$$\left(Ad_{H_0^\mathrm{r}}\right)^T = \begin{bmatrix} R_\mathrm{r}^0 & -R_\mathrm{r}^0 \tilde{p}_0^\mathrm{r} \\ 0 & R_\mathrm{r}^0 \end{bmatrix}$$

It will be convenient to rewrite this adjoint matrix as a function of the configuration space. The translation $\tilde{p}_0^\mathrm{r}$ can be rewritten as

$$p_0^\mathrm{r} = -R_0^\mathrm{r} p_\mathrm{k}^0$$

Using the identity $(Rp)^\sim = R\tilde{p}R^T$ and $\tilde{a}^T = -\tilde{a}$, it is obtained that $-R_\mathrm{r}^0 \tilde{p}_0^\mathrm{r} = \tilde{p}_\mathrm{k}^0 R_\mathrm{r}^0$. Finally, using $R_\mathrm{r}^0 = R_\mathrm{b}^0$, the following adjoint matrix is obtained:

$$\left(Ad_{H_0^\mathrm{r}}\right)^T = \begin{bmatrix} R_\mathrm{b}^0 & \tilde{p}_\mathrm{k}^0 R_\mathrm{b}^0 \\ 0 & R_\mathrm{b}^0 \end{bmatrix}$$

Therefore, the wrench applied to the sphere in inertia frame Ψ_0 is given by:

$$\left(W^{0,\mathrm{s}}\right)^T = \begin{bmatrix} R_\mathrm{b}^0 & \tilde{p}_\mathrm{k}^0 R_\mathrm{b}^0 \\ 0 & R_\mathrm{b}^0 \end{bmatrix} \begin{bmatrix} \frac{r_s}{r_w} I & 0 \\ 0 & \frac{1}{r_w} I \end{bmatrix} \begin{bmatrix} \tilde{\hat{r}}_1^\mathrm{r} \hat{u}_1^\mathrm{r} & \tilde{\hat{r}}_2^\mathrm{r} \hat{u}_2^\mathrm{r} & \dots & \tilde{\hat{r}}_n^\mathrm{r} \hat{u}_n^\mathrm{r} \\ \hat{u}_1^\mathrm{r} & \hat{u}_2^\mathrm{r} & \dots & \hat{u}_n^\mathrm{r} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{bmatrix} \tag{4-45}$$

Three omni-wheel configuration The ballbot of Alten Mechatronics is actuated by three motors with omni-wheels. A schematic image of the position of the three omni-wheel configuration is shown in Figure 4-3. In the xy plane the angle between the actuators is $\frac{2}{3}\pi$ radians (120 degrees).

For this configuration, the actuator direction unit vectors and position unit vectors are given by:

$$\tilde{\hat{r}}_1^\mathrm{r} = \begin{bmatrix} \sin(\alpha) \\ 0 \\ \cos(\alpha) \end{bmatrix} \qquad \tilde{\hat{r}}_2^\mathrm{r} = \begin{bmatrix} -\frac{1}{2}\sin(\alpha) \\ \frac{1}{2}\sqrt{3}\sin\alpha \\ \cos(\alpha) \end{bmatrix} \qquad \tilde{\hat{r}}_3^\mathrm{r} = \begin{bmatrix} -\frac{1}{2}\sin(\alpha) \\ -\frac{1}{2}\sqrt{3}\sin\alpha \\ \cos(\alpha) \end{bmatrix}$$

$$\hat{u}_1^\mathrm{r} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \hat{u}_2^\mathrm{r} = \begin{bmatrix} -\frac{1}{2}\sqrt{3} \\ -\frac{1}{2} \\ 0 \end{bmatrix} \qquad \hat{u}_3^\mathrm{r} = \begin{bmatrix} \frac{1}{2}\sqrt{3} \\ -\frac{1}{2} \\ 0 \end{bmatrix}$$

Using these vectors and having $\alpha = 45^\circ = \frac{\pi}{4}$ rad, the applied wrench to the sphere in

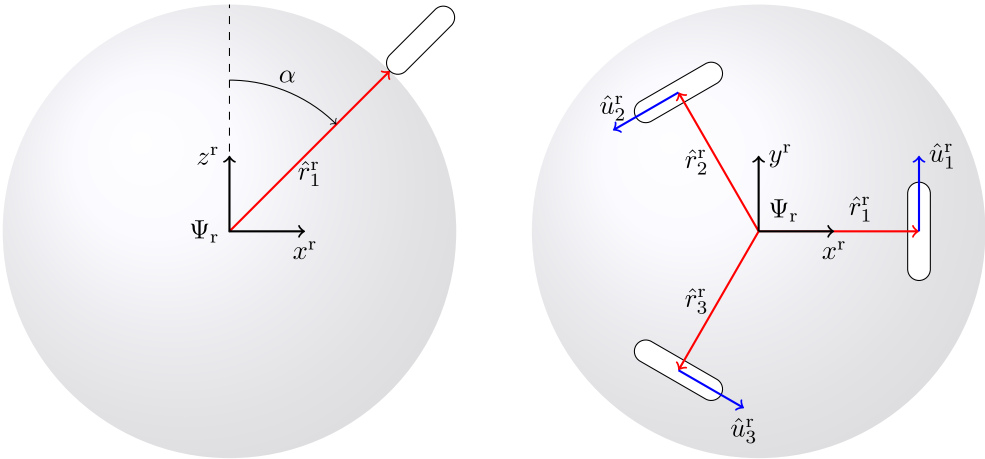


Figure 4-3: 3 omni-wheels configuration in frame Ψ_{r}. Left: front view in frame Ψ_{r}. Right: top view in frame Ψ_{r}

frame Ψ_{k} is given by:

$$\left(W^{0,\mathrm{s}}\right)^T = \begin{bmatrix} R_{\mathrm{b}}^0 & \tilde{p}_{\mathrm{k}}^0 R_{\mathrm{b}}^0 \\ 0 & R_{\mathrm{b}}^0 \end{bmatrix} \begin{bmatrix} \dfrac{r_s}{r_w} I & 0 \\ 0 & \dfrac{1}{r_w} I \end{bmatrix} \begin{bmatrix} -\frac{1}{2}\sqrt{2} & \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{2} \\ 0 & -\frac{1}{4}\sqrt{6} & \frac{1}{4}\sqrt{6} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \\ 0 & -\frac{1}{2}\sqrt{3} & \frac{1}{2}\sqrt{3} \\ 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{4-46}$$

Actuation of the body The wheels and motors are considered as part of the body. As stated by the Newton's third law of motion, when one body exerts a force on a second body, the second body exerts a force with equal magnitude and opposite direction on the first body. This entails that the the wrench applied to the body (in frame Ψ_{r}) is opposite to the previously derived wrench acting on the sphere:

$$W^{\mathrm{r,s}} = -W^{\mathrm{r,r}}$$

Since frame Ψ_{r} is rigidly connected to the body frame Ψ_{b}, the following holds

$$\begin{aligned} W^{\mathrm{r,b}} &= -W^{\mathrm{r,s}} \\ W^{0,\mathrm{b}} &= -W^{0,\mathrm{s}} \end{aligned} \tag{4-47}$$

Determining total external forces It is now possible to express the external forces as a function of the input torques of the motors, combining equations (4-41) with (4-46) and (4-47):

$$F_{\text{ext}}^T = J^T \begin{bmatrix} (W^{\text{s},0})^T \\ (W^{\text{b},0})^T \end{bmatrix}$$

$$= \begin{bmatrix} I & -\tilde{p}_{\text{k}}^0 & 0 & 0 \\ 0 & 0 & I & -\tilde{p}_{\text{k}}^0 \\ 0 & I & 0 & I \end{bmatrix} \begin{bmatrix} I_{6\times 6} \\ -I_{6\times 6} \end{bmatrix} \begin{bmatrix} R_{\text{b}}^0 & \tilde{p}_{\text{k}}^0 R_{\text{b}}^0 \\ 0 & R_{\text{b}}^0 \end{bmatrix} \begin{bmatrix} \dfrac{r_s}{r_w} I & 0 \\ 0 & \dfrac{1}{r_w} I \end{bmatrix} \begin{bmatrix} \tilde{\hat{r}}_1^{\text{r}} \hat{u}_1^{\text{r}} & \tilde{\hat{r}}_2^{\text{r}} \hat{u}_2^{\text{r}} & \dots & \tilde{\hat{r}}_n^{\text{r}} \hat{u}_n^{\text{r}} \\ \hat{u}_1^{\text{r}} & \hat{u}_2^{\text{r}} & \dots & \hat{u}_n^{\text{r}} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{bmatrix}$$

$$= \begin{bmatrix} \frac{r_s}{r_w} R_{\text{b}}^0 \\ -\frac{r_s}{r_w} R_{\text{b}}^0 \\ 0 \end{bmatrix} \begin{bmatrix} \tilde{\hat{r}}_1^{\text{r}} \hat{u}_1^{\text{r}} & \tilde{\hat{r}}_2^{\text{r}} \hat{u}_2^{\text{r}} & \dots & \tilde{\hat{r}}_n^{\text{r}} \hat{u}_n^{\text{r}} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{bmatrix}$$

Therefore, with

$$S^T = \begin{bmatrix} I & 0 & (\tilde{r})^T \\ 0 & I & 0 \end{bmatrix}$$

the input on the system dynamics is given by:

$$S^T F_{\text{ext}}^T = S^T J^T \begin{bmatrix} (W^{\text{s},0})^T \\ (W^{\text{b},0})^T \end{bmatrix}$$

$$S^T F_{\text{ext}}^T = \frac{r_s}{r_w} \begin{bmatrix} I \\ -I \end{bmatrix} R_{\text{b}}^0 \begin{bmatrix} \tilde{\hat{r}}_1^{\text{r}} \hat{u}_1^{\text{r}} & \tilde{\hat{r}}_2^{\text{r}} \hat{u}_2^{\text{r}} & \dots & \tilde{\hat{r}}_n^{\text{r}} \hat{u}_n^{\text{r}} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_n \end{bmatrix} \tag{4-48}$$

For the 3 omni-wheels configuration, the applied input is given by

$$S^T F_{\text{ext}}^T = \frac{r_s}{r_w} \begin{bmatrix} I \\ -I \end{bmatrix} R_{\text{b}}^0 \begin{bmatrix} -\frac{1}{2}\sqrt{2} & \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{2} \\ 0 & -\frac{1}{4}\sqrt{6} & \frac{1}{4}\sqrt{6} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{4-49}$$

Let us conclude this section by rewriting the relation of (4-49) as

$$S^T F_{\text{ext}}^T = B\tau^T \tag{4-50}$$

where B given by

$$B = \frac{r_s}{r_w} \begin{bmatrix} I \\ -I \end{bmatrix} R_{\text{b}}^0 \begin{bmatrix} -\frac{1}{2}\sqrt{2} & \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{2} \\ 0 & -\frac{1}{4}\sqrt{6} & \frac{1}{4}\sqrt{6} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{bmatrix} \tag{4-51}$$

4-4-2 Friction

This section will be concluded with the modelling of the friction. The friction will be denoted by D:

$$\ddot{q}_r = \bar{M}^{-1} \left(-\bar{C}\dot{q}_r - \bar{G} - D + B\tau^T \right)$$

The friction will be modelled by a summation of Coulomb and viscous friction. Friction will be present at two locations in the robot: between the floor and the sphere and between the sphere and the body. Both types of friction are a function of the relative angular velocity between two objects.

The friction acting on the body of the robot is given by

$$D^{0,\mathrm{b}} = D_{c,1}\ \mathrm{sgn}\left(\omega_{\mathrm{b}}^{0,0} - \omega_{\mathrm{s}}^{0,0}\right) + D_{v,1}\ \left(\omega_{\mathrm{b}}^{0,0} - \omega_{\mathrm{s}}^{0,0}\right) \tag{4-52}$$

with $D_{c,1} \geq 0$ and $D_{v,1} \geq 0$ diagonal matrices.

The friction acting on the sphere is given by the sum of the friction between itself and the body and between the floor:

$$D^{0,\mathrm{s}} = D_{c,1}\ \mathrm{sgn}\left(\omega_{\mathrm{s}}^{0,0} - \omega_{\mathrm{b}}^{0,0}\right) + D_{v,1}\ \left(\omega_{\mathrm{s}}^{0,0} - \omega_{\mathrm{b}}^{0,0}\right) + D_{c,2}\ \mathrm{sgn}\left(\omega_{\mathrm{s}}^{0,0}\right) - D_{v,2}\ \omega_{\mathrm{s}}^{0,0} \tag{4-53}$$

with $D_{c,2} \geq 0$ and $D_{v,2} \geq 0$ diagonal matrices.

Now the total friction D acting on the equations of motion is given by

$$D = \begin{bmatrix} D^{\mathrm{b},0} \\ D^{\mathrm{s},0} \end{bmatrix} \tag{4-54}$$

This subsection will be concluded with a final remark. In reality, the true friction model will not be given by a perfect Coulomb and viscous friction. Furthermore, it will be difficult to find a good estimation of the friction parameters, especially between the sphere and different types of floors. Therefore, in subsequent chapters the friction is sometimes excluded from the equations of motion when designing a controller. However, it will also be shown that some of the designed controllers are robust to this unmodelled friction.

4-5 Summary

In this section the full 3D model of the robot was derived using screw theory and the Euler-Lagrange method. The final equations of motion for the ballbot with the 3 omni-wheel configuration are given by:

$$\begin{cases} \ddot{q}_r &= \bar{M}^{-1}\left(-\bar{C}\dot{q}_r - \bar{G} - D + B\tau^T\right) \\ \dot{\bar{q}} &= \Theta(R_{\mathrm{s}}^0, R_{\mathrm{b}}^0)\ S\ \dot{q}_r \end{cases} \tag{4-55}$$

with the states and input

$$\dot{q} = \begin{bmatrix} \omega_{\mathrm{s}}^{0,0} \\ \omega_{\mathrm{b}}^{0,0} \\ v_{\mathrm{k}}^{0,0} \end{bmatrix} \qquad \dot{q}_r = \begin{bmatrix} \omega_{\mathrm{s}}^{0,0} \\ \omega_{\mathrm{b}}^{0,0} \end{bmatrix} \qquad \bar{q} = \begin{bmatrix} \bar{R}_{\mathrm{s}}^0 \\ \bar{R}_{\mathrm{b}}^0 \\ p_{\mathrm{k}}^0 \end{bmatrix} \qquad \tau^T = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

system matrices

$$\begin{aligned}
\bar{M} &= S^T M(q) S \\
M(q) &= J^T \begin{bmatrix} \mathcal{I}^{0,\mathrm{s}} & 0 \\ 0 & \mathcal{I}^{0,\mathrm{b}} \end{bmatrix} J \\
\bar{C} &= S^T C(q,\dot{q}) S \\
\bar{G} &= S^T G(q) \\
B &= \frac{r_s}{r_w} \begin{bmatrix} I \\ -I \end{bmatrix} R_\mathrm{b}^0 \begin{bmatrix} -\frac{1}{2}\sqrt{2} & \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{2} \\ 0 & -\frac{1}{4}\sqrt{6} & \frac{1}{4}\sqrt{6} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{bmatrix}
\end{aligned}$$

and the transformation matrices

$$J = \begin{bmatrix} I & 0 & 0 \\ \tilde{p}_\mathrm{k}^0 & 0 & I \\ 0 & I & 0 \\ 0 & \tilde{p}_\mathrm{k}^0 & I \end{bmatrix} \quad \Theta(R_\mathrm{s}^0, R_\mathrm{b}^0) = \begin{bmatrix} \Phi(R_\mathrm{s}^0)(R_\mathrm{s}^0)^T & 0 & 0 \\ 0 & \Phi(R_\mathrm{b}^0)(R_\mathrm{b}^0)^T & 0 \\ 0 & 0 & I \end{bmatrix} \quad S = \begin{bmatrix} I & 0 \\ 0 & I \\ \tilde{r} & 0 \end{bmatrix}$$

$$\Phi(R_\mathrm{s}^0) = \left[\begin{array}{ccc|ccc|ccc} 0 & R_{13} & -R_{12} & 0 & R_{23} & -R_{22} & 0 & R_{33} & -R_{32} \\ -R_{13} & 0 & R_{11} & -R_{23} & 0 & R_{21} & -R_{33} & 0 & R_{31} \\ R_{12} & -R_{11} & 0 & R_{22} & -R_{21} & 0 & R_{32} & -R_{31} & 0 \end{array}\right]^T$$

4-6 Discussion

In this section the final equations of motion are briefly discussed. First of all, the workspace of the system was determined to be $SE(3) \times SE(3)$. The configuration space was found to be $SO(3) \times SO(3) \times \mathbb{R}^2$, where the configuration space of the dynamics was reduced to $SO(3) \times SO(3)$. The reduced dynamics consists of 6 degrees of freedom. There are only 3 actuators, and henceforth the system is under-actuated. However, considering the system as two subsystems, i.e. the body and the sphere, either the sphere or body can be considered to be fully actuated. This will be used for the geometric controller.

The equations of motion in (4-55) are coordinate-free and are directly expressed as a function of rotation matrices, translations and corresponding angular and linear velocities. Henceforth, the dynamics are not subjected to singularities, which is possible for the parametrisation with Euler angles, or a double coverage of $SO(3)$ which is the case for quaternions.

Despite the fact that the equations of motion are coordinate -free, it is still possible to chose different frames than the frames defined in this chapter. The equation of motion can be related to another choice of frames using the homogeneous matrices between the chosen frames and the new frames and corresponding adjoint matrices.

In equation (4-55) the equations of motion are expressed in inertia frame. However, it is possible to express them in any other frame using the adjoint matrix and its derivative.

Chapter 5

Geometric Nonlinear control

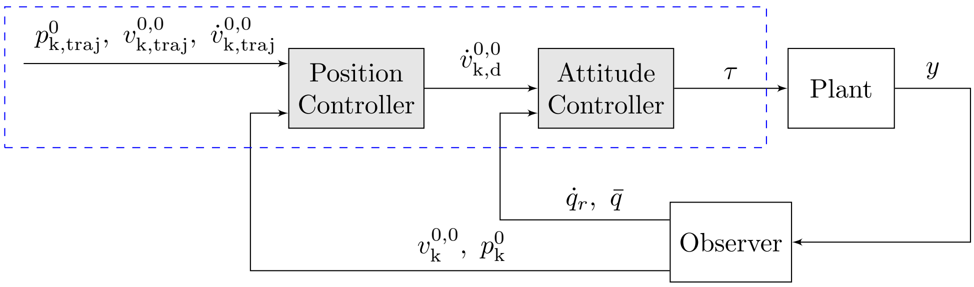


Figure 5-1: Controller structure

In this chapter a nonlinear geometric approach to the control of the ball-balancing robot is proposed. As discussed in Chapter 2 often a linear approach was used to control the robot. However, the linear controllers are synthesised for the linearised model, in which the nonlinear dynamics of the systems are discarded. As a result the controller is only valid in the region close to the linearisation point where the dynamics are approximately linear, while in other regions stability is not assured. Nonlinear control methods have been proposed as well, such as computed torque and sliding mode control. However, these controllers were based on the parametrisation using Euler angles or quaternions. In this chapter a nonlinear controllers are defined using error functions on the underlying manifolds $SO(3)$ or S^2. As a result the controllers remain coordinate-free. Furthermore, the parametrisation of the rotation matrix with Euler angles can result in singularities or a gimbal lock, i.e. for some angles the rotation matrix loses a degree of freedom and the use of quaternions result in a double coverage of $SO(3)$.

In this chapter the highlighted blocks in Figure 5-1 will be discussed, which includes an attitude controller and position controller. It can be seen that a two-loop structure is chosen instead of one overarching controller, similar to the controllers proposed in literature, with the exception of the hierarchical sliding mode controller proposed in [13]. Due to the under-actuated nature of the system, it is challenging to find an unified controller. Instead, it is often chosen to design a two-loop control structure to couple the under-actuated states in the outer-loop to the actuated states in the inner-loop. Other methods proposed for under-actuated systems are for example interconnection and damping assignment passivity based control (IDA PBC) [23] and hierarchical sliding mode control [18]. However, the design of a IDA PBC for under-actuated systems is often challenging. The hierarchical sliding mode controller for a ball-balancing consists of two sliding surfaces for the position and attitude and one overarching surface which is a convex combination of both. This requires two separate control objectives for both the position surface and attitude, e.g. the attitude should be in the upright equilibrium and the robot should move to a certain position. These control objectives act against each other. Hence, instead it is chosen to find a coupling between the position states and the attitude states using the dynamics constraint, as will be explained in Section 5-4.

This chapter will be structured as follows. First of all the control requirements from the problem statement will be formulated formally. Secondly, an error functions for the 2-sphere S^2 will be derived. This section is followed by the design of multiple attitude controllers to control the attitude of the ballbot. The attitude controller section is followed by the design of a position controller, which will a certain desired attitude, given a certain desired position or trajectory. The chapter is concluded with simulation results and a discussion.

5-1 Controller requirements

In the problem statement in Section 1-3 the desired control objectives were stated. In this section these control objectives will be specified in more detail. Note that it is not necessary that a specific controller fulfils all the above control objectives. It is therefore well possible that there are multiple controllers, each for a specific controller objective.

Balancing control The first functionality is to control and keep the body of the robot in the upright position (i.e. the inverted pendulum pose), while there is no desired position for the robot. Furthermore, the body should remain upright, as it cannot pass through the ground.

> *Balancing control:* The attitude of the robot should converge to the upright position
>
> $$R^0_{\mathrm{b}}(t) \to I$$
>
> as $t \to \infty$. Furthermore
>
> $$b_{3,\mathrm{z}}(t) > 0, \ \forall t$$

Position control The functionality of the position controller is to control the robot to a certain position, while maintaining stability. This controller can be used for station-keeping, i.e. maintaining balance while maintaining at a fixed position, or point-to-point control, i.e. control the robot to a certain position. This entails that the robot is able to balance, while maintaining at a certain fixed position.

> *Position control:* Given a certain (initial) position $\begin{bmatrix} x_d & y_d \end{bmatrix}^T$ the following states should converge
>
> $$R_{\text{b}}^0(t) \to I$$
> $$\begin{bmatrix} x(t) & y(t) \end{bmatrix}^T \to \begin{bmatrix} x_d(t) & y_d(t) \end{bmatrix}^T$$
>
> as $t \to \infty$. Furthermore
>
> $$b_{3,\text{z}}(t) > 0, \ \forall t$$

Trajectory tracking The final functionality is trajectory tracking. The tracking controller should be able to track certain predefined trajectory defined by a desired position, velocity and acceleration.

> *Tracking control:* The position of the robot at time t should converge to the trajectory $p_{\text{k,traj}}^0(t)$, $v_{\text{k,traj}}^{0,0}(t)$ and $\dot{v}_{\text{k,traj}}^{0,0}(t)$:
>
> $$p_{\text{k}}^0(t) \to p_{\text{k,traj}}^0(t)$$
> $$v_{\text{k}}^{0,0}(t) \to p_{\text{k,traj}}^0(t)$$
> $$\dot{v}_{\text{k}}^{0,0}(t) \to \dot{v}_{\text{k,traj}}^{0,0}(t)$$
>
> as $t \to \infty$. Furthermore
>
> $$b_{3,\text{z}}(t) > 0, \ \forall t$$

Besides the above control objectives, there are two additional desired functionalities of the controller:

- Disturbance rejection
- Robustness with respect to parametric and model uncertainties

Here the most important parametric and/or model uncertainties come from the friction with the ground and changes in inertial parameters, such as the mass.

5-2 Attitude error function on S^2

Because of the geometric framework, the Euclidean error function $e = x - x_d$ does not always make sense, as the underlining manifold is not linear. In the preliminaries an error function on $SO(3)$ was defined for the error rotation matrix R_e. However, since the robot is omnidirectional, every direction is a valid heading direction, thus the yaw rotation, i.e. a rotation around the z axis of the body frame, does not needs to be controlled. In other words, in order to control the attitude of the robot, we are only interested in controlling the direction of the z axis of the body frame, and not the x and y axis. This direction is given by the third column of $R_{\mathrm{b}}^0 = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}$, hence

$$b_3 = R_{\mathrm{b}}^0 e_3 \tag{5-1}$$

with $e_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. The vector b_3 lies on the 2-sphere $S^2 = \{x \in \mathbb{R}^3 : x^T x = 1\}$. Given a desired and current vector b_3, it would be possible to construct a desired rotation matrix that gives a minimal rotation relative to R_d. Using this constructed desired rotation matrix the error function e_R can be computed. However, in this section an alternative error function is introduced for b_3 on S^2.

Consider the symmetric error function [22, 24]:

$$\phi(b_3, b_{\mathrm{d}}) = 1 - b_{\mathrm{d}}^T b_3 \tag{5-2}$$

This function has a minimum for $b_{\mathrm{d}}^T b_3 = 1$, which corresponds to $b_3 = b_{\mathrm{d}}$, and a maximum for $b_{\mathrm{d}}^T b_3 = -1$, which corresponds to $b_3 = -b_{\mathrm{d}}$. Note that the the minimum and maximum correspond to the case where the two vectors are collinear. The function $\phi(b_3, b_{\mathrm{d}})$ is therefore bounded by $0 \leq \phi(R_e) \leq 2$.

The function $\phi(b_3, b_{\mathrm{d}})$ can be expressed as a function of the attitude error rotation matrix R_e. Using (5-1):

$$\begin{aligned} \phi(b_3, b_{\mathrm{d}}) &= 1 - b_{\mathrm{d}}^T b_3 \\ &= 1 - e_3^T R_d^T R e_3 \\ &= 1 - e_3^T R_e e_3 \end{aligned}$$

Differentiating the error function yields

$$\frac{d}{dt}\phi(b_3, b_{\mathrm{d}}) = -e_3^T R_e \tilde{e}_\omega^{\mathrm{b}} e_3 \tag{5-3}$$

where e_ω^{b} is expressed the **body frame**. Rewriting to inertia frame using $e_\omega^{\mathrm{b}} = \left(R_{\mathrm{b}}^0\right)^T e_\omega^0$ yields:

$$\begin{aligned} \frac{d}{dt}\phi(b_3, b_{\mathrm{d}}) &= -e_3^T R_e \left(R^T e_\omega^0\right)^{\sim} e_3 \\ &= -\underbrace{e_3^T R_d^T}_{b_{\mathrm{d}}^T} \underbrace{R R^T}_{I} \tilde{e}_\omega^0 \underbrace{R e_3}_{b_3} \\ &= b_{\mathrm{d}}^T \tilde{b}_3 e_\omega^0 \\ &= \left(\tilde{b}_{\mathrm{d}} b_3\right)^T e_\omega^0 \end{aligned}$$

Here the properties $\tilde{a}^T = -\tilde{a}$ and $\tilde{a}b = -\tilde{b}a$ are used. The thus given by:

$$\frac{d}{dt}\phi(b_3, b_\mathrm{d}) = \left(\tilde{b}_\mathrm{d} b_3\right)^T e_\omega^0 \tag{5-4}$$

Now the tracking error $e_{b_3} \in \mathbb{R}^3$ on S^2 is chosen to be

$$e_{b_3} = \tilde{b}_\mathrm{d} b_3 \tag{5-5}$$

5-3 Balancing controller

In this section balancing controllers will be designed to control the body of the robot. The two proposed controllers are a computed torque controller and a geometric sliding mode controller. Both controllers use the error functions defined on $SO(3)$ and/or S^2.

Before the design of these controllers, the system dynamics are first rewritten in a different form. Consider the following form of the equations of motion of the system:

$$\ddot{q}_r = -M^{-1}\left(C\dot{q}_r + G\right) + M^{-1}B\tau^T$$

with $\ddot{q}_r = \begin{bmatrix} (\dot{\omega}_\mathrm{s}^{0,0})^T & \left(\dot{\omega}_\mathrm{b}^{0,0}\right)^T \end{bmatrix}^T$. Let us introduce the following partitioning:

$$\begin{bmatrix} f_1(\dot{q}_r, \bar{q}) \\ f_2(\dot{q}_r, \bar{q}) \end{bmatrix} = -M^{-1}\left(C\dot{q}_r + G\right), \qquad \begin{bmatrix} g_1(\dot{q}_r, \bar{q}) \\ g_2(\dot{q}_r, \bar{q}) \end{bmatrix} = M^{-1}B$$

and thus

$$\begin{bmatrix} \dot{\omega}_\mathrm{s}^{0,0} \\ \dot{\omega}_\mathrm{b}^{0,0} \end{bmatrix} = \begin{bmatrix} f_1(\dot{q}_r, \bar{q}) \\ f_2(\dot{q}_r, \bar{q}) \end{bmatrix} + \begin{bmatrix} g_1(\dot{q}_r, \bar{q}) \\ g_2(\dot{q}_r, \bar{q}) \end{bmatrix} \tau^T \tag{5-6}$$

and hence the angular acceleration of the body in inertia frame is given by

$$\dot{\omega}_\mathrm{b}^{0,0} = f_2(\dot{q}_r, \bar{q}) + g_2(\dot{q}_r, \bar{q})\tau^T \tag{5-7}$$

The proposed attitude controllers in this section assume that the variable matrix $g_2(\dot{q}_r, \bar{q})$ is invertible. This section is be structured as follows. First, the inversion of the input map $g_2(\dot{q}_r, \bar{q})$ is proven. Second, the a geometric controller is proposed and finally a sliding mode controller is introduced.

5-3-1 Inversion of $g_2(\dot{q}_r, \bar{q})$

Before the design of the controller, let us prove the investigate the invertability of the input map $g_2(\dot{q}_r, \bar{q})$. The following lemma provides a bound for which this matrix is always invertible.

Lemma 1. *Provided that all system trajectories satisfy $b_{3,\mathrm{z}} \geq 0$, the variable $g_2(\dot{q}_r, \bar{q})$ is always full rank and invertible.*

Proof. First of all $M^{-1}B$ is given by

$$M^{-1}B = \frac{r_s}{r_w} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}^{-1} \begin{bmatrix} I \\ -I \end{bmatrix} R_{\mathrm{b}}^0 \begin{bmatrix} -\frac{1}{2}\sqrt{2} & \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{2} \\ 0 & -\frac{1}{4}\sqrt{6} & \frac{1}{4}\sqrt{6} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{bmatrix} \tag{5-8}$$

Using the inversion formula for block matrices

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BS^{-1}CA^{-1} & -A^{-1}BS^{-1} \\ -S^{-1}CA^{-1} & S^{-1} \end{bmatrix}$$

where S is the Schur complement given by $(D - CA^{-1}B)$, $g_2(\dot{q}_r, \bar{q})$ can be written as

$$g_2(\dot{q}_r, \bar{q}) = -\frac{r_s}{r_w} S^{-1} \left(I + M_{21}M_{11}^{-1}\right) R_{\mathrm{b}}^0 \begin{bmatrix} -\frac{1}{2}\sqrt{2} & \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{2} \\ 0 & -\frac{1}{4}\sqrt{6} & \frac{1}{4}\sqrt{6} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{bmatrix}$$

If $\left(I + M_{21}M_{11}^{-1}\right)$ is invertible, $g_2(\dot{q}_r, \bar{q})$ is a product of all invertible matrices, and hence $g_2(\dot{q}_r, \bar{q})$ itself is invertible. This matrix is given by

$$\left(I + M_{21}M_{11}^{-1}\right) = \begin{bmatrix} c\, b_{3,\mathrm{z}} + 1 & 0 & 0 \\ 0 & c\, b_{3,\mathrm{z}} + 1 & 0 \\ -c\, b_{3,\mathrm{x}} & -c\, b_{3,\mathrm{y}} & 1 \end{bmatrix} \tag{5-9}$$

where $b_{3,\mathrm{i}}$ denotes an element of the last column of R_{b}^0 and c is a system dependent constant given by

$$c = \frac{l\, m_{\mathrm{b}}\, r_s}{j_{\mathrm{s}} + m_{\mathrm{b}}\, {r_s}^2 + m_{\mathrm{s}}\, {r_s}^2} \geq 0$$

For a certain set of system parameters, it could be possible that $\left(I + M_{21}M_{11}^{-1}\right)$ drops rank and is not invertible. If

$$c\, b_{3,\mathrm{z}} = -1 \tag{5-10}$$

$g_2(\dot{q}_r, \bar{q})$ is not invertible. However, by limiting the allowable domain of the controller within $b_{3,\mathrm{z}} \geq 0$, $g_2(\dot{q}_r, \bar{q})$ will always be invertible. □

In the lemma it is assumed that all system trajectories satisfy the bound $b_{3,\mathrm{z}} \geq 0$. This is a reasonable bound on the system trajectories, as the lower bound corresponds to a situation where the body is horizontal. Lower values for $b_{3,\mathrm{z}}$ result in a collision with the floor and are therefore excluded from the allowable system trajectories.

5-3-2 Geometric computed torque control

In this section a computed torque controller to control the attitude of the body is proposed. The computed torque controller is given by

$$\tau^T = g_2^{-1}\left(-f_2 + u\right) \tag{5-11}$$

such that

$$\dot{\omega}_{\mathrm{b}}^{0,0} = u$$

Here $g_2(\dot{q}_r, \bar{q})$ is full rank and invertible, provided that the system trajectories are limited to $b_{3,\mathrm{z}} \geq 0$.

For the remainder of this section the control input u is designed such that the desired attitude of the robot is tracked on either $SO(3)$ or S^2. Attitude tracking on $SO(3)$ implies that a desired rotation matrix is tracked, while tracking on S^2 implies that only the third column of the rotation matrix is tracked (the orientation of the z axis of the body frame).

Attitude tracking on $SO(3)$

Let us first consider tracking on $SO(3)$. Consider the following control input u:

$$u = \dot{\omega}_{\mathrm{d}}^{0,0} - k_p R_{\mathrm{b}}^0 e_R - K_d e_\omega \tag{5-12}$$

with the scalar $k_p > 0$ and matrix $K_d > 0$ and all variables expressed in inertia frame. The error dynamics of the system are then given by

$$e_{\dot{\omega}} + k_p R_{\mathrm{b}}^0 e_R + K_d e_\omega = 0 \tag{5-13}$$

Convergence of the error dynamics is given in the following lemma.

Lemma 2. *Given the error dynamics of equation* (5-13)*,* (e_ω, e_R, R_e) *will converge to* $(0, 0, I)$*, i.e. the system converges to the desired attitude, assuming that the initial conditions satisfy:*

$$\phi(R_e(0)) < 2 \tag{5-14}$$

$$e_\omega^T(0) e_\omega(0) < 2\ k_p \left(2 - \phi(R_e(0))\right) \tag{5-15}$$

Proof. Consider the Lyapunov candidate function

$$V = k_p \phi(R_e) + \frac{1}{2}\left(e_\omega^0\right)^T e_\omega^0 \tag{5-16}$$

where the error functions are expressed in inertia frame. Recall from section 3-3 that $\phi(R_e)$ is bounded by $0 \leq \phi(R_e) \leq 3$ and hence the candidate Lyapunov function is positive definite for $R_e \neq I$. The derivative of the Lyapunov function is given by

$$\begin{aligned}
\dot{V} &= -\frac{1}{2} k_p \mathrm{tr}\left(R_e \tilde{e}_\omega^{\mathrm{b}}\right) + \left(e_\omega^0\right)^T e_{\dot{\omega}}^0 \\
&= k_p \left(e_\omega^{\mathrm{b}}\right)^T e_R + \left(e_\omega^0\right)^T e_{\dot{\omega}}^0 \\
&= k_p \left(\left(e_\omega^0\right)^T R_{\mathrm{b}}^0\right) e_R + \left(e_\omega^0\right)^T e_{\dot{\omega}}^0 \\
&= \left(e_\omega^0\right)^T \left(k_p R_{\mathrm{b}}^0 e_R + \dot{e}_\omega^0\right)
\end{aligned}$$

Note that $\dot{e}_\omega = e_{\dot{\omega}}$, if e_ω is expressed in the **inertia frame** and would not be the case if e_ω was expressed in body frame. Substituting in the error dynamics of (5-13):

$$\dot{V} = -e_\omega^T K_d e_\omega < 0, \quad \forall e_\omega \neq 0 \tag{5-17}$$

Thus $\dot{V}(e_R, 0) < 0$ for an arbitrary e_R. Using LaSalle's theorem, the largest invariant set is found by setting $e_\omega = e_{\dot{\omega}} = 0$ in equation (5-13), resulting in

$$k_p R_{\mathrm{b}}^0 e_R = 0 \tag{5-18}$$

Hence the largest invariant set is $\{e_R \in \mathcal{Q} : e_R = 0\}$. Therefore $(e_\omega, e_R) = (0, 0)$ is asymptotically stable.

The prove is concluded by showing that if e_R is zero and the initial conditions satisfy the conditions in (5-14) and (5-15), R_e converges to I.

Recall that e_R is given by

$$e_R = \frac{1}{2}\left(R_e - R_e^T\right)^\vee$$

There are two solutions to $e_R = 0$, namely $R_e = I$ and $R_e = U^T \mathrm{diag}(1, -1, -1)U$ for some $U \in SO(3)$. The first solution corresponds to the minimum of $\phi(R_e) = 0$ and the second solution to $\phi(R_e) = 2$. Applying the bound of the initial conditions in (5-15) to the candidate Lyapunov function (5-16) and using that $V(t) \leq V(0)$ from (5-17) yields to following inequality:

$$\begin{aligned}
k_p \phi(R_e(t)) &\leq V(t) \leq V(0) < k_p \phi(R_e(0)) + k_p \left(2 - \phi(R_e(0))\right) \\
k_p \phi(R_e(t)) &\leq V(t) \leq V(0) < 2kp \\
\phi(R_e(t)) &\leq \frac{1}{k_p} V(t) \leq \frac{1}{k_p} V(0) < 2
\end{aligned}$$

Hence $\phi(R_e(t))$ remains bounded below 2. As a result the only possible solution to $e_R = 0$ is $R_e = I$, and hence the asymptotic convergence of $(e_\omega, e_R, R_e) = (0, 0, I)$ is proven. □

Attitude tracking on S^2

Now let us find a control input u when it is desired to track the attitude on S^2. This will be done using the error function e_{b_3}, instead of e_R. Consider the following control input u:

$$u = \dot{\omega}_{\mathrm{d}}^{0,0} - k_p e_{b_3} - K_d e_\omega \tag{5-19}$$

with the scalar $k_p > 0$ and matrix $K_d > 0$ and all variables expressed in inertia frame. The error dynamics of the system are then given by

$$e_{\dot{\omega}} + k_p e_{b_3} + K_d e_\omega = 0 \tag{5-20}$$

Converge of the error dynamics is given in the following lemma.

Lemma 3. *Given this error dynamics, the errors* (e_ω, e_{b_3}, b_3) *converge asymptotically to* $(0, 0, b_\mathrm{d})$*, assuming that the initial conditions satisfy:*

$$\phi(b_3(0), b_\mathrm{d}(0)) < 2 \tag{5-21}$$
$$e_\omega^T(0)e_\omega(0) < 2\ k_p\,(2 - \phi(b_3(0), b_\mathrm{d}(0))) \tag{5-22}$$

Proof. Consider the candidate Lyapunov function

$$V = k_p\phi(b_3, b_\mathrm{d}) + \frac{1}{2}e_w^T e_w \tag{5-23}$$

Recall from section 5-2 that $\phi(b_3, b_\mathrm{d})$ is bounded by $0 \leq \phi(b_3, b_\mathrm{d}) \leq 2$ and hence the candidate Lyapunov function is positive definite for $b_3 \neq b_\mathrm{d}$. The derivative of the Lyapunov function is given by

$$\dot{V} = k_p e_{b_3}^T e_\omega + e_\omega^T e_{\dot{\omega}} \tag{5-24}$$

Note that $\dot{e}_\omega = e_{\dot{\omega}}$, if e_ω is expressed in the **inertia frame**. Now substituting the error dynamics of (5-20):

$$\dot{V} = -e_\omega^T K_d e_\omega < 0, \quad \forall e_\omega \neq 0 \tag{5-25}$$

Using LaSalle's theorem, the largest invariant set is found by setting $e_\omega = e_{\dot{\omega}} = 0$ in (5-20), which yields

$$-\,k_p e_{b_3} = 0 \tag{5-26}$$

Hence the largest invariant set is given is $\{e_{b_3} \in \mathcal{Q} : e_{b_3} = 0\}$. Therefore, $(e_\omega, e_{b_3}) = (0, 0)$ is asymptomatically stable.

The prove is concluded by showing that if e_{b_3} is zero and the initial conditions satisfy (5-21) and (5-22), b_3 converges to b_d. This will be very similar to the prove given for $SO(3)$. Recall that e_{b_3} is given by $e_{b_3} = \tilde{b}_\mathrm{d} b_3$. This cross product is zero if the two vectors are collinear, i.e. $b_3 = b_\mathrm{d}$ or $b_3 = -b_\mathrm{d}$. The first solution corresponds to the minimum of $\phi(b_3, b_\mathrm{d}) = 0$ and the second solution to the maximum $\phi(b_3, b_\mathrm{d}) = 2$. Applying the bound of the initial conditions in (5-22) to the candidate Lyapunov function (5-23) and using that $V(t) \leq V(0)$ from (5-25) yields to following inequality:

$$\begin{aligned}
k_p\phi(b_3(t), b_\mathrm{d}(t)) &\leq V(t) \leq V(0) < k_p\phi(b_3(0), b_\mathrm{d}(0)) + k_p\,(2 - \phi(b_3(0), b_\mathrm{d}(0))) \\
k_p\phi(b_3(t), b_\mathrm{d}(t)) &\leq V(t) \leq V(0) < 2kp \\
\phi(b_3(t), b_\mathrm{d}(t)) &\leq \frac{1}{k_p}V(t) \leq \frac{1}{k_p}V(0) < 2
\end{aligned}$$

Hence $\phi(b_3(t), b_\mathrm{d}(t)$ remains bounded below 2. As a result the only possible solution to $e_{b_3} = 0$ is $b_3 = b_\mathrm{d}$, and hence the asymptotic convergence of $(e_\omega, e_{b_3}, b_3) = (0, 0, b_\mathrm{d})$ is proven. □

Note that assumption (5-21) is already satisfied by the imposed bound on $b_{3,\mathrm{z}} \geq 0$ to assure the invertibility of $g_2(\dot{q}_r, \bar{q})$.

Influence of uncertainties and disturbances

In this subsection the influence of uncertainties and disturbances on the geometric computed torque attitude controller is investigated. Since this controller relies on cancelling the dynamics, it can be expected that the influence of uncertainties can have a severe influence on the stability of the system. Let us denote the uncertainties acting on the angular acceleration of the body with d_2. The angular acceleration of the body is then given by:

$$\dot{\omega}_{\mathrm{b}}^{0,0} = f_2(\dot{q}_r, \bar{q}) + g_2(\dot{q}_r, \bar{q})\tau^T + d_2 \tag{5-27}$$

The error dynamics for tracking on $SO(3)$ then become

$$\dot{e}_\omega + k_p R_{\mathrm{b}}^0 e_R + K_d e_\omega = d_2 \tag{5-28}$$

and for S^2

$$\dot{e}_\omega + k_p e_{b_3} + K_d e_\omega = d_2 \tag{5-29}$$

Furthermore, for both types of tracking, the derivative of the candidate Lyapunov function becomes

$$\dot{V} = -e_\omega^T K_d e_\omega + e_\omega^T d_2 \tag{5-30}$$

Given the error dynamics and derivative of the Lyapunov function, it will be very hard to proof stability. For example, let us assume that it is possible to find a K_d such that

$$e_\omega^T K_d e_\omega > e_\omega^T d_2, \quad \forall t$$

Still it is not possible to prove stability, as the largest invariant set found by setting $\dot{e}_\omega = e_\omega = 0$ results in $k_p R_{\mathrm{b}}^0 e_R = d_2$ or $k_p e_{b_3} = d_2$. Only if $d_2 \to 0$ for $t \to \infty$ stability would be obtained.

Summary

In this section a geometric computed torque controller was proposed to control the attitude of the ball-balancing robot. Both tracking on SO(3) and S^2 was considered.

Given the dynamics of the body:

$$\dot{\omega}_{\mathrm{b}}^{0,0} = f_2(\dot{q}_r, \bar{q}) + g_2(\dot{q}_r, \bar{q})\tau^T$$

The computed torque controller is given by

$$\tau^T = g_2^{-1}\left(-f_2 + u\right)$$

Here $g_2(\dot{q}_r, \bar{q})$ is full rank and invertible, provided that the system trajectories are limited to $b_{3,\mathrm{z}} \geq 0$. Depending on the type of tracking, u is given by
Tracking on SO(3):

$$u = \dot{\omega}_{\mathrm{d}}^{0,0} - k_p R_{\mathrm{b}}^0 e_R - K_d e_\omega$$

Under the assumption that the initial conditions satisfy:

$$\phi(R_e(0)) < 2$$
$$e_\omega^T(0)e_\omega(0) < 2\ k_p\,(2 - \phi(R_e(0)))$$

Tracking on S^2:

$$u = \dot{\omega}_{\mathrm{d}}^{0,0} - k_p e_{b_3} - K_d e_\omega$$

Under the assumption that the initial conditions satisfy:

$$\phi(b_3(0), b_{\mathrm{d}}(0)) < 2$$
$$e_\omega^T(0)e_\omega(0) < 2\ k_p\,(2 - \phi(b_3(0), b_{\mathrm{d}}(0)))$$

5-3-3 Geometric sliding mode control

In this section a geometric sliding mode control law for control of the body attitude is proposed, again using the error functions on S^2 and $SO(3)$. The benefit of sliding mode control is that it is more robust to model uncertainties.

The idea behind sliding mode control is to force the system to a certain surface, on which the system converges to the desired state. The control is built up from *equivalent control* and *switching control*. The equivalent control causes the system to slide along the surface, while the switching control forces the system to the surface.

Two sliding surfaces are proposed: one for which the attitude is tracked on SO(3) is tracked, i.e. $R_{\mathrm{b}}^0(t) \to R_{\mathrm{d}}^0$ for $t \to \infty$ and one for which the attitude is tracked on S^2, i.e. $b_3(t) \to b_{\mathrm{d}}$ for $t \to \infty$.

Attitude tracking on SO(3)

The following sliding surface is proposed:

$$S = \alpha e_R + e_\omega^{\mathrm{b}} = 0 \tag{5-31}$$

where $\alpha > 0$. This surface can also be expressed in inertia frame:

$$S = \alpha e_R + R_0^{\mathrm{b}} e_\omega^0 = 0 \tag{5-32}$$

Similar to the computed torque controller, the following assumptions on the initial conditions are made:

$$\phi(R_e(0)) < 2 \tag{5-33}$$

Lemma 4. *If the system is sliding along the sliding surface in equation* (5-31) *(or equivalently* (5-32)*), the error functions e_R and e_ω converge to zero and R_e will converge to the identity matrix, assuming that the the initial condition of the system satisfy* (5-33)*.*

Proof. Given that the system slides along the surface $S = 0$, it follows that

$$e_\omega^{\mathrm{b}} = -\alpha e_R \tag{5-34}$$

Now consider the following Lyapunov function:

$$V_1 = \phi(R_e) \tag{5-35}$$

and its derivative

$$\dot{V}_1 = e_R^T e_\omega^{\mathrm{b}}$$

where e_ω^{b} is expressed in the body frame. Now given equation (5-34), which is satisfied if the system slides along the surface S, $dotV_1$ becomes:

$$\dot{V}_1 = -e_R^T \alpha e_R < 0, \quad \forall e_R \neq 0 \tag{5-36}$$

To complete the prove, LaSalle's theorem is used by setting $e_R = 0$ and using (5-34), resulting in

$$e_\omega^{\mathrm{b}} = R_0^{\mathrm{b}} e_\omega^0 = 0 \tag{5-37}$$

Hence the largest invariant set is given by $\{e_\omega^{\mathrm{b}} \in \mathcal{Q} : e_\omega^{\mathrm{b}} = 0\}$ or $\{e_\omega^0 \in \mathcal{Q} : e_\omega^0 = 0\}$. Hence (e_R, e_ω) converges asymptotically to $(0, 0)$.

The prove is concluded by showing that if e_R is zero and the initial conditions satisfy the condition in (5-33), R_e is equal to I.

As was shown in the prove for the geometric computed torque controller, there are two solutions for R_e if $e_R = 0$. It will be shown that the bound on the initial condition in (5-33) will result in a convergence of $R_e \to I$ as $t \to \infty$. Applying the bound of the initial conditions on the candidate Lyapunov function (5-35) and using that $V_1(t) \leq V_1(0)$ from (5-36), yields to following inequality:

$$\begin{aligned}
\phi(R_e(t)) &= V_1(t) \leq V_1(0) < k_p \phi(R_e(0)) \\
\phi(R_e(t)) &= V_1(t) \leq V_1(0) < 2
\end{aligned}$$

Hence $\phi(R_e(t))$ remains bounded below 2. As a result the only possible solution to $e_R = 0$ is $R_e = I$, and hence the asymptotic convergence of $(e_\omega, e_R, R_e) = (0, 0, I)$ is proven. □

Equivalent control The equivalent control assures that the dynamics of the sliding surface are stable and can be found by solving $\dot{S} = 0$ and solving for the input. The derivative of S is given by:

$$\dot{S} = \alpha \dot{e}_R + R_0^{\mathrm{b}} \dot{e}_\omega^0 - R_0^{\mathrm{b}} \tilde{\omega}_{\mathrm{b}}^{0,0} e_\omega^0 \tag{5-38}$$

with

$$\begin{aligned}
\dot{e}_\omega^0 &= \dot{\omega}_{\mathrm{b}}^{0,0} - \dot{\omega}_{\mathrm{d}}^{0,0} \\
\dot{e}_R &= \frac{d}{dt}\left(\frac{1}{2}\left(R_e - R_e^T\right)^\vee\right) \\
&= \frac{1}{2}\left(R_e \tilde{e}_\omega^{\mathrm{b}} + \tilde{e}_\omega^{\mathrm{b}} R_e^T\right)^\vee \\
&= \frac{1}{2}\left(R_e \left(R_0^{\mathrm{b}} e_\omega^0\right)^\sim + \left(R_0^{\mathrm{b}} e_\omega^0\right)^\sim R_e^T\right)^\vee \\
&= \frac{1}{2}\left(R_e R_0^{\mathrm{b}} \tilde{e}_\omega^0 R_{\mathrm{b}}^0 + R_0^{\mathrm{b}} \tilde{e}_\omega^0 R_{\mathrm{b}}^0 R_e^T\right)^\vee \\
&= \frac{1}{2}\left(R_d^T \tilde{e}_\omega^0 R_{\mathrm{b}}^0 + R_0^{\mathrm{b}} \tilde{e}_\omega^0 R_d\right)^\vee
\end{aligned}$$

Consider again the angular acceleration of the body in inertia frame in the following form:

$$\dot{\omega}_{\mathrm{b}}^{0,0} = f_2(\dot{q}_r, \bar{q}) + g_2(\dot{q}_r, \bar{q})\tau^T \tag{5-39}$$

Solving $\dot{S} = 0$ for τ^T:

$$\begin{aligned}
\alpha \dot{e}_R + R_0^{\mathrm{b}} \dot{e}_\omega^0 - R_0^{\mathrm{b}} \tilde{\omega}_{\mathrm{b}}^{0,0} e_\omega^0 &= 0 \\
\dot{e}_\omega^0 &= -R_{\mathrm{b}}^0 \alpha \dot{e}_R + \tilde{\omega}_{\mathrm{b}}^{0,0} e_\omega^0 \\
\dot{\omega}_{\mathrm{b}}^{0,0} &= \dot{\omega}_{\mathrm{d}}^{0,0} - R_{\mathrm{b}}^0 \alpha \dot{e}_R + \tilde{\omega}_{\mathrm{b}}^{0,0} e_\omega^0 \\
f_2 + g_2 \tau^T &= \dot{\omega}_{\mathrm{d}}^{0,0} - R_{\mathrm{b}}^0 \alpha \dot{e}_R + \tilde{\omega}_{\mathrm{b}}^{0,0} e_\omega^0 \\
\tau^T &= g_2^{-1} \left(\dot{\omega}_{\mathrm{d}}^{0,0} - R_{\mathrm{b}}^0 \alpha \dot{e}_R + \tilde{\omega}_{\mathrm{b}}^{0,0} e_\omega^0 - f_2 \right)
\end{aligned}$$

Hence the equivalent control is given by

$$u_{eq} = g_2^{-1}(\dot{q}_r, \bar{q}) \left(\dot{\omega}_{\mathrm{d}}^{0,0} - R_{\mathrm{b}}^0 \alpha \dot{e}_R + \tilde{\omega}_{\mathrm{b}}^{0,0} e_\omega^0 - f_2(\dot{q}_r, \bar{q}) \right) \tag{5-40}$$

By assuming that $b_{3,\mathrm{z}} \geq 0$ for all system trajectories, the invertibility of g_2 is assured as given by Lemma 1.

Switching control The switching law forces the system to the sliding surface. The switching control is chosen such that $\dot{S} = -\eta \,\mathrm{sgn}(S) - kS$ with $\eta > 0$ and $k > 0$ and sgn() denotes the sign function:

$$\mathrm{sgn}(x) = \begin{cases} -1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x < 0 \end{cases} \tag{5-41}$$

This is accomplished with the following switching law:

$$u_{sw} = g_2^{-1}(\dot{q}_r, \bar{q}) \left(-R_{\mathrm{b}}^0 \left(\eta \,\mathrm{sgn}(S) + kS \right) \right) \tag{5-42}$$

The complete control law is given by the summation of the equivalent and switching control:

$$\tau^T = u_{eq} + u_{sw} \tag{5-43}$$

Theorem 3. *Given the control law*

$$\tau^T = g_2^{-1}(\dot{q}_r, \bar{q}) \left(\dot{\omega}_{\mathrm{d}}^{0,0} - R_{\mathrm{b}}^0 \alpha \dot{e}_R + \tilde{\omega}_{\mathrm{b}}^{0,0} e_\omega^0 - f_2(\dot{q}_r, \bar{q}) - R_{\mathrm{b}}^0 \left(\eta \,\mathrm{sgn}(S) + kS \right) \right) \tag{5-44}$$

the system of (5-39) *converges to the desired states R_d and $\omega_{\mathrm{d}}^{0,0}$, assuming that the the initial condition of the system satisfy* (5-33) *and all system trajectories satisfy $b_{3,\mathrm{z}} \geq 0$.*

Proof. Consider the candidate Lyapunov function

$$V_2 = \frac{1}{2} S^T S \tag{5-45}$$

And its derivative $\dot{V}$:

$$\begin{aligned}\dot{V}_2 &= S^T\dot{S}\\ &= -\eta|S| - kS^TS < 0, \quad \forall S \neq 0\end{aligned}$$

Hence S converges to 0 in *finite time* due to $-\eta\ |S|$. From Lemma 4 it follows that if $S = 0$, also $(e_R, e_\omega, R_e) \to (0, 0, I)$, which concludes the proof. □

Attitude tracking on S^2

In this subsection a geometric sliding mode attitude controller for S^2 is designed. The following sliding surface is proposed:

$$S = \alpha e_{b_3} + e_\omega^0 = 0 \tag{5-46}$$

Similar to the computed torque controller, the following assumptions on the initial conditions are made:

$$\phi(b_3(0), b_\mathrm{d}(0)) < 2 \tag{5-47}$$

Lemma 5. *If the system is sliding along the sliding surface in equation* (5-46) *, the error functions e_{b_3} and e_ω converge to zero and b_3 will converge to b_d, assuming that the the initial condition of the system satisfy* (5-47)*.*

Proof. Given that the system slides along the surface $S = 0$, it follows that

$$e_\omega^0 = -\alpha e_{b_3} \tag{5-48}$$

Now consider the following Lyapunov function:

$$V_1 = \phi(b_3, b_\mathrm{d}) \tag{5-49}$$

and its derivative

$$\dot{V}_1 = e_{b_3}^T e_\omega^0$$

where e_ω^0 is expressed in the inertia frame. Now given equation (5-48), which is satisfied if the system slides along the surface S, $\dot{V}_1$ becomes:

$$\dot{V}_1 = -e_{b_3}^T \alpha e_{b_3} < 0, \quad \forall e_{b_3} \neq 0 \tag{5-50}$$

To complete the prove, LaSalle's theorem is used by setting $e_{b_3} = 0$ and using (5-48), resulting in

$$e_\omega^0 = 0 \tag{5-51}$$

Hence the largest invariant set is given by $\{e_\omega^0 \in \mathcal{Q} : e_\omega^0 = 0\}$. Hence (e_{b_3}, e_ω) converges asymptotically to $(0, 0)$.

The prove is concluded by showing that if e_{b_3} is zero and the initial conditions satisfy the condition in (5-47), b_3 converges to b_d.

As was shown in the prove for the geometric computed torque controller, there are two solutions for b_3 if $e_{b_3} = 0$. It will be shown that given the bound on the initial condition in (5-47) will result in a convergence of $b_3 \to b_\mathrm{d}$ as $t \to \infty$. Applying the bound of the initial conditions on the candidate Lyapunov function (5-49) and using that $V_1(t) \le V_1(0)$ from (5-50) yields to following inequality:

$$\begin{aligned}\phi(b_3(t), b_\mathrm{d}(t)) &= V_1(t) \le V_1(0) < \phi(b_3(0), b_\mathrm{d}(0)) \\ \phi(b_3(t), b_\mathrm{d}(t)) &= V_1(t) \le V_1(0) < 2\end{aligned}$$

Hence $\phi(b_3(t), b_\mathrm{d}(t))$ remains bounded below 2. As a result the only possible solution to $e_{b_3} = 0$ is $b_3 = b_\mathrm{d}$, and hence the asymptotic convergence of $(e_\omega, e_{b_3}, b_3) = (0, 0, b_\mathrm{d})$ is proven. □

Equivalent control Again the equivalent control is found by setting $\dot{S} = 0$ and solving for the input. Differentiating (5-48) results in

$$\dot{S} = \alpha \dot{e}_{b_3} + \dot{e}^0_\omega \tag{5-52}$$

with

$$\begin{aligned}\dot{e}^0_\omega &= \dot{\omega}^{0,0}_\mathrm{b} - \dot{\omega}^{0,0}_\mathrm{d} \\ \dot{e}_{b_3} &= \frac{d}{dt}\left(\tilde{b}_\mathrm{d} b_\mathrm{d}\right) \\ &= \frac{d}{dt}\left((R_d e_3)^\sim R^0_\mathrm{b} e_3\right) \\ &= \left(\omega^{0,0}_\mathrm{d} R_d e_3\right)^\sim b_3 + \tilde{b}_\mathrm{d} \tilde{\omega}^{0,0}_\mathrm{b} R^0_\mathrm{b} e_3 \\ &= \left(\tilde{\omega}^{0,0}_\mathrm{d} b_\mathrm{d}\right)^\sim b_3 + \tilde{b}_\mathrm{d} \tilde{\omega}^{0,0}_\mathrm{b} b_3 \\ &= -\tilde{b}_3 \tilde{\omega}^{0,0}_\mathrm{d} b_\mathrm{d} - \tilde{b}_\mathrm{d} \tilde{b}_\mathrm{d} \omega^{0,0}_\mathrm{b} \\ &= \tilde{b}_3 \tilde{b}_\mathrm{d} \omega^{0,0}_\mathrm{d} - \tilde{b}_\mathrm{d} \tilde{b}_3 \omega^{0,0}_\mathrm{b}\end{aligned}$$

Now solving $\dot{S} = 0$ for τ^T:

$$\begin{aligned}\alpha \dot{e}_{b_3} + \dot{e}^0_\omega &= 0 \\ \dot{e}^0_\omega &= -\alpha \dot{e}_{b_3} \\ \dot{\omega}^{0,0}_\mathrm{b} &= \dot{\omega}^{0,0}_\mathrm{d} - \alpha \dot{e}_{b_3} \\ f_2 + g_2 \tau^T &= \dot{\omega}^{0,0}_\mathrm{d} - \alpha \dot{e}_{b_3} \\ \tau^T &= g_2^{-1}\left(\dot{\omega}^{0,0}_\mathrm{d} - \alpha \dot{e}_{b_3} - f_2\right)\end{aligned}$$

Hence the equivalent control is given by

$$u_{eq} = g_2^{-1}(\dot{q}_r, \bar{q})\left(\dot{\omega}^{0,0}_\mathrm{d} - \alpha \dot{e}_{b_3} - f_2(\dot{q}_r, \bar{q})\right) \tag{5-53}$$

By assuming that $b_{3,\mathrm{z}} \ge 0$ for all system trajectories, the invertibility of g_2 is assured as given by Lemma 1. Note that by bounding the system trajectories to $b_{3,\mathrm{z}} \ge 0$, assumption (5-47) is satisfied.

Switching control The switching law forces the system to the sliding surface. The switching control is chosen such that $\dot{S} = -\eta \, \mathrm{sgn}(S) - kS$ with $\eta > 0$ and $k > 0$. This is accomplished with the following switching law:

$$u_{sw} = g_2^{-1}(\dot{q}_r, \bar{q}) \left(-\eta \, \mathrm{sgn}(S) - kS\right) \tag{5-54}$$

The complete control law is given by the summation of the equivalent and switching control:

$$\tau^T = u_{eq} + u_{sw} \tag{5-55}$$

Theorem 4. *Given the control law*

$$\tau^T = g_2^{-1}(\dot{q}_r, \bar{q}) \left(\dot{\omega}_{\mathrm{d}}^{0,0} - \alpha \dot{e}_{b_3} - f_2(\dot{q}_r, \bar{q}) - \eta \, \mathrm{sgn}(S) - kS\right) \tag{5-56}$$

the system of (5-39) *converges to the desired states* b_{d} *and* $\omega_{\mathrm{d}}^{0,0}$*, assuming that the the initial condition of the system satisfy* (5-33) *and all system trajectories satisfy* $b_{3,\mathrm{z}} \geq 0$

Proof. Consider the candidate Lyapunov function

$$V_2 = \frac{1}{2} S^T S \tag{5-57}$$

And its derivative $\dot{V}$:

$$\begin{aligned} \dot{V}_2 &= S^T \dot{S} \\ &= -\eta |S| - k S^T S < 0, \quad \forall S \neq 0 \end{aligned}$$

Hence S converges to 0 in finite time due to $-\eta|S|$. From Lemma 5 it follows that if $S = 0$, also $(e_R, e_\omega) \to (0, 0)$, which concludes the proof. □

Controller parameters

Now let us briefly evaluate the controller parameters α, k and η. First of all consider the controller parameter for the sliding surface α. This parameter directly influences the convergence of the states, when the system is on the sliding along the surface $S = 0$, as the Lyapunov function V_1 degreases with $\dot{V}_1 = e_R^T \alpha e_R$. As a result, a higher value of α results in a stronger asymptotic convergence, provided that the system slides along the surface $S = 0$.

The parameters k and η influence how fast the system reaches the surface $S = 0$. The switching behaviour $\eta \, \mathrm{sgn}(S)$ results in that the derivative of the Lyapunov function V_2 is always negative definite for $S \neq 0$ with a constant rate. This is important, as an asymptotic convergence would not result in $S = 0$ in finite time. However, a high gain η, and thus fast convergence of V_2 results in *chatter*, i.e. rapid switch around the sliding surface. Chatter can be reduced by choosing a lower value of η. A fast convergence is then obtained by introducing the additional switching control term $\dot{S} = -kS$. A higher value of k results in a higher rate of convergence. However, the effect of this switching behaviour is always asymptotic, thus the actual surface is never reached exactly by using this switching law alone. Therefore the combination of both $-kS$ and $\eta \mathrm{sgn}(S)$

is made, as the former results in fast asymptotic convergence and the latter results in that the surface is reached in finite time. Finally, note that increasing k and η does not always results in faster convergence of the error functions, since these parameters affect how fast the sliding surface is reached and not the rate of convergence on the surface. Of course, choosing low values for k and η imply slow convergence to the sliding surface and hence convergence to the desired states will be slower. In short, low values of k and η can slow down the convergence to the desired states, but increasing k and η can only decrease the time of convergence up to a certain limit.

As mentioned before, the switching behaviour $-\eta\,\mathrm{sgn}(S)$ will cause chattering of the system. In order to reduce the chattering the sign function of the ideal sliding mode controller can be replaced by a saturation function. For this function, within a small boundary around the sliding surface, the sign function is replaced by a asymptotic switching law. The saturation function is defined by:

$$\mathrm{sat}(x) = \begin{cases} -1 & \text{if } x > \Delta \\ \dfrac{x}{\Delta} & \text{if } |x| \leq \Delta \\ 1 & \text{if } x < -\Delta \end{cases} \tag{5-58}$$

where Δ denotes the thickness of the boundary layer around the sliding surface. Using this function will decreases the chattering around the surface, but it cannot be assured that S becomes exactly zero in finite time.

Influence of uncertainties and disturbances

In this subsection the influence of uncertainties and disturbances on the geometric sliding mode attitude controller is investigated. Let us denote the uncertainties acting on the angular acceleration of the body with d_2. The angular acceleration of the body is then given by:

$$\dot{\omega}_{\mathrm{b}}^{0,0} = f_2(\dot{q}_r, \bar{q}) + g_2(\dot{q}_r, \bar{q})\tau^T + d_2 \tag{5-59}$$

Depending on the type of attitude tracking, applying the control law in (5-44) or (5-56) to the corresponding derivative of their sliding surface will yield in to

$$\dot{S} = -\eta\,\mathrm{sgn}(S) - kS + d_2^* \tag{5-60}$$

where $d_2^* = R_0^{\mathrm{b}} d_2$ for the attitude controller for $SO(3)$ and $d_2^* = d_2$ for the attitude controller for S^2. Hence the derivative of their candidate Lyapunov functions becomes

$$\dot{V}_2 = -\eta|S| - kS^TS + S^Td_2^* \tag{5-61}$$

Let us define $D = \|d_2^*\|_\infty$, which is the maximum disturbance. Using this maximum disturbance, we can define an upper bound for $\dot{V}_2$

$$\dot{V}_2 \leq -kS^TS - (\eta - D)|S| \tag{5-62}$$

hence if η can be chosen such that $\eta > D$, the derivative of V_2 will be negative definite for all $S \neq 0$, and hence the disturbance can be rejected. Note that if the uncertainty or disturbance would be depended of the states of the system, it might not be possible to find and η higher than D, as D also becomes larger for a higher value of η.

Summary

In this section a sliding mode controller was proposed to control the attitude of the ball-balancing robot. Both tracking on SO(3) and S^2 was considered.

Given the dynamics of the body:

$$\dot{\omega}_{\mathrm{b}}^{0,0} = f_2(\dot{q}_r, \bar{q}) + g_2(\dot{q}_r, \bar{q})\tau^T$$

Here $g_2(\dot{q}_r, \bar{q})$ is full rank and invertible, provided that the system trajectories are limited to $b_{3,\mathrm{z}} \geq 0$.

Depending on the type of tracking, the sliding mode controller is given by

Tracking on SO(3):

$$\tau^T = g_2^{-1}(\dot{q}_r, \bar{q}) \left(\dot{\omega}_{\mathrm{d}}^{0,0} - R_{\mathrm{b}}^0 \alpha \dot{e}_R + \tilde{\omega}_{\mathrm{b}}^{0,0} e_\omega^0 - f_2(\dot{q}_r, \bar{q}) - R_{\mathrm{b}}^0 \left(\eta \,\mathrm{sgn}(S) + kS\right)\right)$$

Under the assumption that the initial conditions satisfy:

$$\phi(R_e(0)) < 2$$

Tracking on S^2:

$$\tau^T = g_2^{-1}(\dot{q}_r, \bar{q}) \left(\dot{\omega}_{\mathrm{d}}^{0,0} - \alpha \dot{e}_{b_3} - f_2(\dot{q}_r, \bar{q}) - \eta \,\mathrm{sgn}(S) - kS\right)$$

Under the assumption that the initial conditions satisfy:

$$\phi(b_3(0), b_{\mathrm{d}}(0)) < 2$$

Furthermore, the influence of chattering can be reduced by replacing the sign function by a saturation function.

5-4 Position control: geometric shape-space planner

In the previous section we derived controllers to track a certain attitude of the ballbot body. In this section a simplified version of the shape-space planner of [11] is adapted to a geometric framework. This planner finds a desired attitude that needs to be tracked in order to control the robot to a certain position.

Desired position to accelerations The first step is to find a desired linear acceleration $\dot{v}_{\mathrm{k,d}}^{0,0}$. Given a desired position, velocity and acceleration, tracking the following linear acceleration would achieve the system to converge to the desired states:

$$\dot{v}_{\mathrm{k,d}}^{0,0} = -C_p e_p - C_d e_v + \dot{v}_{\mathrm{k,traj}}^{0,0} \tag{5-63}$$

with $C_p > 0$, $C_d > 0$ and the error functions

$$\begin{aligned} e_p &= p_{\mathrm{k}}^0 - p_{\mathrm{k,traj}}^0 \\ e_v &= v_{\mathrm{k}}^{0,0} - v_{\mathrm{k,traj}}^{0,0} \end{aligned}$$

and $p_{\mathrm{k,traj}}^0$, $v_{\mathrm{k,traj}}^{0,0}$ and $\dot{v}_{\mathrm{k,traj}}^{0,0}$ denote the desired position, linear velocity and the desired linear acceleration respectively corresponding to the trajectory.

Proof. Consider the candidate Lyapunov function

$$V = \frac{1}{2} e_p^T C_p e_p + \frac{1}{2} e_v^T e_v \tag{5-64}$$

and its derivative

$$\dot{V} = e_v^T C_p e_p + e_v^T e_{\dot{v}} \tag{5-65}$$

Given equation (5-63), $e_{\dot{v}}$ is given by

$$e_{\dot{v}} = -C_p e_p - C_d e_v \tag{5-66}$$

This results in the following derivative

$$\begin{aligned} \dot{V} &= e_v^T \left(C_p e_p + e_{\dot{v}} \right) \\ &= -e_v^T C_d e_v < 0, \quad \forall e_v \neq 0 \end{aligned}$$

Using LaSalle's theorem, if $e_v = 0$, $e_{\dot{v}} = 0$, it follows from equation (5-66) that also $e_p = 0$, hence the largest invariant set is given by $\{e_p \in \mathcal{Q} : e_p = 0\}$. Hence the set $(e_p, e_v) = (0, 0)$ is asymptotically stable. $\square$

Note that point-to-point control can be considered as a special form of trajectory tracking, where the position $p_{\mathrm{k,traj}}^0$ is set to be the constant desired position and the desired velocity and acceleration are zero.

The desired linear acceleration is expressed as a function of the desired position trajectory. However, the equations of motion where expressed in the angular velocities of the body and sphere. The linear acceleration $\dot{v}_{\mathrm{k}}^{0,0}$ is related to the angular acceleration $\dot{\omega}_{\mathrm{s}}^{0,0}$ by differentiating the rolling constraint:

$$\begin{aligned} v_{\mathrm{k}}^{0,0} &= \tilde{r} \omega_{\mathrm{s}}^{0,0} = \begin{bmatrix} 0 & r_s & 0 \\ -r_s & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \omega_{\mathrm{s}}^{0,0} \\ \dot{v}_{\mathrm{k}}^{0,0} &= \tilde{r} \dot{\omega}_{\mathrm{s}}^{0,0} \end{aligned}$$

which results in the following relation

$$\begin{bmatrix} \dot{\omega}_{\mathrm{s,x}}^{0,0} \\ \dot{\omega}_{\mathrm{s,y}}^{0,0} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{r_s} \\ \frac{1}{r_s} & 0 \end{bmatrix} \begin{bmatrix} \dot{v}_{\mathrm{k,x}}^{0,0} \\ \dot{v}_{\mathrm{k,y}}^{0,0} \end{bmatrix} \tag{5-67}$$

Note that $\dot{\omega}_{\mathrm{s,z}}^{0,0}$ does not influence the position of the robot and hence no desired $\dot{\omega}_{\mathrm{s,d,z}}^{0,0}$ is imposed on the system.

Angular acceleration to body attitude Given a desired position, a desired angular acceleration $\dot{\omega}_{\text{s,d}}^{0,0}$ is found. The next step is to find a desired body attitude which, when tracked, yields the desired angular velocity, which in turn lets the system converge to the desired position. In order to find a relation between the angular acceleration and the attitude, the equations of motion from (4-55) are rewritten. First of all, consider the following partitioning.

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \dot{\omega}_{\text{s}}^{0,0} \\ \dot{\omega}_{\text{b}}^{0,0} \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \omega_{\text{s}}^{0,0} \\ \omega_{\text{b}}^{0,0} \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} B \\ -B \end{bmatrix} \tau^T$$

Using this partition, the equations of motion can be rewritten as

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{11}+M_{21} & M_{12}+M_{22} \end{bmatrix} \begin{bmatrix} \dot{\omega}_{\text{s}}^{0,0} \\ \dot{\omega}_{\text{b}}^{0,0} \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{11}+C_{21} & C_{12}+C_{22} \end{bmatrix} \begin{bmatrix} \omega_{\text{s}}^{0,0} \\ \omega_{\text{b}}^{0,0} \end{bmatrix} + \begin{bmatrix} G_1 \\ G_1+G_2 \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix} \tau^T$$

and hence the following dynamic constraint on the system is found:

$$\begin{bmatrix} M_{11}+M_{21} & M_{12}+M_{22} \end{bmatrix} \begin{bmatrix} \dot{\omega}_{\text{s}}^{0,0} \\ \dot{\omega}_{\text{b}}^{0,0} \end{bmatrix} + \begin{bmatrix} C_{11}+C_{21} & C_{12}+C_{22} \end{bmatrix} \begin{bmatrix} \omega_{\text{s}}^{0,0} \\ \omega_{\text{b}}^{0,0} \end{bmatrix} + (G_1+G_2) = 0 \tag{5-68}$$

Thus

$$\dot{\omega}_{\text{s}}^{0,0} = -\left(M_{11}+M_{21}\right)^{-1} \Big(\left(M_{12}+M_{22}\right) \dot{\omega}_{\text{b}}^{0,0} + \left(C_{11}+C_{21}\right) \omega_{\text{s}}^{0,0} + \left(C_{12}+C_{22}\right) \omega_{\text{b}}^{0,0} + G_1 + G_2 \Big) \tag{5-69}$$

Now assuming a fast attitude controller, i.e. $\dot{\omega}_{\text{b}}^{0,0}$ and $\omega_{\text{b}}^{0,0}$ converge quickly to zero and hence $\dot{\omega}_{\text{b}}^{0,0} = 0$, $\omega_{\text{b}}^{0,0} = 0$, yields a relation between the angular acceleration and the third column of the rotation body rotation matrix R_{b}^{0}:

$$\dot{\omega}_{\text{s}}^{0,0} = f(b_3) = -\left(M_{11}+M_{21}\right)^{-1} \left(G_1+G_2\right) \tag{5-70}$$

where $R_{\text{b}}^{0} = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}$. Note that also $\left(C_{11}+C_{21}\right) \omega_{\text{s}}^{0,0} = 0$, as $\left(C_{11}+C_{21}\right) = 0$ if $\omega_{\text{b}}^{0,0} = 0$.

As seen before, for the position control only desired values for $\dot{\omega}_{\text{s,x}}^{0,0}$ and $\dot{\omega}_{\text{s,y}}^{0,0}$ where defined. The found relation gives the angular acceleration of the sphere given the attitude of the body. However, the inverse of this relation is desired, i.e. desired attitude vector b_3, given a certain desired angular acceleration. Furthermore, solutions of this inverse should satisfy the fact that the attitude vector b_3 is defined on the manifold $S^2 = \{x \in \mathbb{R}^3 : \|x\| = 1\}$.

Inverting the relationship between the desired angular acceleration and the attitude The solution to the inversion of the nonlinear relation

$$\dot{\omega}_{\text{s,d}}^{0,0} = f(b_3) \tag{5-71}$$

has often two solutions, namely $b_{3,\text{z}} > 0$ and $b_{3,\text{z}} < 0$. Physically this means that the same acceleration of the sphere can be achieved if the pendulum is either in the upright half of the sphere or at the exact opposite position on the sphere.. Of course, we are only interested in solution with $b_{3,\mathrm{z}} > 0$, as the body of the robot cannot pass through the floor. Given the property of a vector on S^2:

$$1 = \sqrt{(b_{3,\mathrm{x}})^2 + (b_{3,\mathrm{y}})^2 + (b_{3,\mathrm{z}})^2}$$

$b_{3,\mathrm{z}}$ can be expressed as

$$b_{3,\mathrm{z}} = \pm\sqrt{1 - (b_{3,\mathrm{x}})^2 - (b_{3,\mathrm{y}})^2}$$

Excluding the solutions $b_{3,\mathrm{z}} < 0$, results in the following solution space for b_3

$$b_3 = \begin{bmatrix} b_{3,\mathrm{x}} \\ b_{3,\mathrm{y}} \\ \sqrt{1 - (b_{3,\mathrm{x}})^2 - (b_{3,\mathrm{y}})^2} \end{bmatrix} \tag{5-72}$$

Hence the relation between the angular acceleration of the sphere and the attitude can be expressed as $\dot{\omega}^{0,0}_{\mathrm{s,d}} = f(b_{3,\mathrm{x}}, b_{3,\mathrm{y}})$.

Let us use the short hand $\dot{\omega}_\mathrm{d} = \begin{bmatrix}\dot{\omega}^{0,0}_{\mathrm{s,x,d}} & \dot{\omega}^{0,0}_{\mathrm{s,y,d}}\end{bmatrix}^T$ and $b_\mathrm{d} = \begin{bmatrix}b_{3,\mathrm{x,d}} & b_{3,\mathrm{y,d}}\end{bmatrix}^T$. The relation between $\dot{\omega}_\mathrm{d}$ and b_d is inverted by approximating the relation with a first order Taylor expansion and linear inversion. Due to this approximation, solutions of b_3 might not lie on S^2, such that the approximated solutions need to be cast back on S^2.

The inversion is done in the following systematic manner. First of all the relation of (5-71) is linearised around the upright position $b_{3,\mathrm{xy}} = \begin{bmatrix}0 & 0\end{bmatrix}^T$:

$$\dot{\omega}_\mathrm{d} = f(b_{3,\mathrm{xy}}) + \underbrace{\frac{df(b_\mathrm{d})}{db_\mathrm{d}}}_{A}(b_\mathrm{d} - b_{3,\mathrm{xy}}) \tag{5-73}$$

and the inversion is given by

$$b_\mathrm{d} = A^{-1}\left(\dot{\omega}_\mathrm{d} - f(b_{3,\mathrm{xy}})\right) + b_{3,\mathrm{xy}} \tag{5-74}$$

$$b_{3,\mathrm{z,d}} = \pm\sqrt{1 - (b_{3,\mathrm{x,d}})^2 - (b_{3,\mathrm{y,d}})^2} \tag{5-75}$$

Obviously, as $f(b_\mathrm{d})$ is nonlinear and a linearisation around $b_{3,\mathrm{xy}}$ might be inaccurate if $b_{3,\mathrm{xy}}$ is not in the neighbourhood of b_d. To increase the accuracy, the desired attitude can be improved iteratively by setting the point of linearisation to be $b_{3,\mathrm{xy}} = b_\mathrm{d}$ and computing (5-74) again until $b_\mathrm{d}(k+1) \approx b_\mathrm{d}(k)$, where k denotes the number of iterations. However, note that this iterative process is only valid if $f(b_{3,\mathrm{xy}})$ is monotonously increasing or decreasing.

Saturation of $\dot{\omega}_\mathrm{d}$ The (iterative) relation of equation (5-74) introduces the problem that for non-feasible values of $\dot{\omega}_\mathrm{d}$ result in non-feasible solutions to b_3 that do not lie on S^2. In order to prevent this, $\dot{\omega}_\mathrm{d}$ is saturated by $\dot{\omega}_\mathrm{max}$ using the maximum allowable tilt angle. The maximum tilt angle related directly to b_3, from which $\dot{\omega}_\mathrm{max}$ can be computed.

Example 5-4.1 (Maximum angular velocity)**.** Assuming a maximum tilt angle of $\frac{\pi}{4}$, $b_{3,\mathrm{z}}$ is given by

$$b_{3,\mathrm{z}} = \cos\left(\frac{\pi}{4}\right) = \frac{1}{2}\sqrt{2}$$

and thus

$$\|b_{3,\mathrm{xy}}\| = \frac{1}{2}\sqrt{2}$$

Assuming a symmetrical robot, the maximum acceleration in x and y direction are equal, hence $b_{3,\mathrm{max}}$ can be found by setting $b_{3,\mathrm{y}} = 0$, resulting in

$$b_{3,\mathrm{max}} = \begin{bmatrix} \frac{1}{2}\sqrt{2} \\ 0 \\ \frac{1}{2}\sqrt{2} \end{bmatrix}$$

and the maximum angular velocity can be obtained by

$$\|\dot{\omega}_{\mathrm{max}}\| = \|f(b_{3,\mathrm{max}})\|$$

From the maximum angular velocity, the actual desired angular velocity can be found using:

$$\dot{\omega}_{\mathrm{max}} = f(b_{3,\mathrm{max}}) \tag{5-76}$$

$$\begin{cases} \dot{\omega}_{\mathrm{d}}^{*} = \dot{\omega}_{\mathrm{d}} & \text{if } \|\dot{\omega}_{\mathrm{d}}\| < \|\dot{\omega}_{\mathrm{max}}\| \\ \dot{\omega}_{\mathrm{d}}^{*} = \dfrac{\dot{\omega}_{\mathrm{d}}}{\|\dot{\omega}_{\mathrm{d}}\|}\|\dot{\omega}_{\mathrm{max}}\| & \text{if } \|\dot{\omega}_{\mathrm{d}}\| > \|\dot{\omega}_{\mathrm{max}}\| \end{cases} \tag{5-77}$$

Tracking on $SO(3)$ Thus far a desired attitude vector on S^2 is returned, given a desired position or trajectory, which can be used as input for the attitude tracking controller on S^2. However, although the robot is omnidirectional, it might be desired that the robot has a certain heading direction $b_{1,d}$. A desired rotation matrix can than be constructed using

$$b_2' = \frac{b_{3,d} \times b_{1,d}}{\|b_{3,d} \times b_{1,d}\|}$$

$$b_1' = \frac{b_2 \times b_{3,d}}{\|b_2 \times b_{3,d}\|}$$

$$R_d = \begin{bmatrix} b_1' & b_2' & b_{3,d} \end{bmatrix}$$

Summary

Given a desired position p_{k}^{0} and linear velocity $v_{\mathrm{k}}^{0,0}$, a desired linear velocity can be given by

$$\dot{v}_{\mathrm{k,d}}^{0,0} = -C_p\left(p_{\mathrm{k}}^{0} - p_{\mathrm{k,traj}}^{0}\right) - C_d\left(v_{\mathrm{k}}^{0,0} - v_{\mathrm{k,traj}}^{0,0}\right) + \dot{v}_{\mathrm{k,traj}}^{0,0}$$

This linear acceleration is related to the angular acceleration by

$$\dot{\omega}_{\mathrm{d}} = \begin{bmatrix} 0 & -\frac{1}{r_s} \\ \frac{1}{r_s} & 0 \end{bmatrix} \begin{bmatrix} \dot{v}^{0,0}_{\mathrm{k,x}} \\ \dot{v}^{0,0}_{\mathrm{k,y}} \end{bmatrix}$$

From the under-actuated nature of the system, the following relation can be found

$$\dot{\omega}^{0,0}_{\mathrm{s}} = -\left(M_{11} + M_{21}\right)^{-1} \Big(\left(M_{12} + M_{22}\right) \dot{\omega}^{0,0}_{\mathrm{b}} + \\ \left(C_{11} + C_{21}\right) \omega^{0,0}_{\mathrm{s}} + \left(C_{12} + C_{22}\right) \omega^{0,0}_{\mathrm{b}} + G_1 + G_2 \Big)$$

which relates the angular velocity the a desired attitude:

$$\dot{\omega}_{\mathrm{d}} = f(b_{\mathrm{d}})$$

The inversion of this relation is found by the inversion of a first order Taylor expansion:

$$b_{\mathrm{d}} = A^{-1} \left(\dot{\omega}_{\mathrm{d}} - f(b_{3,\mathrm{xy}})\right) + b_{3,\mathrm{xy}}$$
$$b_{3,\mathrm{z,d}} = \pm\sqrt{1 - \left(b_{3,\mathrm{x,d}}\right)^2 - \left(b_{3,\mathrm{y,d}}\right)^2}$$

By iteratively linearising around subsequent desired attitudes, a better solution is found from the approximated inverted relation. Finally, in order to ensure feasible desired attitude vectors on S^2, the desired angular acceleration is saturated, using the maximum allowed attitude:

$$\dot{\omega}_{\max} = f(b_{3,\max})$$
$$\begin{cases} \dot{\omega}^*_{\mathrm{d}} = \dot{\omega}_{\mathrm{d}} & \text{if } \|\dot{\omega}_{\mathrm{d}}\| < \|\dot{\omega}_{\max}\| \\ \dot{\omega}^*_{\mathrm{d}} = \dfrac{\dot{\omega}_{\mathrm{d}}}{\|\dot{\omega}_{\mathrm{d}}\|}\|\dot{\omega}_{\max}\| & \text{if } \|\dot{\omega}_{\mathrm{d}}\| > \|\dot{\omega}_{\max}\| \end{cases}$$

5-5 Simulation results

In this section the effectiveness of the proposed controllers will be shown in simulation.

5-5-1 Attitude control

First of all let us consider the attitude control without position control. The desired attitude is set to be $R_d = I$, $b_{\mathrm{d}} = e_3$ and $\omega^{0,0}_{\mathrm{d}} = 0_{3\times1}$ The system is initialised with an initial rotation of

$$R^0_{\mathrm{b}}(0) = R_{\mathrm{z}''}(\theta_3) R_{\mathrm{y}'}(\theta_2) R_{\mathrm{x}}(\theta_1)$$

with

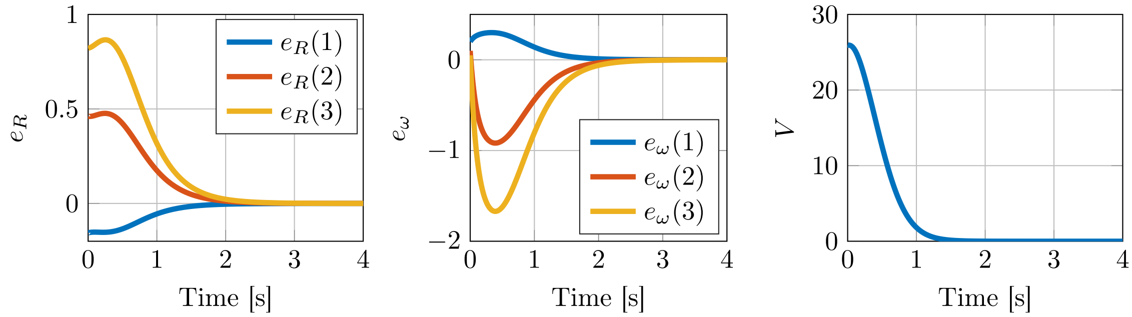
$$R_{\mathrm{x}}(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{bmatrix}, \quad R_{\mathrm{y}'}(\theta_2) = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix}, \quad R_{\mathrm{z}''}(\theta_3) = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

with $\theta_1 = \frac{30}{180}\pi$ rad, $\theta_2 = \frac{40}{180}\pi$ rad, $\theta_3 = \frac{110}{180}\pi$ rad. The initial angular velocities are chosen to be $\omega_{\mathrm{b}}^{0,0}(0) = \begin{bmatrix}0.2 & 0.1 & 0.05\end{bmatrix}^T$ rad/s. Note that this parametrisation of the rotation matrix is only done for the sole purpose of initialising the system with a viable initial rotation matrix and the controllers remain coordinate-free. For the purpose of simulation no friction is present. In the next subsection the influence of unmodelled friction will be shown.

The control parameters of the computed torque controller are set to be $k_p = 20$, $K_d = 10I$. The control parameters of the sliding mode controller are set to be $a_1 = 3$, $k = 30$ $\eta = 0.1I$, $\Delta = 0.05$. These parameters result in similar performances for both controllers.

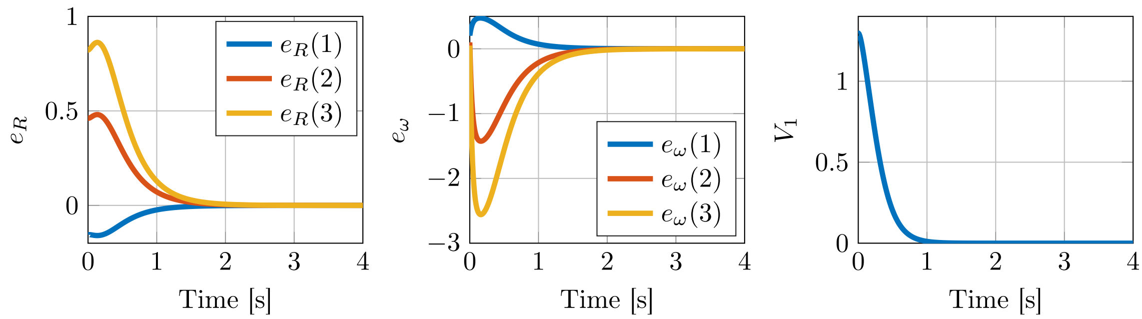
Difference between computed torque and sliding mode First of all, let us compare the difference between the computed torque and sliding mode controller for the case when there is perfect knowledge of the system. In Figure 5-2 and 5-4 the simulation results for the computed torque controller are shown for $SO(3)$ and S^2 respectively and in Figure 5-3 and 5-5 the simulation results for the sliding mode controller for $SO(3)$ and S^2 respectively. It can be observed that for both controllers very similar trajectories of the error functions are obtained, given the used controller parameters. However, for the sliding mode controller the error functions change more aggressively in the initial start up period, i.e. the time in which $S \gg 0$. Due to the high gain k the start up behaviour of the sliding mode controller is more aggressive. When the system is sliding alone the sliding surface, the error functions of the sliding mode controller evolve in a similar manner as the computed torque controller. Note that if the gain η for the switching control signal of the sliding mode controller would be chosen relatively large, a same result as a high value of k would be obtained. However, this would result in a high frequency switching around $S = 0$.

Differences between $SO(3)$ and S^2 Now let us consider the differences between $SO(3)$ and S^2. Figure 5-2 and 5-3 show the evolution of the error functions for attitude tracking on $SO(3)$, using either a computed torque or sliding mode controller, and Figure 5-4 and 5-5 show the evolution of the error functions for attitude tracking on S^2. The tracking on $SO(3)$ involves also tracking a specific the heading direction, while tracking on S^2 only tracks the orientation of the z axis of the body frame. As a result, all three elements of e_R start non-zero, while for tracking on S^2 only $e_{b_3}(1)$ and $e_{b_3}(2)$ are non-zero. This difference in the value of the error function indicates the difference between controlling the yaw rotation or not. This difference between the tracking types has also influence on the position trajectory of the robot. In Figure 5-6a and 5-6 the tracking on $SO(3)$ and S^2 are shown respectively, using the computed torque attitude controller. Since the attitude tracking on S^2 has no influence on the yaw, the robot is rotated around the axis of minimal rotation and hence the control is in only one direction, resulting in a straight trajectory. However, when tracking on $SO(3)$, also the heading direction is controlled, resulting in a control signal around multiple axes. As a result, the trajectory of the ball becomes curved, as shown in Figure 5-6a.

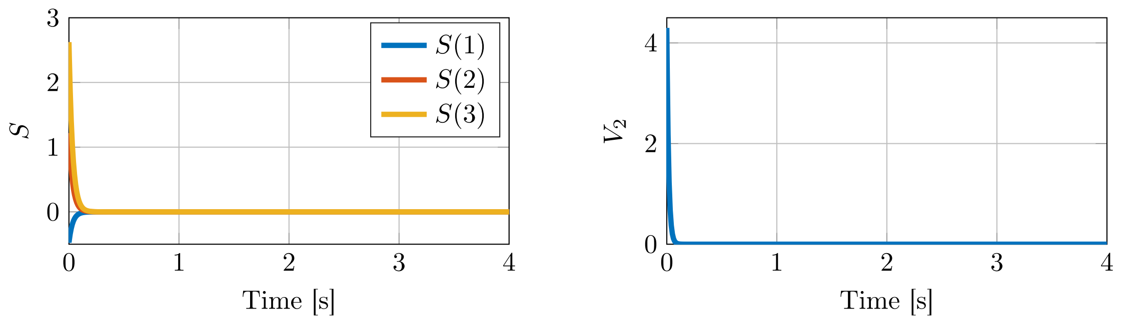


(a) Attitude error **(b)** Angular velocity error **(c)** Lyapunov function

Figure 5-2: Attitude tracking of the geometric computed torque controller on $SO(3)$

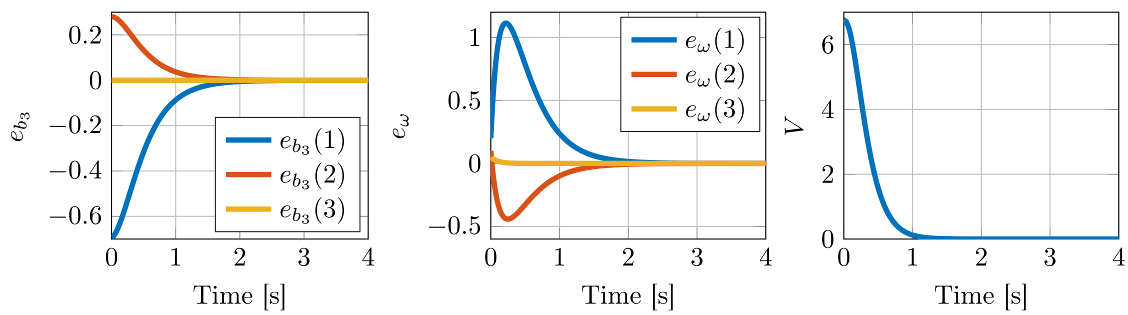


(a) Attitude error **(b)** Angular velocity error **(c)** Lyapunov function



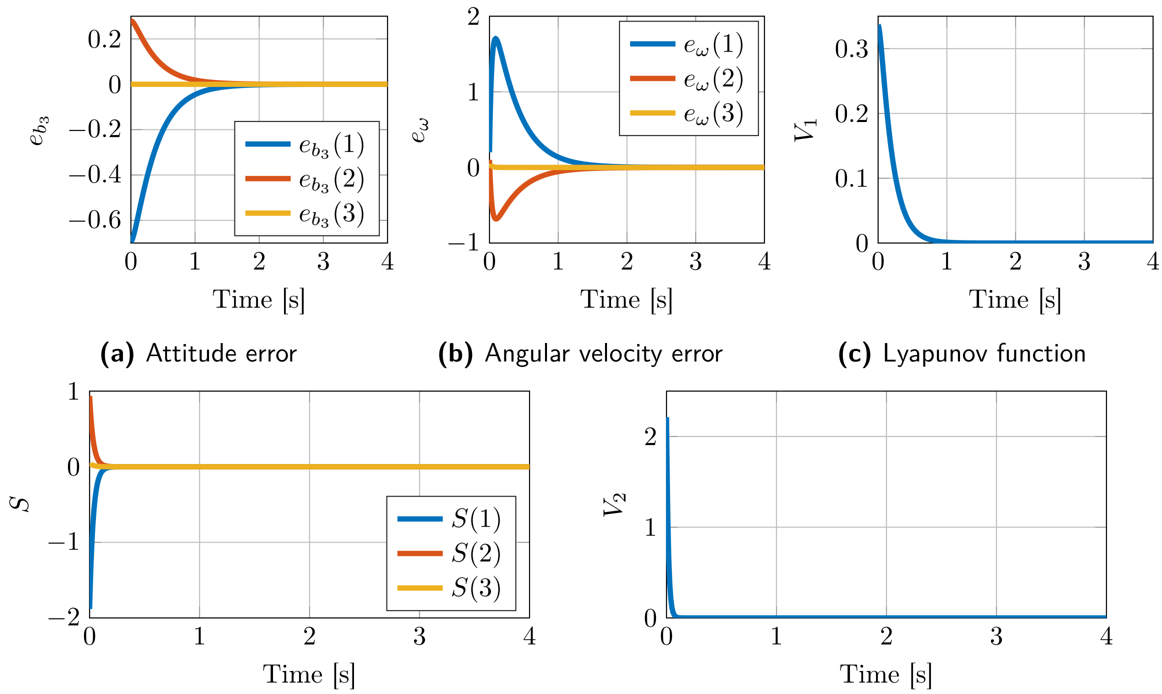
(d) Sliding surface **(e)** Lyapunov function of the sliding surface

Figure 5-3: Attitude tracking of the geometric sliding mode controller on $SO(3)$



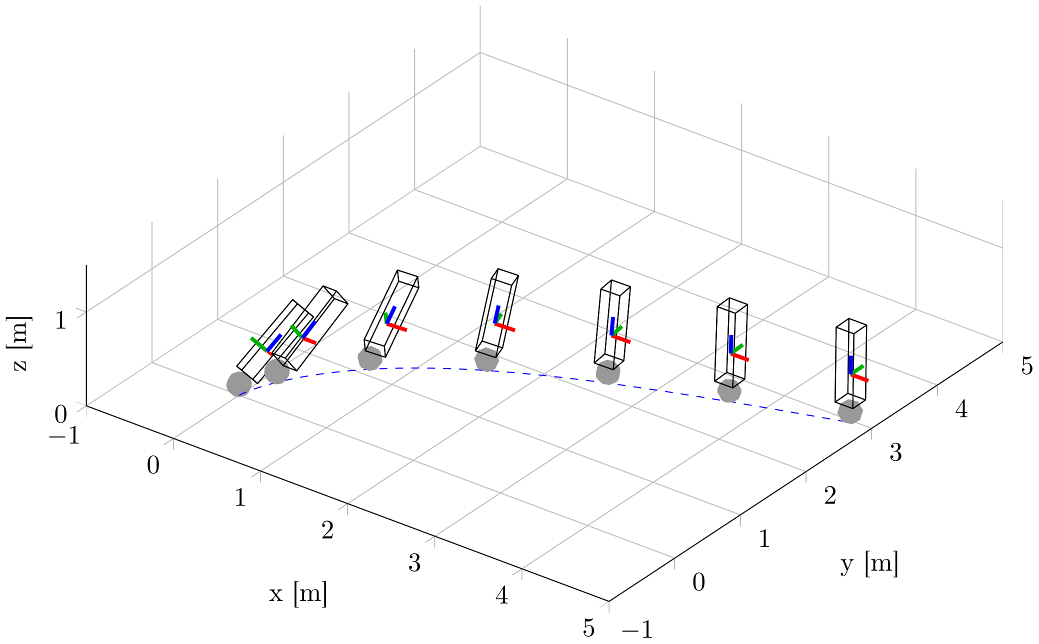
(a) Attitude error **(b)** Angular velocity error **(c)** Lyapunov function

Figure 5-4: Attitude tracking of the geometric computed torque controller on S^2

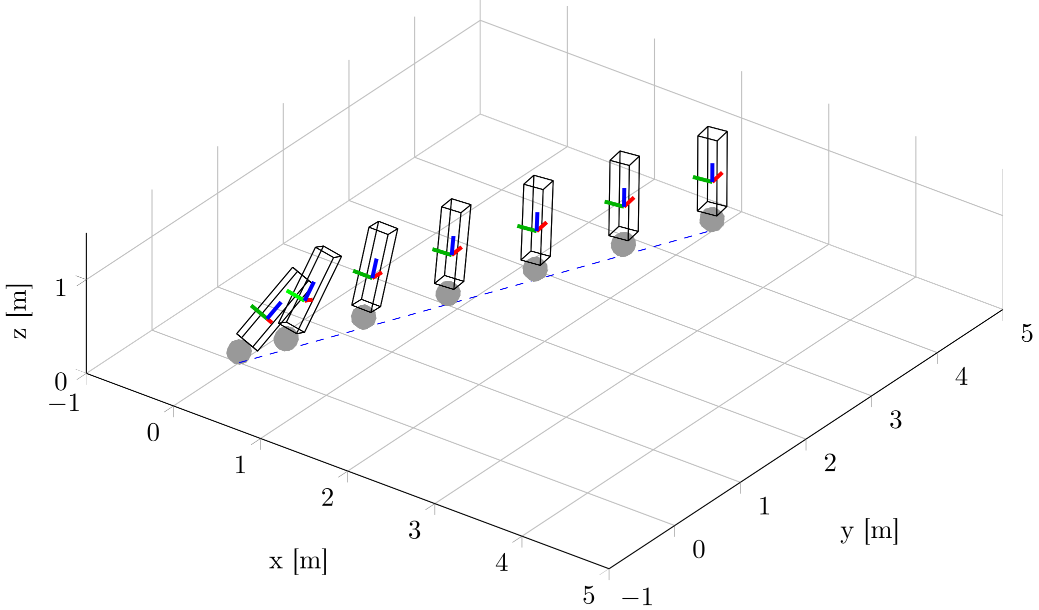


(d) Sliding surface **(e)** Lyapunov function of the sliding surface

Figure 5-5: Attitude tracking of the geometric sliding mode controller on S^2



(a) Tracking on $SO(3)$



(b) Tracking on S^2

Figure 5-6: Motion of the robot under different types of computed torque attitude controllers

5-5-2 Parametric uncertainty

In this subsection the influence of parametric uncertainties on the performance of the two attitude controllers is shown. The same initial conditions and controller parameters as in the previous section will be used. Recall that using these controller parameters a similar performance for both computed torque and sliding mode controller was obtained.

In Table 5-1 the nominal parameters as used in the controller and alternative parameters set are shown. The nominal parameters are identified as described in Section 7-1 and are taken from Table 7-1.

Table 5-1: Model parameters

Parameter	Nominal	Alternative	Unit
m_s	3.07	5	kg
m_b	12.93	20	kg
j_s	0.0255	0.5	$\mathrm{kg} \cdot \mathrm{m}^2$
$j_{\mathrm{b},x}$	1.400	3	$\mathrm{kg} \cdot \mathrm{m}^2$
$j_{\mathrm{b},y}$	1.406	3	$\mathrm{kg} \cdot \mathrm{m}^2$
$j_{\mathrm{b},z}$	0.183	1	$\mathrm{kg} \cdot \mathrm{m}^2$
p_b^r	$\begin{bmatrix} 0 & 0 & 0.4 \end{bmatrix}^T$	$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$	m

Computed torque Let us first consider the computed torque controller with tracking on $SO(3)$. In Figure 5-7 the trajectories of $b_{3,\mathrm{z}}$ are shown. This parameter represents the last element of the attitude vector b_3. As stated before, when $b_{3,\mathrm{z}}$ is below zero, the body robot of the robot surpasses a horizontal position and is very likely to collide with the ground. Furthermore, if $b_{3,\mathrm{z}} < 0$, the attitude controller is not defined. Therefore the computed torque controller was designed such that $b_{3,\mathrm{z}} \geq 0$, $\forall t > 0$ could be assured, given that certain conditions where met and provided *perfect knowledge of the system*.

It can be seen from Figure 5-7 that when the actual system parameters are the alternative system parameters given in Table 5-1, the controller synthesised using the nominal parameters does not stabilise the system and even results in a violation of the bound $b_{3,\mathrm{z}} \geq 0$. This result is to be expected. The computed torque controller assumes perfect knowledge of the system, and uses this knowledge to cancel out the dynamics. When the actual system is significantly different, stability cannot be assured. However, the region of stability can be increased by increasing the values of k_p and K_d. For tracking on S_2 similar results are obtained. The computed torque controller could be made robust to (initial) parameter uncertainties, by augmenting the control scheme with an adaptive control law to estimate the uncertain parameters.

Sliding mode It has been shown that the computed torque controller can result in unstable closed loop behaviour if the model in the controller significantly deviates from the actual system. Now let us consider the sliding mode controller. The convergence

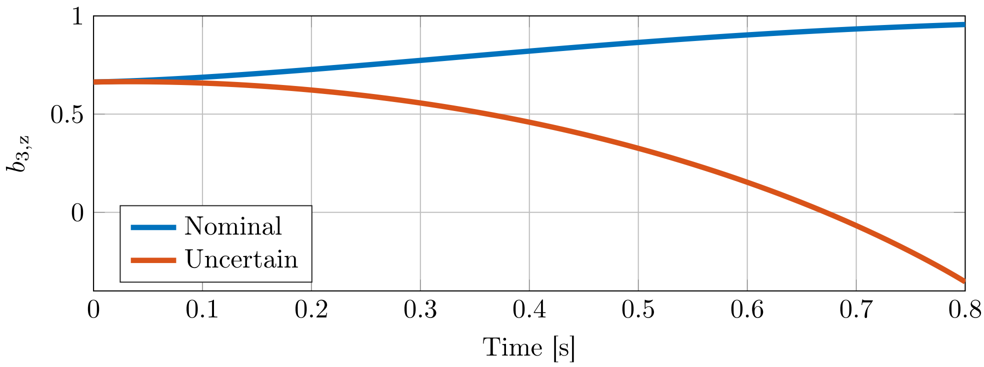


Figure 5-7: The trajectory of $b_{3,z}$ using the computed torque controller on $SO(3)$. In the nominal case the system parameters match the controller parameters. In the uncertain case the system has deviating system parameters, see Table 5-1.

of the sliding surface and the states will be indicated by the Lyapunov functions V_2 for the surface and V_1 for the states, given by $V_2 = \frac{1}{2}S^T S$ with S defined by (5-31) or (5-46) and V_1 by (5-35) or (5-49), for tracking on $SO(3)$ or S^2 respectively. Figure 5-8 and 5-9 show the performance of the sliding mode controller subjected to parametric uncertainties for $SO(3)$ and S^2 respectively.

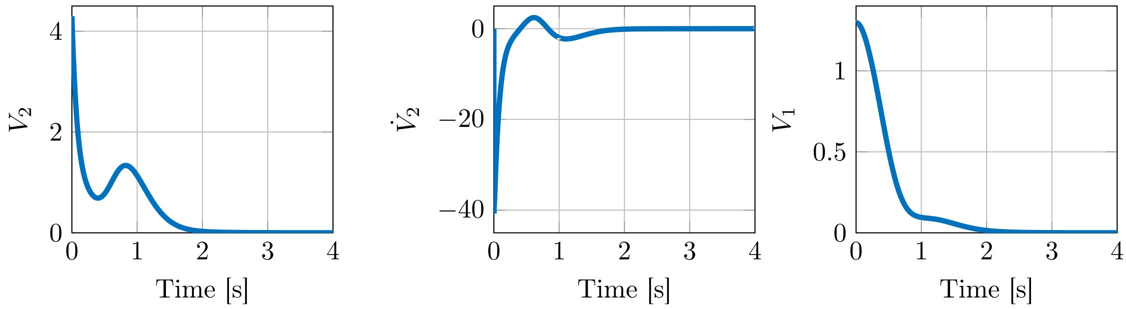
It can be seen from Figure 5-8a and 5-8b that even though the Lyapunov functions eventually converge to zero, the formal proof for stability, i.e. $\dot{V}_2 < 0 \ \forall S \neq 0$, is violated. $\dot{V}_2 > 0$ implies that the maximum of the disturbances D in (5-62) is larger than the control parameter η. Indeed, by increasing η, it is possible to obtain $\dot{V}_2 < 0$ for all t.

Remarkably, in Figure 5-9 it is shown that the attitude controller on S^2 does achieve $\dot{V} < 0$ for all $S \neq 0$ with the same controller parameters as stated before. This is especially remarkable since the condition for which $\dot{V}_2 < 0$ is satisfied when the system is subjected to disturbances, i.e. $\eta > D$, is the same for both $SO(3)$ and S^2. This can be explained that the condition $\eta > D$ results in an upper bound on $\dot{V}_2$ that is negative definite. However, this does not entail that smaller values of η cannot result in the negative definiteness of $\dot{V}_2$.

The difference between the negative definiteness of $\dot{V}_2$ for $SO(3)$ and S^2 can be explained by the fact that the influence of the parametric uncertainties on the sliding surface is larger for $SO(3)$ than for S^2. This can be expected, as tracking $SO(3)$ implies tracking in multiple directions simultaneously, compared to tracking in a single direction for S^2. As a result, η should be larger for $SO(3)$ to reject all parametric uncertainties than for S^2.

5-5-3 Unmodelled friction

In this section the effect of unmodelled friction on the controller performance is evaluated. The friction dynamics will not be modelled in the controller synthesis. The motivation behind this is that the estimation of the friction is relatively challenging and will also vary for different floor types and henceforth will be challenging to cancel out perfectly. It is therefore decided to have no friction model present in $f_2(\dot{q}_r, \bar{q})$.

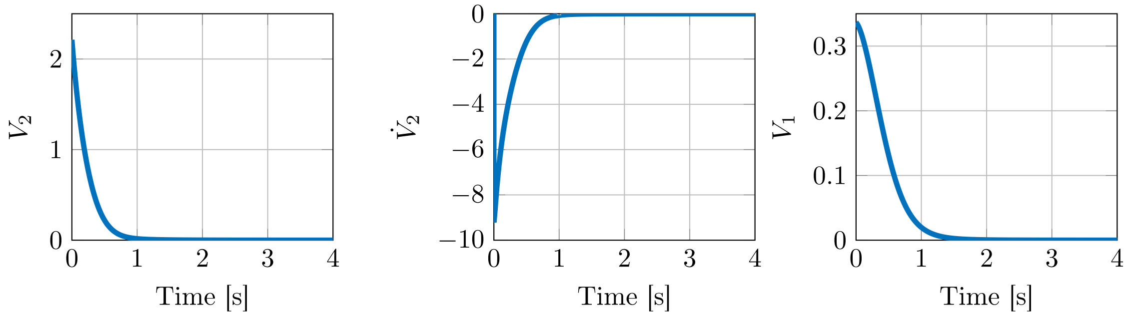


(a) Lyapunov function of the sliding surfaces

(b) Derivative of the Lyapunov function V_2

(c) Lyapunov function of the states

Figure 5-8: Sliding mode attitude control on $SO(3)$ with parametric uncertainties



(a) Lyapunov function of the sliding surfaces

(b) Derivative of the Lyapunov function V_2

(c) Lyapunov function of the states

Figure 5-9: Sliding mode attitude control on S^2 with parametric uncertainties

For this simulation, the friction model will be given by equations (4-52), (4-53) and (4-54). To speed up the simulation the sign function is replaced by a saturation function, similar to the use of the saturation function in the sliding mode controller. The friction parameters for this simulation are chosen to be $D_{c,1} = \operatorname{diag}(0.1, 0.1, 0.1)$, $D_{c,2} = \operatorname{diag}(0.1, 0.1, 0.1)$, $D_{v,1} = \operatorname{diag}(0.2, 0.2, 0.2)$ and $D_{v,2} = \operatorname{diag}(1, 1, 2)$.

For the design of the geometric sliding mode controller it was argued that by setting η higher than the maximum bound disturbance d_2^* the disturbance could be rejected. Since increasing η will have a relative small influence on the convergence of error functions, it will be considered a fair comparison to use the same controller parameters for the computed torque controller and sliding mode controller as in Section 5-5-1, but with $\eta = 50$. The initial conditions will again be the same as in Section 5-5-1.

The result of the unknown friction on the performance of the S^2 attitude controllers is shown in Figure 5-10 is for the computed torque controller and in Figure 5-11 for the sliding mode controller. As could be expected, it is observed that the computed torque controller is not able to stabilise the attitude of the system. This was already predicted in Section 5-3-2 when evaluating the influences of uncertainties. However, it is observed that the attitude of the attitude becomes steady-state. The performance could be improved by introducing an integrator to the error dynamics to compensate for the steady state error of the attitude. Alternatively, a mismatch in friction parameter could be diminished by applying adaptive control.

For the sliding mode controller on the other hand, it can be observed that the controller is able to let the error functions converge to zero in finite time. Moreover, the performance of the controller subjected to unmodelled friction compared to the nominal case in Figure 5-5 is almost identical. Note that convergence of the sliding surfaces is much faster in this case of unmodelled friction, as η is chosen much higher. Due to the high gain of the switching control, the system is able to converge in finite time to the sliding surface and the effect of the unmodelled friction is diminished.

5-5-4 Position control

In this subsection the position control is briefly shown. It is chosen to only show simulation results where unknown friction is present, in order to simulate the most realistic case. Since the friction is unknown, only simulation results for the sliding mode controller will be presented, since this controller is able to compensate for unknown dynamics. It should be noted a very similar result will be obtained if a computed torque controller with known friction is used and hence this scenario is omitted.

The position controller assumes $\omega_{\mathrm{b}}^{0,0} \approx 0$, $\dot{\omega}_{\mathrm{b}}^{0,0} \approx 0$ and friction is neglected, i.e. the position controller assumes slow evolving dynamics. As a result, the control parameters C_p and C_d are chosen relatively low. However, friction introduces an additional unknown dissipation in the system, such that it is advised to choose a higher C_p than in the nominal case without friction.

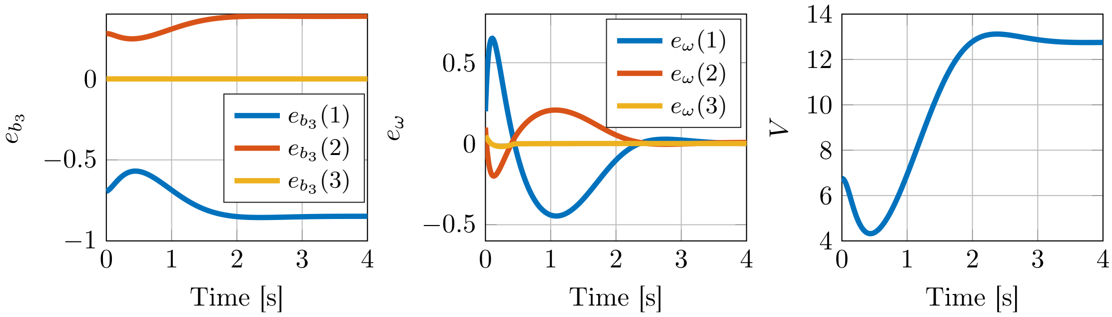
Station keeping Let us first considering station keeping, i.e. the robot should maintain at a fixed position. In this simulation the robot is started with a certain initial position and the robot should be able to control the system back to its starting position and remain there.

The position controller parameters are chosen to be $C_p = 4$ and $C_d = 2$. The attitude controller parameters will again be given by $\alpha = 3$, $k = 30$, $\eta = 50$ and $\Delta = 0.05$. Again the same initial conditions as before are used. Figure 5-12 and 5-13 show the performance of the position controller in combination with the sliding mode attitude controller for $SO(3)$ and S^2 respectively.

Figures 5-12a and 5-13a show the desired attitude vector b_{d} as returned by the position controller. It can be observed that the actual attitude vector b_3 lags behind the desired attitude b_{d}. To increase time of convergence of the attitude, it would make sense to increase the gains of the attitude controller. However, this would have a negative effect on the performance, as increasing α results in $\dot{\omega}_{\mathrm{b}}^{0,0} \gg 0$, $\omega_{\mathrm{b}}^{0,0} \gg 0$ and hence the assumptions under which the position controller returns a desired attitude are violated.

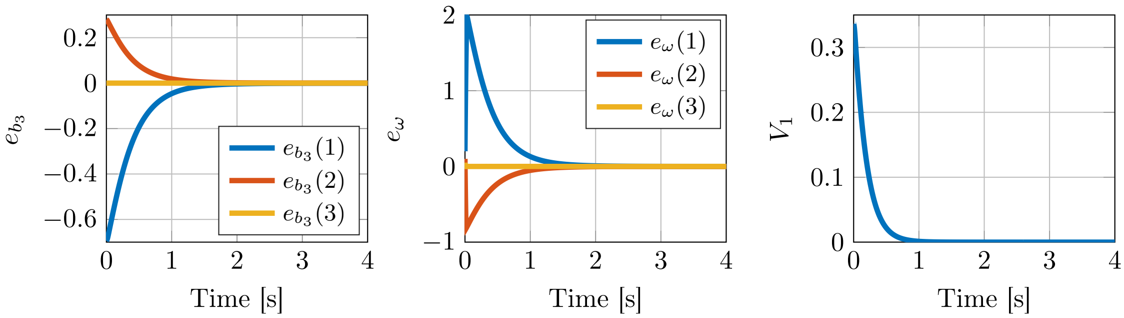
In Figures 5-12b and 5-13b it can be seen that indeed the position of the robot returns to zero after the initial disturbance.

Figure 5-14 shows a clear difference between station keeping using either an $SO(3)$ or S^2 attitude controller. Due to the desired heading attitude for tracking on $SO(3)$, also a control action around the z axis of the body frame is required. As a result, the xy trajectory of the robot becomes curved. For the tracking on S^2 on the other hand,

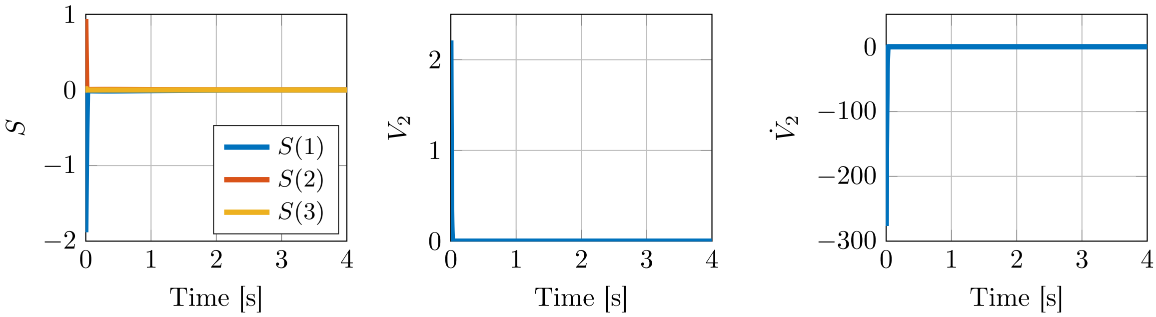


(a) Attitude error **(b)** Angular velocity error **(c)** Lyapunov function

Figure 5-10: Attitude tracking with unknown friction of the geometric computed torque controller on S^2



(a) Attitude error **(b)** Angular velocity error **(c)** Lyapunov function



(d) Sliding surface **(e)** Lyapunov function of the sliding surface **(f)** Derivative of the Lyapunov function V_2

Figure 5-11: Attitude tracking with unknown friction of the geometric sliding mode controller on S^2

the heading direction is free. Hence the system is only controlled around the axis of minimal rotation and as a result the xy trajectory is smaller and forms an (almost) straight line. Note that the small curve results from the initial angular velocity of the body.

Point-to-point control

In this subsection the point to point control of the robot is briefly shown. Two examples will be given.

First of all a relative small distance is considered $p^0_{\mathrm{k,d}} = \begin{bmatrix} x_d & y_d \end{bmatrix}^T = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$. The result is shown in Figure 5-15. From Figure 5-15b it can be observed that ball first moves backwards before it moves into the desired direction. This behaviour is due to the zero dynamics of the system. Furthermore, it can be observed that the system indeed converges to the desired position.

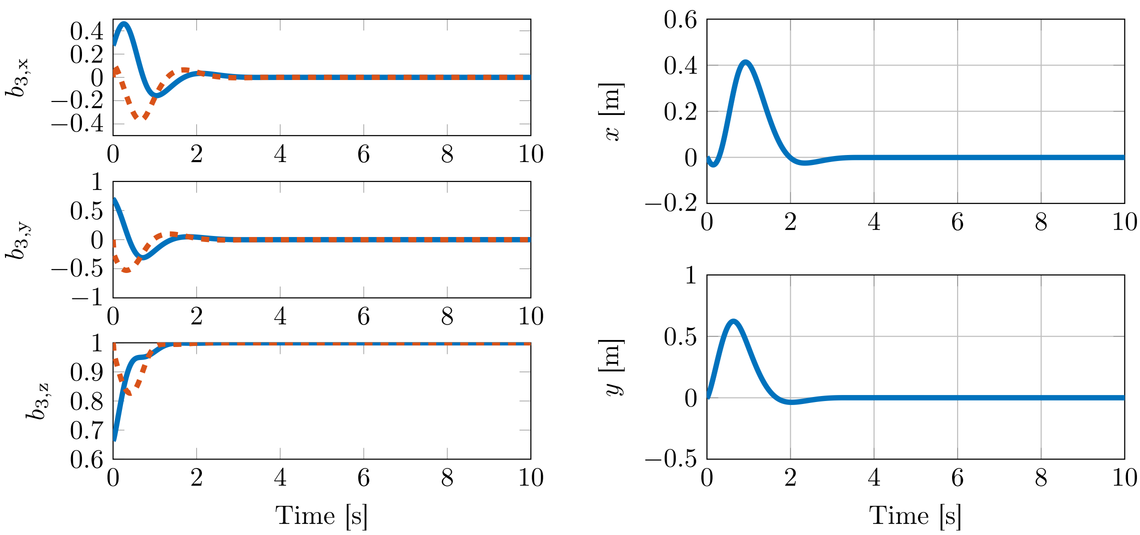
Now let us consider a larger distance, namely the setpoint $x_d = 15\ m$, $y_d = 15\ m$. The result is shown in Figure 5-16. Again it is observed that the controller is able to control the robot to the desired position. However, it can be observed that the actual attitude slightly overshoots the desired attitude vector b_{d}. This could result in problems, as the desired attitude is saturated to have a certain maximum allowed tilt angle. It especially becomes problematic if the attitude $b_{3,\mathrm{z}}$ drops below zero, as the both the position and attitude controller are not defined for $b_{3,\mathrm{z}} < 0$. The overshoot in attitude is a result from the boundary layer of the saturation function. Due to the unmodelled friction and boundary layer, the system is not able to converge to $S = 0$ in finite time and will be within the interval $0 \leq S < \Delta$. Indeed, the overshoot can be reduced by reducing the boundary layer.

5-5-5 Disturbance rejection

In this subsection the disturbance rejection of the overall control algorithm is shown. This includes the attitude controller and position controller. The disturbances applied to the system will be a 'push', i.e. a torque will be applied for a short period of time, and a persistent torque pushing against the body of the robot. Furthermore, the friction acting on the system will be regarded as unknown and will not be present in f_2. For the simulation the sliding mode controller on S^2 will be used. Note that similar results will be obtained when using the computed torque attitude controller with known friction or for tracking on $SO(3)$.

For the simulation, the same controller parameters for the sliding mode controller on S^2 with unknown friction will be used. The setpoint for the position controller is set at $x_d = 4$, $y_d = 0$. The push will be simulated by applying a torque of $\tau_d = \begin{bmatrix} -20 & 0 & 0 \end{bmatrix}$ Nm to the body of the robot at $t = 1$ for 0.2 seconds. The resulting simulation result is shown in Figure 5-17. As a result of the push, the robot deviates from a straight line to the objective. However, the system is still able to converge to the desired position.

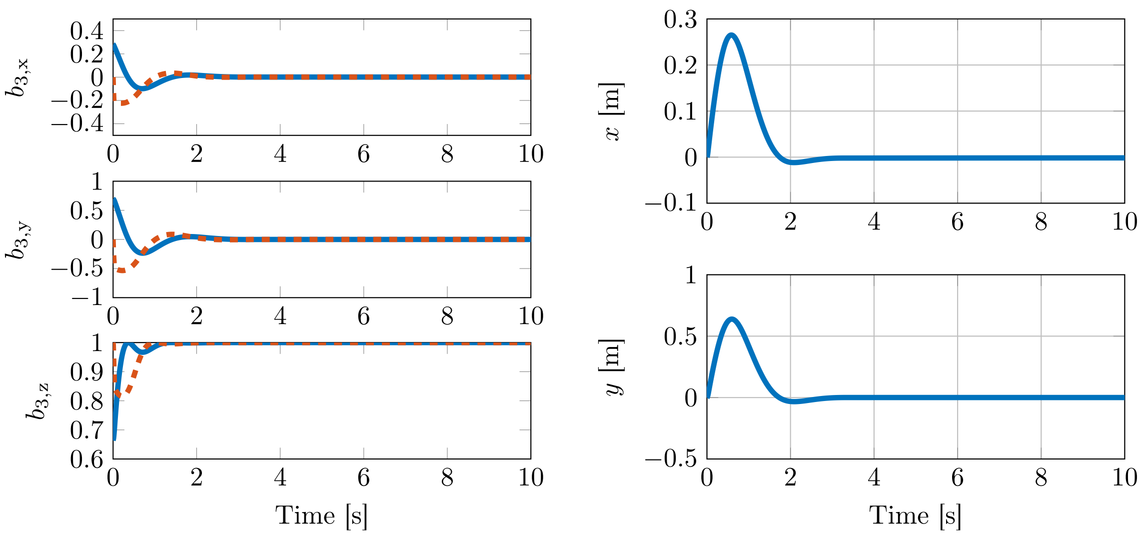
In Figure 5-18 a persistent torque is applied to the system. While the attitude controller is able to reject disturbances up to a certain magnitude, the position controller is not able to reject persistent disturbances. As a result, the position shows a steady-



(a) Attitude vector. (Dashed: desired attitude b_d, solid: actual attitude)

(b) xy displacement

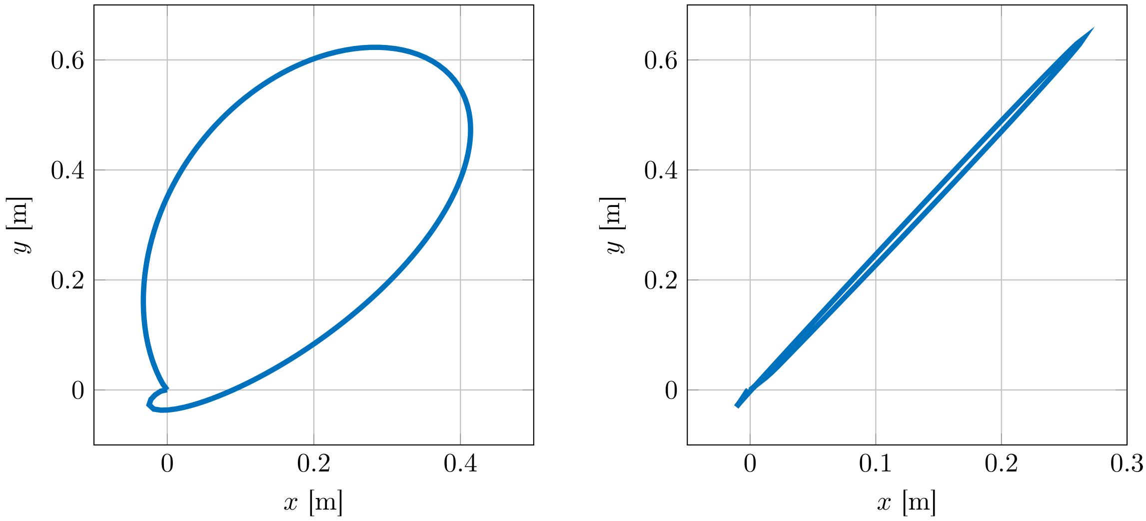
Figure 5-12: Station keeping using the sliding mode controller on $SO(3)$ with unknown friction



(a) Attitude vector. (Dashed: desired attitude b_d, solid: actual attitude)

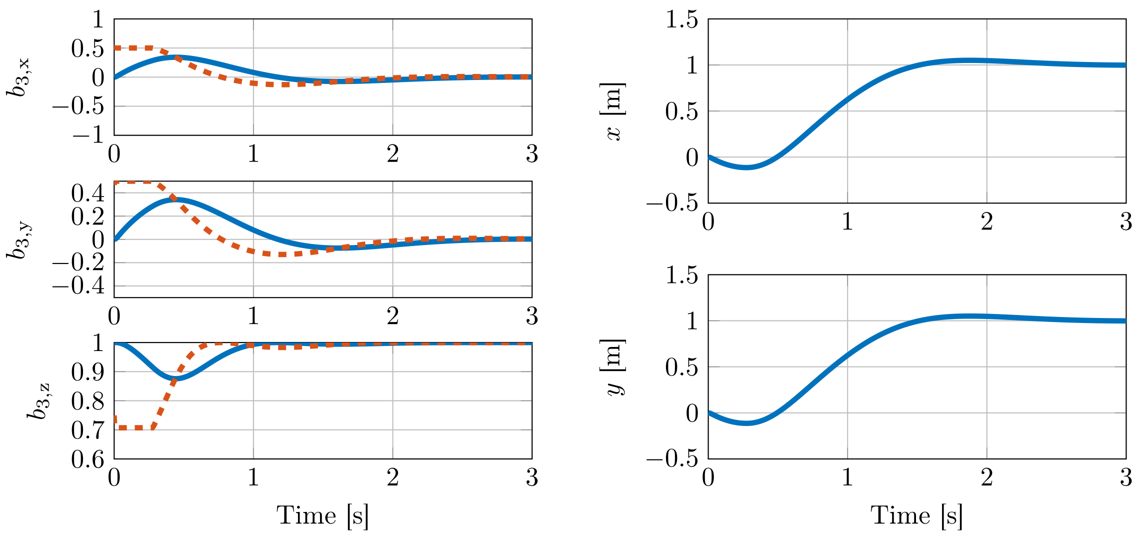
(b) xy displacement

Figure 5-13: Station keeping using the sliding mode controller on S^2 with unknown friction



(a) $SO(3)$ **(b)** S^2

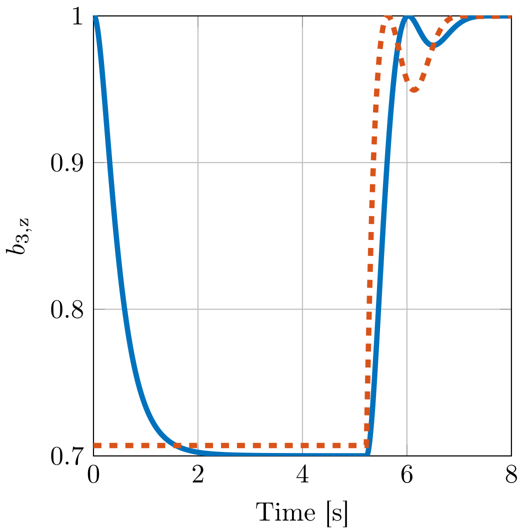
Figure 5-14: xy trajectory of the ball



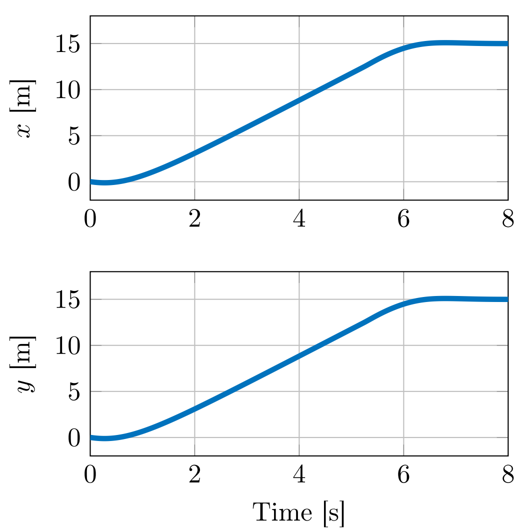
(a) Attitude vector. (Dashed: desired attitude b_d, solid: actual attitude)

(b) x and y displacement of the ball

Figure 5-15: Point-to-point control using the sliding mode controller on S^2 with the desired position $x_d = 1\ m$, $y_d = 1\ m$



(a) z component of the attitude vector. (Dashed: desired attitude b_d, solid: actual attitude)



(b) x and y displacement of the ball

Figure 5-16: Point-to-point control using the sliding mode controller on S^2 with the desired position $x_d = 15\ m$, $y_d = 15\ m$

state error in Figure 5-18b. When the disturbance were to be removed, the system would be able to converge to the desired position. The steady-state error in position could be removed by introducing an integrator in the error dynamics of the position controller. However, one might should consider if it is desired for their robot to be able to 'enforce' position tracking when a constant torque is applied to the robot. For example, if the robot is used as a demonstrator at an event, it is not desired if the robot tries to enforce a certain position, even if there is a person in its path.

5-5-6 Trajectory tracking

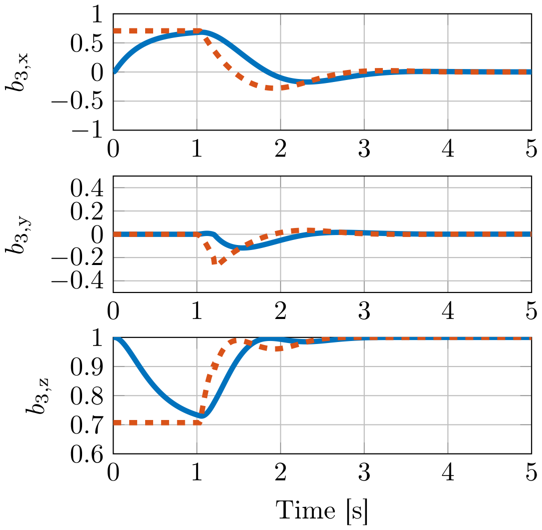
In this subsection the trajectory tracking of a circle and triangle is shown. The triangular trajectory is chosen because of its steep change in direction at the corners, which shows the omni-directionality of the robot. Since both attitude controllers show very similar performance when no uncertainties are present, it is chosen to only show the trajectory tracking using the geometric sliding mode attitude controller.

The maximum desired angular velocity of the robot is set to be 54.5510 rad/s^2, which corresponds to a maximum tilt angle of $\frac{\pi}{4}$ radians with respect to the z axis.

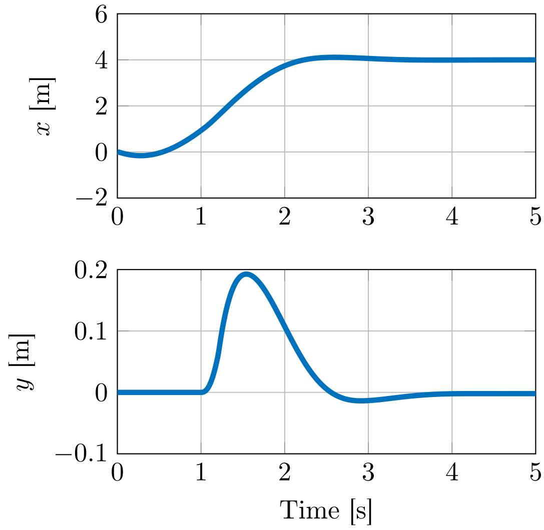
Circular trajectory First of all the tracking of a circle is shown. The desired trajectory in the xy plane is given by

$$p^0_{\mathrm{k,traj}} = \begin{bmatrix} a\sin(ct) \\ a\cos(ct) \end{bmatrix} \qquad v^{0,0}_{\mathrm{k,traj}} = \begin{bmatrix} a\ c\cos(ct) \\ -a\ c\sin(ct) \end{bmatrix} \qquad \dot{v}^{0,0}_{\mathrm{k,traj}} = \begin{bmatrix} -a\ c^2\sin(ct) \\ -a\ c^2\cos(ct) \end{bmatrix} \tag{5-78}$$

where a denotes the radius of the circle and c the angular frequency. For this simulation the parameters of the trajectory are chosen to be $a = 2$, $c = \dfrac{1}{5}2\pi$, hence one period is

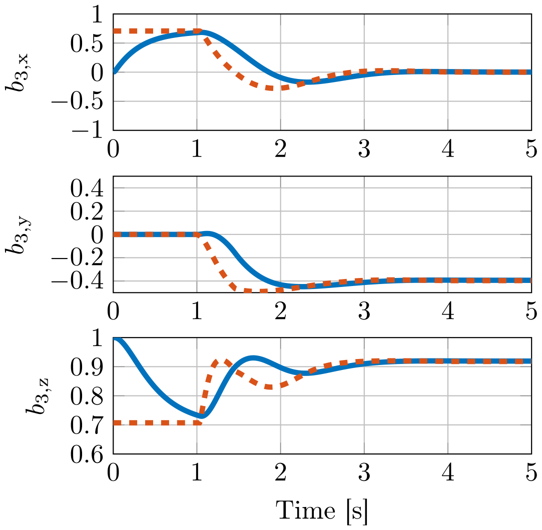


(a) Attitude vector. (Dashed: desired attitude b_d, solid: actual attitude)

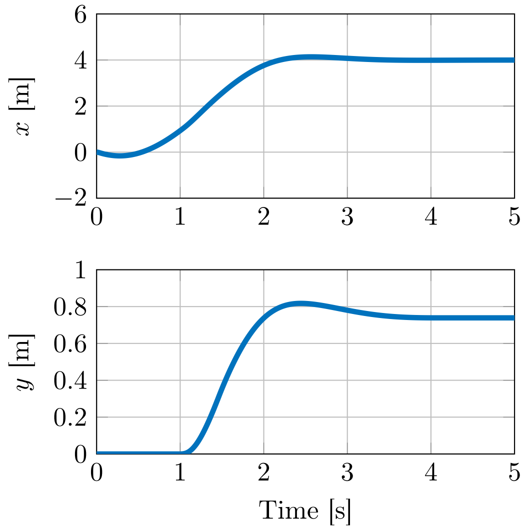


(b) xy displacement

Figure 5-17: Point-to-point movement with a block disturbance of -20 Nm applied to the body at $t = 1$, $\Delta t = 0.2$



(a) Attitude vector. (Dashed: desired attitude b_d, solid: actual attitude)



(b) xy displacement

Figure 5-18: Point-to-point movement with a persistent disturbance of -20 Nm applied to the body at $t = 1$

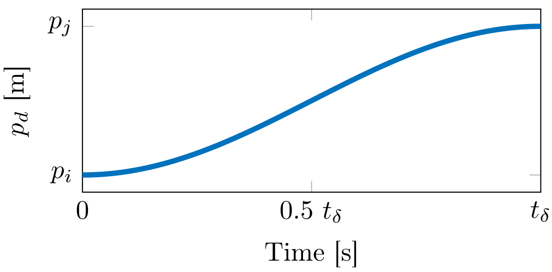


Figure 5-19: Sinusoidal trajectory between two vertices of the desired trajectory

5 seconds. Furthermore, the parameters of the position controller are set to be $C_p = 4$ and $C_d = 2$. The geometric sliding mode controller parameters are chosen to be $\alpha = 3$, $k = 30$, $\eta = 50$, $\Delta = 0.05$.

In Figure 5-20 the simulation results are shown. As can be expected, the actual attitude lags behind the true attitude. Furthermore, given these control parameters it can be observed that the actual xy trajectory has a smaller radius than the desired radius. This is caused due the fact that the desired trajectory requires faster dynamics than that results from the controller. The tracking error could be reduced by increasing the position controller gain. However, faster position controller gains increases the acceleration of the system for which can results in worse performance, due to the assumption of slow system dynamics. Alternatively the tracking can be improved by introducing an integral term in the position controller control law. Finally, another method to improve the tracking is iterative learning control. Here the control law is updated each cycle using the tracking error of that cycle in order to achieve a better tracking performance in the next cycle.

Triangular trajectory Let us consider a triangular trajectory. There are two possibilities to track a triangle, i.e. subsequent point-to-point control or defining a time dependent predefined trajectory. Since the performance of the point-to-point controller already has been shown, it is chosen to show the latter.

It is chosen to control the robot between to points with a sinusoidal trajectory, given by the function:

$$p_d(t) = \frac{1}{2}\Big(-(p_j - p_i)\cos\Big(\frac{\pi}{t_\delta}(t - t_\delta)\Big) + (p_j - p_i)\Big) + p_i$$

Here p_i and p_j denotes the first and second vertex respectively and t_δ the time between the two vertices. This sinusoid is illustrated in Figure 5-19. The desired linear velocity and acceleration is found by taking the first and second derivative of p_d.

In Figure 5-21 the simulations are shown for triangular trajectory tracking for a trajectory with the vertices at

$$p_1 = \begin{bmatrix} 3 & 1.5 \end{bmatrix}^T, \qquad p_2 = \begin{bmatrix} 0 & 3 \end{bmatrix}^T, \qquad p_3 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$$

and $t_\delta = 3$ seconds. The controller parameters are again $C_p = 4$ and $C_d = 2$, $\alpha = 3$, $k = 30$, $\eta = 50$, $\Delta = 0.05$.

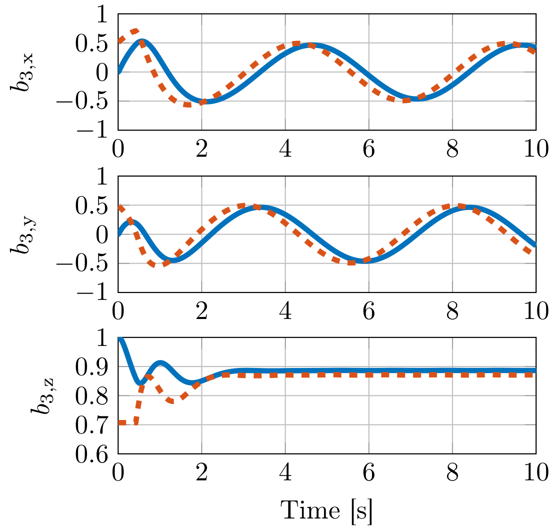
As could be expected, it can be observed that actual trajectory lags behind the desired trajectory. As a result the robot changes direction slightly before the actual vertex is reached. Again, the performance of this cyclic trajectory could be improved using iterative learning methods that use the errors per cycle to improve the tracking performance in the next.

In this section the main results from the geometric controller will be summarised. In Chapter 4 the dynamics of the system was modelled. The robot consists of two bodies. Hence the workspace of the system is given by $SE(3) \times SE(3)$. It was shown that the reduced configuration space is given by $SO(3) \times SO(3) \times \mathbb{R}^2$ and the configuration space of the dynamics $SO(3) \times SO(3)$, i.e. the system dynamics can be fully described by the rotations of the body and ball of the robot. However, when controlling the robot, the goal space of the robot is not equal to the workspace and configuration space. Two types of attitude control where shown, one for $SO(3)$ and one for S^2. Furthermore, it was desired to control the position of the robot. The goal space is hence given by $SO(3) \times \mathbb{R}^2$ or $S^2 \times \mathbb{R}^2$. In the latter the heading direction of the robot is free. This makes sense for the ballbot, since it is an omnidirectional robot.

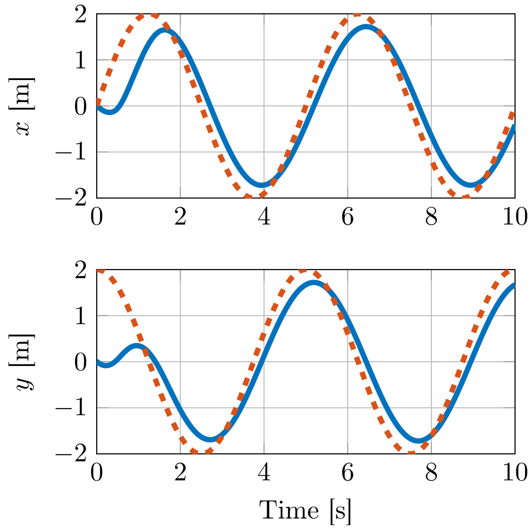
Tracking on $SO(3)$ and S^2 First of all, two type of attitude tracking were defined; one on the special orthogonal group $SO(3)$ and one on the 2-sphere S^2. The former tracks the attitude with a specific heading direction, while the latter only tracks the attitude a free heading direction of the body. For both types of tracking special error functions where defined and conditions on the initial conditions where defined to assure stability. For the tracking on $SO(3)$ the bounds on the initial conditions also implied conditions to the initial maximum allowed yaw rotation and as a result not all viable initial conditions (i.e. all attitudes for which $b_{3,\mathrm{z}}(0) \geq 0$) were covered in the region of attraction. For the tracking on S^2 on the other hand the bounds on the initial attitude where already satisfied, as they where all outside the viable region $b_{3,\mathrm{z}}(0) \geq 0$. Finally, the benefit of S^2 over $SO(3)$ is that it provides the minimal rotation to recover from an initial disturbance to the upright position.

Computed torque and sliding mode control Two types of nonlinear attitude control were designed to track the attitude on either $SO(3)$ or S^2. Both controllers are able to stabilize the attitude when there is perfect knowledge of the system. However, the required bounds on the initial conditions for the computed torque controller are more strict than the bounds for the sliding mode controller, as the computed torque controllers also require a bound for the angular velocity. However, both controllers still have to satisfy that all system trajectories remain satisfy $b_{3,\mathrm{z}}(t) \geq 0$ for all t, which indirectly also poses a bound on the initial condition of the angular velocities for the sliding mode controller.

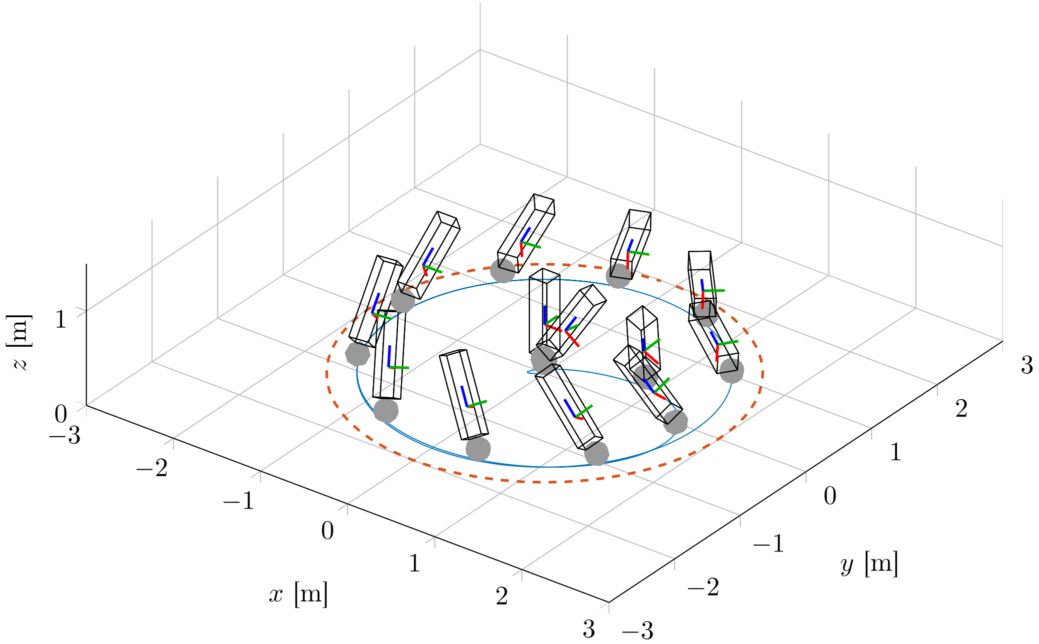
When the system is subjected to parametric uncertainty and/or unmodelled dynamics, such as unmodelled friction, it was shown that the sliding mode controller was more robust than the computed torque controller. Furthermore, it was shown that the sliding attitude controller on S^2 was more robust than the attitude controller on $SO(3)$. Both controller were capable of assuring stability, provided that $\eta > D$. It was argued that the stability of the computed torque controller could be improved by applying



(a) Attitude vector. (Dashed: desired attitude b_{d}, solid: actual attitude)

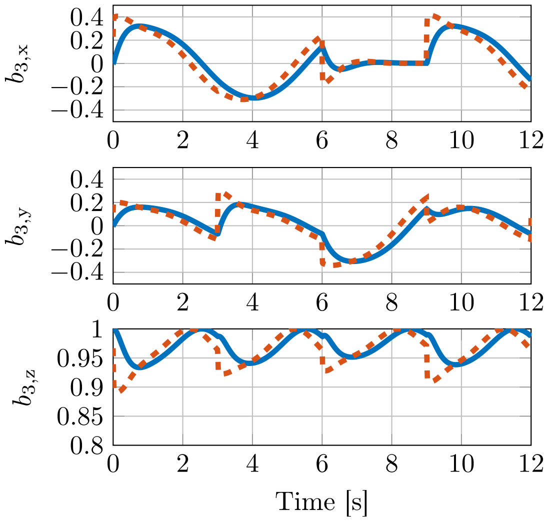


(b) Trajectory of the ball in the xy plane. (Dashed: desired attitude b_{d}, solid: actual attitude)

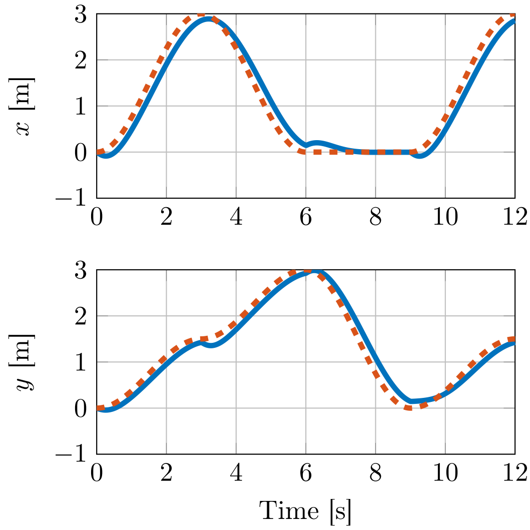


(c) 3D movement

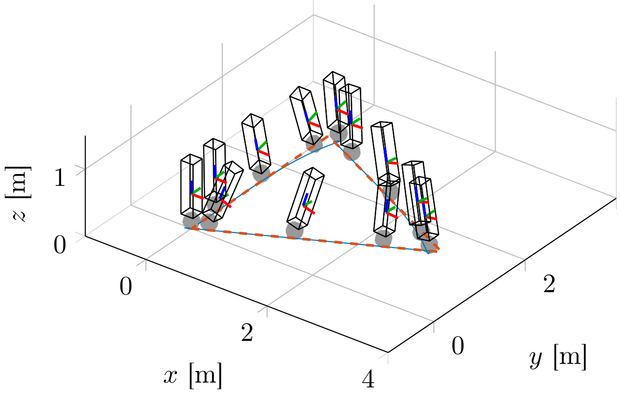
Figure 5-20: Trajectory tracking of a circle



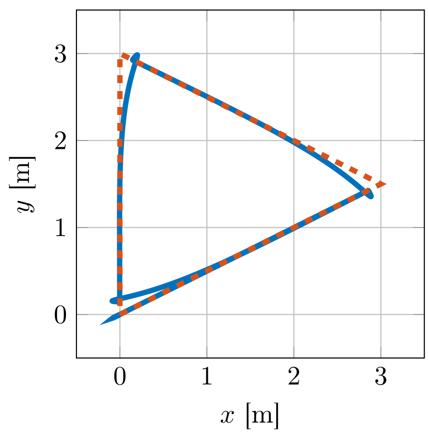
(a) Attitude vector. (Dashed: desired attitude b_{d}, solid: actual attitude)



(b) Trajectory of the ball in the xy plane. (Dashed: desired trajectory, solid: actual trajectory)



(c) 3D movement



(d) xy movement. (Dashed: desired trajectory, solid: actual trajectory)

Figure 5-21: Trajectory tracking of a triangle adaptive control to estimate the uncertain parameters.

The downside of the sliding mode controller is that the discontinuous switching control can cause chattering. Furthermore, high frequency switching can cause fatigue of the system, resonate with unmodelled dynamics and high power consumption. In order to reduce the chattering, a saturation function was introduced. In this saturation function, a small boundary around the sliding surface is created in which the switching law is replaced by an asymptotic control signal. It was shown that if this boundary layer is too large, a small and slow converging error to the sliding surface will persist.

In Table 5-2 the computed torque and sliding mode controller are compared.

Table 5-2: Controller comparison

	Computed torque $SO(3)$	S^2	Sliding mode $SO(3)$	S^2
Bound initial conditions	$\phi(R_e) < 2$, $e_\omega^T e_\omega < k_p\,(2 - \phi(R_e))$	$\phi(b_3, b_\mathrm{d}) < 2$, $e_\omega^T e_\omega < k_p\,(2 - \phi(b_3, b_\mathrm{d}))$	$\phi(R_e) < 2$	$\phi(b_3, b_\mathrm{d}) < 2$
Heading direction	Yes	No	Yes	No
Robustness uncertain parameters	No	No	Yes [1]	Yes [1]

Position controller The position control for the robot was done through a simplified adaptation of the shape-space planner of [11] to a geometric framework. This controller finds a desired attitude, given a desired acceleration in $\mathbb{R}^2$. The advantage of this method is its simplicity and it is possible to compute on-line, in contrast to trajectory optimization algorithms such as direct collocation methods.

It was shown that the overall control structure was able to stabilize the system at a certain position. Since the position controller returns an approximated desired attitude, no optimality is achieved. Furthermore, it was assumed that the desired angle can be perfectly tracked, which is in reality is often not true.

It was found that the overall combination of the position and attitude controller was able to reject 'pushes' to the system. However, it was found that persistent disturbances could not be rejected, i.e. the system would converge to a steady state error with respect to the position. However, it was argued that for the use case of the robot this is desirable.

The combination of attitude controller parameters and position controller parameters should be chosen such that $\dot{\omega}_\mathrm{b}^{0,0}$ and $\omega_\mathrm{b}^{0,0}$ remain relative small, as the position controller assumes $\dot{\omega}_\mathrm{b}^{0,0} = \omega_\mathrm{b}^{0,0} = 0$. Choosing one or both controllers to aggressive results in unstable behaviour. No formal prove of the overall stability is given.

While domain of the attitude controller consists of the upper halve of a sphere, it is still not desired to take advantage of this domain for the position tracking controller.

[1] Provided that $\eta > D$

(a) Joe [26]

(b) Segway RMP [27]

Figure 5-22: Examples of two-wheeled inverted pendulum robots

As mentioned before, the position controller assumes slow evolving dynamics. Using steep angles for the position tracking controller can result in large overshoots in the position and hence worse tracking of the desired trajectory. Furthermore, if the attitude is overshot, the system might violate the required bound $b_{3,\mathrm{z}} \geq 0$ which might result in a singularity in the control law.

It was found that for some scenario's the position controller was subjected to steady-state errors. These steady-state error could be compensated by introducing an integrator in the error dynamics of the position controller. The position controller could also be improved by taking the dynamics into account, i.e. $\dot{\omega}_{\mathrm{b}}^{0,0} \neq 0$, $\omega_{\mathrm{b}}^{0,0} \neq 0$. Finally, it was argued that cyclic trajectory tracking could be improved by applying iterative learning control.

5-5-7 Extension to other systems

In this chapter a geometric controller was proposed for a ball-balancing robot. The goal space of such a system is given by a translation on $\mathbb{R}^2$ and a rotation on either $SO(3)$ or S^2. Due to the under-actuated nature of the system,i.e. either the translational space $\mathbb{R}^2$ or the rotational space $SO(3)$ can be considered fully actuated, but not the overall system. In this section similar systems as the ballbot are explored and a brief discussion is given on how the proposed geometric controller could be extended to these systems.

First of all, let's consider fully actuated system on S^2 or $SO(3)$. For these systems the attitude controllers can be directly extended. Examples of such systems the inverted 2D/3D pendulum, and a satellite.

Examples of under-actuated systems with a goal space on $\mathbb{R}^n \times SO(3)$ or $\mathbb{R}^n \times S^2$ are quadrotors, two-wheeled inverted pendulum systems, the pole-on-cart system and the crane system.

A geometric approach for quadrotors has been proposed in [25]. However, here it is chosen to let the configuration coincide coincide with the goal space, by imposing a desired heading direction to the system. However, a quadrotor is an omni-directional system and it might be possible that no specific orientation is desired. In that case the tracking error on S^2 could for the control of the attitude, as demonstrated for the ball-balancing robot.

Let us consider the two-wheeled inverted pendulum, such as for example Joe or the Segway [26, 27], see Figure 5-22. The workspace of such a system is $SE(3) \times SE(3) \times SE(3)$, i.e a rotation and translation for the body and both wheels. The reduced configuration space could be represented by S^2 for the body and $S \times S$ for both wheels, thus the reduced configuration space is given by $S^2 \times S \times S \times \mathbb{R}^2$. The goal space of the system is given by $S^2 \times \mathbb{R}^2$. By considering the dynamics on S^2 as the fully actuated space, it is possible to use similar techniques as proposed in this work.

The 3D crane or 3D pendulum on cart are very similar to the ballbot. The workspace is given by $SE(3) \times SE(3)$ and the configuration space by $\mathbb{R}^2$ for the cart and S^2 for the payload/pendulum. The goal space is given by $S^2 \times \mathbb{R}^2$. By considering the dynamics on S^2 to be fully actuated, a similar control technique as proposed in this thesis can be used.

Chapter 6

Geometric observer

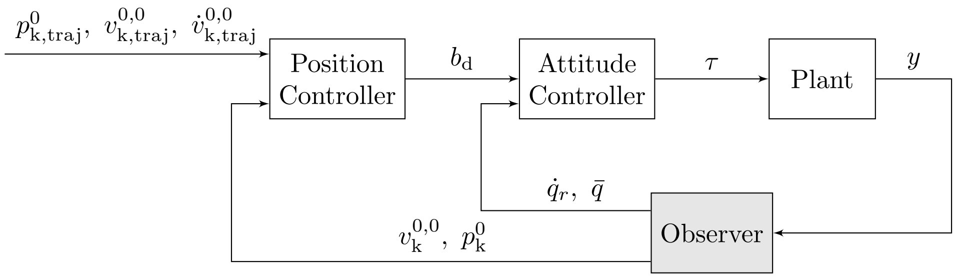


Figure 6-1: Controller structure

In this chapter a geometric observers will be proposed. Several geometric observers for the special orthogonal group $SO(3)$ that use the error functions e_R and e_ω have been proposed in literature. Examples are [28], in which an observer is proposed for the orthogonal group that can be compared to a complementary filter and [29] in which the observer is designed based on the error functions and a Lyapunov function. However, these approaches are limited to the kinematics of the system on $SO(3)$. In this chapter a similar approach is taken as in [29], i.e. using the error functions and a Lyapunov function. However, in contrary to the kinematic observers, also the dynamics will be considered using an approach similar to global observers [30]. The aim of this approach is to use the same error functions as defined for the geometric controller to design a 'simple' global geometric observer.

This chapter will begin by describing the available measurement data of the ball-balancing robot. Secondly, a global geometric observer for a second order system on $SO(3)$ will be designed. This observer is followed by a geometric observer for the ball-balancing robot, followed by simulation results. The chapter will be concluded by a discussion of the proposed observer.

6-1 Available measurement data

This section is devoted to describing the available measurement data from the ballbot available at Alten Mechatronics, and therefore is platform specific. In the end the relations between the measurements and states will be described in the form

$$y = h(x) \tag{6-1}$$

6-1-1 Body

In order to measure the states of the body, a STM32F3 Discovery board as sensor board is used. This board is equipped with an accelerometer, gyroscope and magnetometer.

Gyroscope The gyroscope measures angular velocity $\omega_{\mathrm{g}}^{\mathrm{g},0}$, where frame Ψ_{g} denotes the specific gyroscope frame. The gyroscope frame relates to the body frame by:

$$R_{\mathrm{g}}^{\mathrm{b}} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6-2}$$

Using this rotation matrix and using the fact that the gyroscope is rigidly connected to the body, i.e. $\omega_{\mathrm{g}}^{\mathrm{g},0} = \omega_{\mathrm{b}}^{\mathrm{g},0}$:

$$\omega_{\mathrm{b}}^{0,0} = R_{\mathrm{b}}^{0} R_{\mathrm{g}}^{\mathrm{b}} \omega_{\mathrm{g}}^{\mathrm{g},0}$$

However, in the implementation of the current robot, the transformation of 6-2 is already resolved in the sensor board. Therefore, this transformation will not be considered in the design of the observer, i.e. the measured state is considered to be $\omega_{\mathrm{b}}^{\mathrm{b},0}$:

$$\omega_{\mathrm{b}}^{0,0} = R_{\mathrm{b}}^{0} \omega_{\mathrm{b}}^{\mathrm{b},0}$$

or in the form $y = h(x)$:

$$\omega_{\mathrm{b}}^{\mathrm{b},0} = R_{0}^{\mathrm{b}} \omega_{\mathrm{b}}^{0,0} \tag{6-3}$$

Accelerometer Similar to the gyroscope, the accelerometer has also its own frame Ψ_{a}. The accelerometer relates to the body frame by:

$$R_{\mathrm{a}}^{\mathrm{b}} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6-4}$$

However, in the current implementation this is also resolved within the sensor board.

The accelerometer measures the specific forces $v_a \in \mathbb{R}^3$. It is assumed that these specific forces can be approximated by only the normal force. Therefore, the normalised accelerometer data would be:

$$\frac{v_a}{\|v_a\|} = R_0^{\mathrm{b}} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{6-5}$$

where denotes the accelerometer measurements.

Magnetometer The magnetometer measures the direction of the magnetic field of the earth $v_m \in \mathbb{R}^3$ expressed in body frame. Let us denote the direction of the magnetic field of the earth as m. The normalized magnetometer is given by:

$$\frac{v_m}{\|v_m\|} = R_0^{\mathrm{b}} m \tag{6-6}$$

6-1-2 Sphere

The states of the sphere are measured through the motor encoders, although the position of the sphere is unobservable [3]. The three motors are equipped with encoders, which measure the travelled angles ϕ_i of the omni-wheels $i \in \{1, 2, 3\}$.

Now let us derive a relation between the measured variables and the state variables. Consider the relation between input torques and the applied torque on the sphere, expressed in inertial frame, from Chapter 4 in equation (4-49):

$$\left(\tau^{0,\mathrm{s}}\right)^T = \frac{r_s}{r_w} R_{\mathrm{b}}^0 \begin{bmatrix} -\frac{1}{2}\sqrt{2} & \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{2} \\ 0 & -\frac{1}{4}\sqrt{6} & \frac{1}{4}\sqrt{6} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{6-7}$$

The total supplied torque to the system is independent of coordinate frame and parametrisation. Therefore the following relation for τ_i, $i \in \{1, 2, 3\}$ and $\tau^{0,\mathrm{s}}$ holds:

$$P = \tau\dot{\phi} = \tau^{0,\mathrm{s}} \omega_{\mathrm{s}}^{0,0} \tag{6-8}$$

Now, substituting equation (6-7) yields:

$$\tau\dot{\phi} = \tau \frac{r_s}{r_w} \begin{bmatrix} -\frac{1}{2}\sqrt{2} & 0 & \frac{1}{2}\sqrt{2} \\ \frac{1}{4}\sqrt{2} & -\frac{1}{4}\sqrt{6} & \frac{1}{2}\sqrt{2} \\ \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{6} & \frac{1}{2}\sqrt{2} \end{bmatrix} \left(R_{\mathrm{b}}^0\right)^T \omega_{\mathrm{s}}^{0,0}$$

Thus

$$\dot{\phi} = \frac{r_s}{r_w} \begin{bmatrix} -\frac{1}{2}\sqrt{2} & 0 & \frac{1}{2}\sqrt{2} \\ \frac{1}{4}\sqrt{2} & -\frac{1}{4}\sqrt{6} & \frac{1}{2}\sqrt{2} \\ \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{6} & \frac{1}{2}\sqrt{2} \end{bmatrix} \left(R_{\mathrm{b}}^0\right)^T \omega_{\mathrm{s}}^{0,0} \tag{6-9}$$

which relates the angular velocity of the omni-wheels and the states of the system. The relation between the travelled angle can be obtained by (numerically) integrating equation (6-9):

$$\phi = \int \dot{\phi}(\omega_{\mathrm{s}}^{0,0}, R_{\mathrm{b}}^0) \tag{6-10}$$

6-1-3 Summary

In this section the available measurements were related to the state variables. The measurement states y are:

$$y = \begin{bmatrix} \omega_{\mathrm{b}}^{\mathrm{b},0} & \frac{acc}{\|acc\|} & \frac{mag}{\|mag\|} & \phi \end{bmatrix}^T \tag{6-11}$$

The corresponding relations are shown below.

Gyroscope:

$$\omega_{\mathrm{b}}^{\mathrm{b},0} = R_0^{\mathrm{b}} \omega_{\mathrm{b}}^{0,0}$$

Accelerometer:

$$\frac{v_a}{\|v_a\|} = R_0^{\mathrm{b}} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

Magnetometer:

$$\frac{v_m}{\|v_m\|} = R_0^{\mathrm{b}} m$$

Encoder: Solve (numerically) the integral:

$$\phi = \int \dot{\phi}(\omega_{\mathrm{s}}^{0,0}, R_{\mathrm{b}}^0)$$

with

$$\dot{\phi} = \frac{r_s}{r_w} \begin{bmatrix} -\frac{1}{2}\sqrt{2} & 0 & \frac{1}{2}\sqrt{2} \\ \frac{1}{4}\sqrt{2} & -\frac{1}{4}\sqrt{6} & \frac{1}{2}\sqrt{2} \\ \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{6} & \frac{1}{2}\sqrt{2} \end{bmatrix} \left(R_{\mathrm{b}}^0\right)^T \omega_{\mathrm{s}}^{0,0}$$

6-1-4 Reconstructing a rotation matrix

In the proposed observers, the measured rotation matrix R_{y} is used. This matrix found using the accelerometer and the magnetometer. These sensors measure two non-collinear vectors and hence the crossproduct is a non-zero vector orthogonal to these two vectors. Using these vectors, it is possible to construct a frame Ψ_{m} and a corresponding rotation matrix R_0^{m}

The first axis of frame Ψ_{m} is chosen parallel to the vector outputted by accelerometer:

$$a_1(t) = \frac{v_a(t)}{\|v_a(t)\|} \tag{6-12}$$

The second axis is given by the vector orthonormal to the plane span by the accelerometer and magnetometer vectors:

$$a_2(t) = \frac{v_a(t) \times v_m(t)}{\|v_a(t) \times v_m(t)\|} \tag{6-13}$$

Finally, the orthonormal basis in $\mathbb{R}^3$ is completed by the vector orthonormal to a_1 and a_2:

$$a_3(t) = \frac{a_1(t) \times a_2(t)}{\|a_1(t) \times a_2(t)\|} \tag{6-14}$$

Now the rotation matrix between the inertial frame and frame Ψ_{m} is given by

$$R_0^{\mathrm{m}}(t) = \begin{bmatrix} a_1(t) & a_2(t) & a_3(t) \end{bmatrix} \tag{6-15}$$

To obtain the relation between this measured rotation matrix and the body rotation matrix R_{b}^0 the system has to be calibrated once at $R_{\mathrm{b}}^0(0) = I$ to find the relative rotation

$R_{\mathrm{b}}^{\mathrm{m}}$:

$$\begin{aligned} R_{\mathrm{b}}^{\mathrm{m}} &= R_0^{\mathrm{m}}(0)R_{\mathrm{b}}^0(0) \\ R_{\mathrm{b}}^{\mathrm{m}} &= R_0^{\mathrm{m}}(0) \end{aligned}$$

Finally, the measured body rotation matrix is given by

$$R_{\mathrm{y}}(t) = R_{\mathrm{b}}^0(t) = \left(R_{\mathrm{b}}^{\mathrm{m}}\right)^T R_0^{\mathrm{m}}(y) \tag{6-16}$$

The angular velocity of the body is measured in body frame. However, in this chapter the angular velocity is regularly expressed in inertia frame. Assuming that frame of the accelerometer and magnetometer are the same as the frame of the gyroscope, the measured angular velocity can be expressed in the inertia frame with

$$\omega_{\mathrm{y}}^{0,0} = R_{\mathrm{y}}\omega_{\mathrm{y}}^{\mathrm{y},0} \tag{6-17}$$

6-2 Global geometric observer on $SO(3)$

In this section an global observer on $SO(3)$ is developed, i.e. systems of the form

$$\begin{cases} \dot{\omega} &= f(R, \omega, u) \\ \dot{R} &= \tilde{\omega}R \end{cases} \tag{6-18}$$

with $R \in SO(3)$, $\tilde{\omega} \in so(3)$ and u denotes the input. Note that the angular velocity is denoted in inertia frame.

In Section (3-3) error functions on $SO(3)$ were derived. For the observer these error functions will be used to drive the estimated states to the true states. Summarising the results and substituting the desired states with the measured states:

> The error functions for $SO(3)$ are defined as:
>
> $$\begin{aligned} e_R &= \frac{1}{2}\left(R_{\mathrm{y}}^T\hat{R} - \hat{R}^T R_{\mathrm{y}}\right)^{\vee} \\ e_\omega^{\mathrm{b}} &= \hat{\omega}_{\mathrm{b}}^{\mathrm{b},0} - \hat{R}^T R_{\mathrm{y}}\omega_{\mathrm{y}}^{\mathrm{y},0} \\ e_\omega^0 &= \hat{\omega}_{\mathrm{b}}^{0,0} - \omega_{\mathrm{y}}^{0,0} \end{aligned}$$
>
> where e_ω^{b} and e_ω^0 are in the estimated body frame or inertia frame respectively.

Note that error function of the angular velocity is expressed on the estimated body frame, which is does not coincide with the actual body frame if $\hat{R} \neq R_{\mathrm{y}}$.

In the remainder of this section the angular velocity and acceleration will be expressed in inertia frame, unless it is stated otherwise. A benefit of choosing the inertia frame is that the derivative of the angular velocity error is given by an Euclidean error, i.e. $e = x - x_y$, while taking the derivative of the angular velocity error in body frame results in additional terms due to the chain rule.

6-2-1 Candidate Lyapunov function

The design of the observers are based on the same candidate Lyapunov function as found in [29] and extents on the proposed filter by including the dynamics and proving global stability of the error functions. The following Lyapunov function is proposed:

$$V = \alpha\phi(R_e) + \frac{1}{2}\left(e_\omega^0\right)^T e_\omega^0 \tag{6-19}$$

where $\alpha > 0$ a scalar and $\phi(R_e)$ is the error function defined by

$$\phi(R_e) = \frac{1}{2}\mathrm{tr}\left(I - R_e\right)$$

It was seen before that $\phi(R_e) \geq 0$ and hence $V > 0$ for $(R_e, e_\omega^0) \neq (I, 0)$.

Now the derivative of V is given by:

$$\begin{aligned}\dot{V} &= -\alpha\frac{1}{2}\mathrm{tr}\left(R_e\tilde{e}_\omega^{\mathrm{b}}\right) + \left(e_\omega^0\right)^T \dot{e}_\omega^0 \\ &= \alpha\left(e_\omega^{\mathrm{b}}\right)^T e_R + \left(e_\omega^0\right)^T \dot{e}_\omega^0\end{aligned}$$

Here the property $\mathrm{tr}(A\tilde{x}) = -x^T\left(A - A^T\right)^\vee$ is used. Now expressing the angular velocity error in inertia frame using $e_\omega^{\mathrm{b}} = R^T e_\omega^0$:

$$\begin{aligned}\dot{V} &= \alpha\left(R^T e_\omega^0\right)^T e_R + \left(e_\omega^0\right)^T \dot{e}_\omega^0 \\ &= \alpha(Re_R)^T e_\omega^0 + \left(e_\omega^0\right)^T \dot{e}_\omega^0\end{aligned} \tag{6-20}$$

Note that the Lyapunov function is completely expressed in inertia frame. To simplify notation, the superscript denoting the frame will be dropped in the remainder. The convergence of the error functions is given in the following lemma

Lemma 6. *The error functions will converge to their desired value, i.e. $(e_\omega, e_R, R_e) \rightarrow (0, 0, I)$, if $\dot{V} < 0$ for all $(e_\omega, e_R, R_e) \neq (0, 0, I)$ and if the initial conditions satisfy assuming that the initial conditions satisfy:*

$$\phi(R_e(0)) < 2 \tag{6-21}$$

$$e_\omega^T(0)e_\omega(0) < 2\,\alpha\left(2 - \phi(R_e(0))\right) \tag{6-22}$$

Proof. Using $\dot{V} < 0$ and the bounds imposed on the initial positions in (6-21) and (6-22), we have:

$$\begin{aligned}\alpha\phi(R_e(t)) &\leq V(t) \leq V(0) < \alpha\phi(R_e(0)) + \alpha\left(2 - \phi(R_e(0))\right) \\ \alpha\phi(R_e(t)) &\leq V(t) \leq V(0) < 2\alpha \\ \phi(R_e(t)) &\leq \frac{1}{\alpha}V(t) \leq \frac{1}{\alpha}V(0) < 2\end{aligned}$$

Hence $\phi(R_e(t))$ remains bounded below 2. As a result the only possible solution to $e_R = 0$ is $R_e = I$, and hence the asymptotic convergence of $(e_\omega, e_R, R_e) = (0, 0, I)$ is proven. □

Now let us denote

$$x_1 = \begin{bmatrix} e_\omega \\ \alpha R e_R \end{bmatrix}, \qquad\qquad x_2 = \begin{bmatrix} \dot{e}_\omega \\ e_\omega \end{bmatrix} \tag{6-23}$$

The derivative of the Lyapunov function can now be rewritten as

$$\dot{V} = x_1^T x_2 \tag{6-24}$$

6-2-2 Proposed observer

The objective is now to find an observer that satisfies $\dot{V} < 0$. In order to satisfy this criteria, the following observer is proposed:

$$\begin{cases} \dot{\hat{\omega}} &= f(\hat{R}, \hat{\omega}, u) - L_1 e_\omega \\ \dot{\hat{R}} &= \left(\hat{\omega} - \alpha L_2 \hat{R} e_R\right)^\sim \hat{R} \end{cases} \tag{6-25}$$

Here $L_1 > 0$ and $L_2 > 0$ are diagonal matrices. Note that the angular velocity in the kinematics is given by

$$\hat{\omega}' = \hat{\omega} - \alpha L_2 \hat{R} e_R \tag{6-26}$$

Given this observer, the error dynamics of x_2 can be rewritten as:

$$x_2 = \begin{bmatrix} \dot{e}_\omega \\ e_\omega \end{bmatrix} = -L x_1 + F \tag{6-27}$$

where $L > 0$ is a diagonal matrix $L = \text{diag}(L_1, L_2)$ and F a residual term given by

$$F = \begin{bmatrix} f(\hat{R}, \hat{\omega}, u) - f(R, \omega, u) \\ \hat{\omega} - \omega \end{bmatrix} \tag{6-28}$$

In order to prove stability it is required to make the following assumption that is very reminiscent to the global Lipschitz condition. A system is globally Lipschitz if

$$\|f(x,u) - f(z,u)\| \leq K\|x - z\| \qquad\qquad \forall x, z, u \tag{6-29}$$

Now for the system, assume that

$$\|F\| \leq K\|x_1\| \qquad\qquad \forall \hat{\omega}, \hat{R}, \omega, R, u \tag{6-30}$$

with x_1 and F given in (6-23) and (6-28) respectively. Now the derivative of the candidate Lyapunov function of (6-24) becomes:

$$\begin{aligned} \dot{V} &= -x_1^T L x_1 + x_1^T F \\ &\leq -x_1^T L x_1 + K\|x_1\|^2 \end{aligned}$$

Given that the assumptions in (6-30), (6-21) and (6-22) hold and also assuming $x_1 L x_1 > K\|x_1\|^2$ global asymptotic convergence of the error functions is achieved. Note however that these assumptions can be difficult to prove.

6-2-3 Simulation Results

To conclude this section the effectiveness of the observer is shown in simulation for a spherical pendulum.

In this example the global observer on $SO(3)$ will be demonstrated for a 3D pendulum. The equations of motion of such a pendulum denoted in body frame is given by [31]

$$\begin{cases} \dot{\omega}_{\mathrm{b}}^{\mathrm{b},0} &= J_{\mathrm{b}}^{-1}\left(\left(J_{\mathrm{b}}\omega_{\mathrm{b}}^{\mathrm{b},0}\right)^{\sim}\omega_{\mathrm{b}}^{\mathrm{b},0} - m_{\mathrm{b}}g\ \tilde{p}_{\mathrm{b}}^{0}\left(R_{\mathrm{b}}^{0}\right)^{T}e_3^T + \tau^{\mathrm{b,b}}\right) \\ \dot{R}_{\mathrm{b}}^{0} &= R_{\mathrm{b}}^{0}\tilde{\omega}_{\mathrm{b}}^{\mathrm{b},0} \end{cases} \tag{6-31}$$

with $e_3^T = \begin{bmatrix}0 & 0 & 1\end{bmatrix}$. The equations can be easily rewritten to inertia frame, using $\dot{\omega}_{\mathrm{b}}^{0,0} = R_{\mathrm{b}}^{0}\dot{\omega}_{\mathrm{b}}^{\mathrm{b},0}$ and $\omega_{\mathrm{b}}^{\mathrm{b},0} = R_{0}^{\mathrm{b}}\omega_{\mathrm{b}}^{0,0}$:

$$\begin{cases} \dot{\omega}_{\mathrm{b}}^{0,0} &= R_{\mathrm{b}}^{0}\ J_{\mathrm{b}}^{-1}\left(\left(J_{\mathrm{b}}\ R_{0}^{\mathrm{b}}\ \omega_{\mathrm{b}}^{0,0}\right)^{\sim}R_{0}^{\mathrm{b}}\ \omega_{\mathrm{b}}^{0,0} - m_{\mathrm{b}}g\ \tilde{p}_{\mathrm{b}}^{0}\left(R_{\mathrm{b}}^{0}\right)^{T}e_3^T + \tau^{0,\mathrm{b}}\right) \\ \dot{R}_{\mathrm{b}}^{0} &= \tilde{\omega}_{\mathrm{b}}^{0,0}R_{\mathrm{b}}^{0} \end{cases} \tag{6-32}$$

The equations expressed in inertia frame result in equations in the form of (6-18). Now the observer of (6-25) is applied to the system, using $L_1 = 10I$,$\alpha = 10$ and $L_2 = I$.

The simulation is done for the pendulum with no control input, i.e. $\tau^{0,\mathrm{b}} = 0$.

The system is initialised with an initial rotation of

$$R_{\mathrm{b}}^{0}(0) = R_{\mathrm{z''}}(\theta_3)R_{\mathrm{y'}}(\theta_2)R_{\mathrm{x}}(\theta_1)$$

with

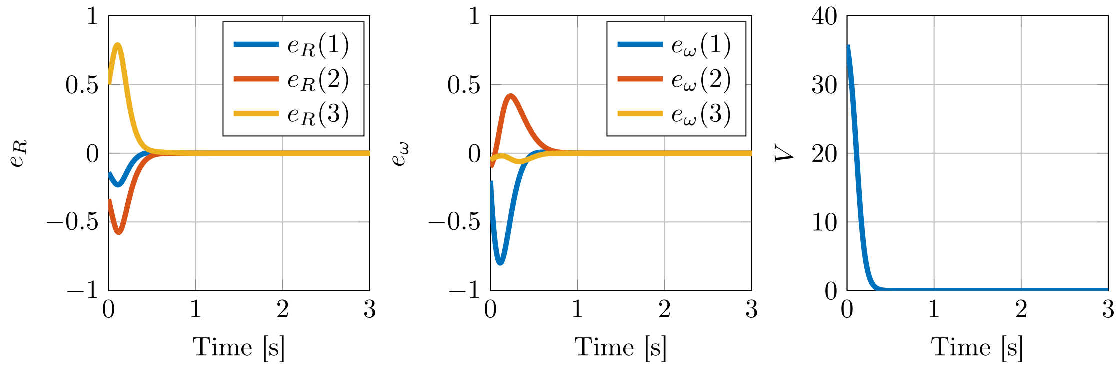
$$R_{\mathrm{x}}(\theta_1) = \begin{bmatrix}1 & 0 & 0\\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1\end{bmatrix} \quad R_{\mathrm{y'}}(\theta_2) = \begin{bmatrix}\cos\theta_2 & 0 & \sin\theta_2\\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2\end{bmatrix} \quad R_{\mathrm{z''}}(\theta_3) = \begin{bmatrix}\cos\theta_3 & -\sin\theta_3 & 0\\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1\end{bmatrix}$$

with $\theta_1 = -\frac{40}{180}\pi$ rad, $\theta_2 = \frac{60}{180}\pi$ rad, $\theta_3 = -\frac{110}{180}\pi$ rad. The initial angular velocities are chosen to be: $\omega(0) = \begin{bmatrix}0.2 & 0.1 & 0.05\end{bmatrix}^T$ rad/s. Note that this parametrisation of the rotation matrix is only done for the sole purpose of given a viable initial rotation matrix and the observer remains coordinate-free. The observer is initialised at $\hat{R} = I$ and $\hat{\omega} = 0$. The error functions e_R and e_ω and the candidate Lyapunov function V are shown in Figure 6-2. It can be seen that the error functions indeed converge to zero. Furthermore, it can be observed that the Lyapunov function decreases monotonically to zero. As a result the estimates converge to their true states.

6-3 Geometric ballbot observer

In the previous section a global observer on $SO(3)$ was developed and its effectiveness was shown for a spherical pendulum. In this section the global observer on $SO(3)$ is extended, such that all observable states of a ballbot can be estimated globally.

The observable states of the ballbot are the angular velocities $\omega_{\mathrm{b}}^{0,0}$, $\omega_{\mathrm{s}}^{0,0}$, the rotation matrix of the body R_{b}^{0} and the encoder angle and velocity φ and $\dot{\varphi}$. The equation of



(a) Attitude error **(b)** Angular velocity error **(c)** Lyapunov function

Figure 6-2: Convergence of the estimation error of an observer on $SO(3)$ for a spherical pendulum

motion of the ballbot can be written in the form

$$\begin{cases} \dot{\omega}_{\mathrm{b}}^{0,0} &= f_1(R_{\mathrm{b}}^0, \omega_{\mathrm{b}}^{0,0}, \omega_{\mathrm{s}}^{0,0}, \tau) \\ \dot{\omega}_{\mathrm{s}}^{0,0} &= f_2(R_{\mathrm{b}}^0, \omega_{\mathrm{b}}^{0,0}, \omega_{\mathrm{s}}^{0,0}, \tau) \\ \dot{R}_{\mathrm{b}}^0 &= \tilde{\omega}_{\mathrm{b}}^{0,0} R_{\mathrm{b}}^0 \\ \dot{\varphi} &= C\ R_0^{\mathrm{b}}\ \omega_{\mathrm{s}}^{0,0} \end{cases} \tag{6-33}$$

Here C denotes a constant full rank matrix. Furthermore, notice that velocities and accelerations are written in the inertia frame Ψ_0. In the remainder of this chapter the arguments of the functions f_1 and f_2 are dropped to ease notation.

In this section it is assumed that the measured states are the angular velocity of the body $\omega_y = \omega_{\mathrm{b}}^{0,0}$, the attitude of the body in the form of a rotation matrix $R_{y_{\mathrm{b}}} = R_{\mathrm{b}}^0$ and the encoder angles $\varphi_y = \varphi$.

Candidate Lyapunov function In the previous section a Lyapunov function for $SO(3)$ was proposed. This same Lyapunov function will augmented for the ballbot observer. The proposed Lyapunov function has the form

$$V = V_1 + V_2 \tag{6-34}$$

where V_1 and V_2, corresponding to the states of the body and sphere respectively, are given by:

$$V_1 = \alpha\phi(R_e) + \frac{1}{2}\left(e_{\omega_{\mathrm{b}}}\right)^T e_{\omega_{\mathrm{b}}} \tag{6-35}$$

$$V_2 = \frac{1}{2} e_\varphi^T e_\varphi + \frac{1}{2}\left(e_{\omega_{\mathrm{s}}}\right)^T e_{\omega_{\mathrm{s}}} \tag{6-36}$$

Here R_e denotes the error rotation matrix of the body frame Ψ_{b}, $e_{\omega_{\mathrm{b}}}$ the angular velocity error of the body frame Ψ_{b} expressed in inertia frame and $e_{\omega_{\mathrm{s}}}$ the angular velocity error of the sphere frame Ψ_{s} expressed in inertia frame. Finally, e_φ denotes the error between the measured encoder angles and estimated angles:

$$\begin{aligned} e_{\omega_{\mathrm{b}}} &= \hat{\omega}_{\mathrm{b}}^{0,0} - \omega_{\mathrm{b}}^{0,0} \\ R_e &= R_0^{\mathrm{b}} \hat{R}_{\mathrm{b}}^0 \\ e_{\omega_{\mathrm{s}}} &= \hat{\omega}_{\mathrm{s}}^{0,0} - \omega_{\mathrm{s}}^{0,0} \\ e_\varphi &= \hat{\varphi} - \varphi \end{aligned}$$

The derivative of the candidate Lyapunov function is given by

$$\dot{V} = \alpha (Re_R)^T e_{\omega_{\mathrm{b}}} + (e_{\omega_{\mathrm{b}}})^T \dot{e}_{\omega_{\mathrm{b}}} + e_\varphi^T \dot{e}_\varphi + (e_{\omega_{\mathrm{s}}})^T \dot{e}_{\omega_{\mathrm{s}}} \tag{6-37}$$

Now let us relate the errors of the encoders angles to the angular velocity errors. Inverting the relation $\dot{\varphi} = C\ R_0^{\mathrm{b}}\ \omega_{\mathrm{s}}^{0,0}$, yields

$$\omega_{\mathrm{s}}^{0,0} = R_{\mathrm{b}}^0\ C^{-1}\ \dot{\varphi} \tag{6-38}$$

Recall that C is full rank and thus invertible. The angular velocity error in inertia frame can now be expressed as

$$\begin{aligned} e_{\omega_{\mathrm{s}}} &= \hat{\omega}_{\mathrm{s}}^{0,0} - \omega_{\mathrm{s}}^{0,0} \\ &= \hat{R}_{\mathrm{b}}^0\ C^{-1}\ \dot{\hat{\varphi}} - R_{\mathrm{b}}^0\ C^{-1}\ \dot{\varphi} \end{aligned}$$

Notice that two different rotation matrices are used, as both angular velocities are on different tangent spaces. However, this also results in a complex error function and would heavily increase the complexity of the design of the observer. Hence, to simplify it is assumed that the attitude observer rapidly converges:

$$\begin{aligned} e_{\omega_{\mathrm{s}}} &\approx \hat{R}_{\mathrm{b}}^0\ C^{-1}\ \dot{\hat{\varphi}} - \hat{R}_{\mathrm{b}}^0\ C^{-1}\ \dot{\varphi} \\ &\approx \hat{R}_{\mathrm{b}}^0\ C^{-1}\ \dot{e}_\varphi \end{aligned} \tag{6-39}$$

Substituting (6-39) into the derivative of the Lyapunov function of (6-37):

$$\dot{V}_2 = \alpha (Re_R)^T e_{\omega_{\mathrm{b}}} + (e_{\omega_{\mathrm{b}}})^T \dot{e}_{\omega_{\mathrm{b}}} + e_\varphi^T \dot{e}_\varphi + \left(\hat{R}_{\mathrm{b}}^0\ C^{-1} \dot{e}_\varphi\right)^T \dot{e}_{\omega_{\mathrm{s}}} \tag{6-40}$$

Now let us denote

$$x_1 = \begin{bmatrix} e_{\omega_{\mathrm{b}}} \\ \alpha Re_R \end{bmatrix}, \qquad x_2 = \begin{bmatrix} \dot{e}_{\omega_{\mathrm{b}}} \\ e_{\omega_{\mathrm{b}}} \end{bmatrix} \qquad x_3 = \begin{bmatrix} \hat{R}_{\mathrm{b}}^0\ C^{-1} \dot{e}_\varphi \\ e_\varphi \end{bmatrix}, \qquad x_4 = \begin{bmatrix} \dot{e}_{\omega_{\mathrm{s}}} \\ \dot{e}_\varphi \end{bmatrix} \tag{6-41}$$

The derivative of the Lyapunov function can now be rewritten as

$$\dot{V} = x_1^T x_2 + x_3^T x_4 \tag{6-42}$$

The convergence of the error dynamics is given in the following lemma.

Lemma 7. *The error functions will converge to their desired value, i.e.* $(e_{\omega_{\mathrm{b}}}, e_R, R_e, e_{\omega_{\mathrm{s}}}, e_\varphi) \to (0, 0, I, 0, 0)$*, if* $\dot{V} < 0$ *for all* $(e_\omega, e_R, R_e) \neq (0, 0, I)$ *and if the initial conditions satisfy assuming that the initial conditions satisfy:*

$$\phi(R_e(0)) < 2 \tag{6-43}$$

$$e_{\omega_{\mathrm{b}}}^T(0) e_{\omega_{\mathrm{b}}}(0) + e_\varphi^T(0) e_\varphi(0) + e_{\omega_{\mathrm{s}}}^T(0) e_{\omega_{\mathrm{s}}}(0) < 2\ \alpha\left(2 - \phi(R_e(0))\right) \tag{6-44}$$

Proof. Using $\dot{V} < 0$ and the bounds imposed on the initial positions in (6-43) and (6-44), we have:

$$\begin{aligned} \alpha\phi(R_e(t)) &\leq V(t) \leq V(0) < \alpha\phi(R_e(0)) + \alpha\left(2 - \phi(R_e(0))\right) \\ \alpha\phi(R_e(t)) &\leq V(t) \leq V(0) < 2\alpha \\ \phi(R_e(t)) &\leq \frac{1}{\alpha}V(t) \leq \frac{1}{\alpha}V(0) < 2 \end{aligned}$$

Hence $\phi(R_e(t))$ remains bounded below 2. As a result the only possible solution to $e_R = 0$ is $R_e = I$, and hence the asymptotic convergence of

$$(e_{\omega_\mathrm{b}}, e_R, R_e, e_{\omega_\mathrm{s}}, e_\varphi) \to (0, 0, I, 0, 0)$$

is proven. □

Proposed observer Again an observer is designed such that $\dot{V} < 0$ for $(e_{\omega_\mathrm{b}}, e_R, e_{\omega_\mathrm{s}}, e_\varphi) \neq (0, 0, 0, 0)$. To achieve this, the error dynamics of x_2 and x_3 are designed to be:

$$x_2 = \begin{bmatrix} \dot{e}_{\omega_\mathrm{b}} \\ e_{\omega_\mathrm{b}} \end{bmatrix} = -L_b x_1 + F_b \qquad x_4 = \begin{bmatrix} \dot{e}_{\omega_\mathrm{s}} \\ \dot{e}_\varphi \end{bmatrix} = -L_s x_3 + F_s \tag{6-45}$$

where $L_b > 0$ and $L_s > 0$ are diagonal matrices $L_b = \mathrm{diag}(L_1, L_2)$, $L_s = \mathrm{diag}(L_3, L_4)$ and F_b and F_s residual terms given by

$$F_b = \begin{bmatrix} \hat{f}_1 - f_1 \\ \hat{\omega}_\mathrm{b} - \omega_\mathrm{b} \end{bmatrix} \qquad F_s = \begin{bmatrix} \hat{f}_2 - f_2 \\ C\left(\hat{R}^0_\mathrm{b}\right)^T \hat{\omega}_\mathrm{s} - C\left(R^0_\mathrm{b}\right)^T \omega_\mathrm{s} \end{bmatrix} \tag{6-46}$$

However, the observer cannot match these error dynamics, as $\dot{e}_\varphi$ is unknown, since $\dot{\varphi}$ is not measured. Therefore a virtual encoder velocity is introduced:

$$\dot{\varphi}_{y,\mathrm{virtual}} = C\left(\hat{R}^0_\mathrm{b}\right)^T \hat{\omega}_\mathrm{s} - L_3 e_\varphi \tag{6-47}$$

Hence the encoder velocity error becomes approximately

$$\dot{e}_\varphi \approx L_3 e_\varphi \tag{6-48}$$

In order to prove stability again an assumption is made very reminiscent to the global Lipschitz condition. Assume that

$$\|F_b\| \leq K_b\|x_1\|, \qquad \|F_s\| \leq K_s\|x_3\| \tag{6-49}$$

with x_1, x_3, F_b and F_s given in (6-41) and (6-46) respectively. Now the derivative of the candidate Lyapunov function of (6-42) becomes:

$$\begin{aligned} \dot{V} &= -x_1^T L_b x_1 + x_1^T F_b - x_3^T L_s x_3 + x_3^T F_s \\ &\leq -x_1^T L_b x_1 + K_b\|x_1\|^2 - x_3^T L_s x_3 + K_s\|x_2\|^2 \end{aligned}$$

Given that the assumptions in (6-49), (6-43), (6-44) hold and also assuming $x_1^T L_b x_1 > K_b \|x_1\|^2$ and $x_3^T L_s x_3 > K_s \|x_3\|^2$ global asymptotic convergence of the error functions is achieved.

The resulting observer is given by:

$$\begin{cases} \dot{\hat{\omega}}_\mathrm{b} &= f_1(\hat{R}_\mathrm{b}^0, \hat{\omega}_\mathrm{b}, \hat{\omega}_\mathrm{s}, \tau) - L_1 e_\omega^\mathrm{b} - e_R \\ \dot{\hat{\omega}}_\mathrm{s} &= f_2(\hat{R}_\mathrm{b}^0, \hat{\omega}_\mathrm{b}, \hat{\omega}_\mathrm{s}, \tau) - L_3 \hat{R}_\mathrm{b}^0 \; C^{-1} L_4 e_\varphi \\ \dot{\hat{R}}_\mathrm{b} &= (\hat{\omega}_\mathrm{b} - \alpha L_2 R e_R)^\sim \; \hat{R}_\mathrm{b}^0 \\ \dot{\varphi} &= C \left(\hat{R}_\mathrm{b}^0\right)^T \hat{\omega}_\mathrm{s} - L_4 e_\varphi \end{cases} \tag{6-50}$$

Again the proposed observer is relies on assumptions that might be difficult to verify.

Summary

Given the equations of motion of the robot:

$$\begin{cases} \dot{\omega}_\mathrm{b}^{0,0} &= f_1(R_\mathrm{b}^0, \omega_\mathrm{b}^{0,0}, \omega_\mathrm{s}^{0,0}, \tau) \\ \dot{\omega}_\mathrm{s}^{0,0} &= f_2(R_\mathrm{b}^0, \omega_\mathrm{b}^{0,0}, \omega_\mathrm{s}^{0,0}, \tau) \\ \dot{R}_\mathrm{b}^0 &= \tilde{\omega}_\mathrm{b}^{0,0} R_\mathrm{b}^0 \\ \dot{\varphi} &= C \; R_0^\mathrm{b} \; \omega_\mathrm{s}^{0,0} \end{cases}$$

the following observer will result in asymptotic convergence of the state estimations to the true states

$$\begin{cases} \dot{\hat{\omega}}_\mathrm{b} &= f_1(\hat{R}_\mathrm{b}^0, \hat{\omega}_\mathrm{b}, \hat{\omega}_\mathrm{s}, \tau) - L_1 e_\omega^\mathrm{b} - e_R \\ \dot{\hat{\omega}}_\mathrm{s} &= f_2(\hat{R}_\mathrm{b}^0, \hat{\omega}_\mathrm{b}, \hat{\omega}_\mathrm{s}, \tau) - L_3 \hat{R}_\mathrm{b}^0 \; C^{-1} L_4 e_\varphi \\ \dot{\hat{R}}_\mathrm{b} &= (\hat{\omega}_\mathrm{b} - \alpha L_2 R e_R)^\sim \; \hat{R}_\mathrm{b}^0 \\ \dot{\varphi} &= C \left(\hat{R}_\mathrm{b}^0\right)^T \hat{\omega}_\mathrm{s} - L_4 e_\varphi \end{cases}$$

provided the following conditions on the initial condition

$$\phi(R_e(0)) < 2$$
$$e_{\omega_\mathrm{b}}^T(0) e_{\omega_\mathrm{b}}(0) + e_\varphi^T(0) e_\varphi(0) + e_{\omega_\mathrm{s}}^T(0) e_{\omega_\mathrm{s}}(0) < 2 \; \alpha \left(2 - \phi(R_e(0))\right)$$

the global Lipschitz conditions hold for all $t \geq 0$:

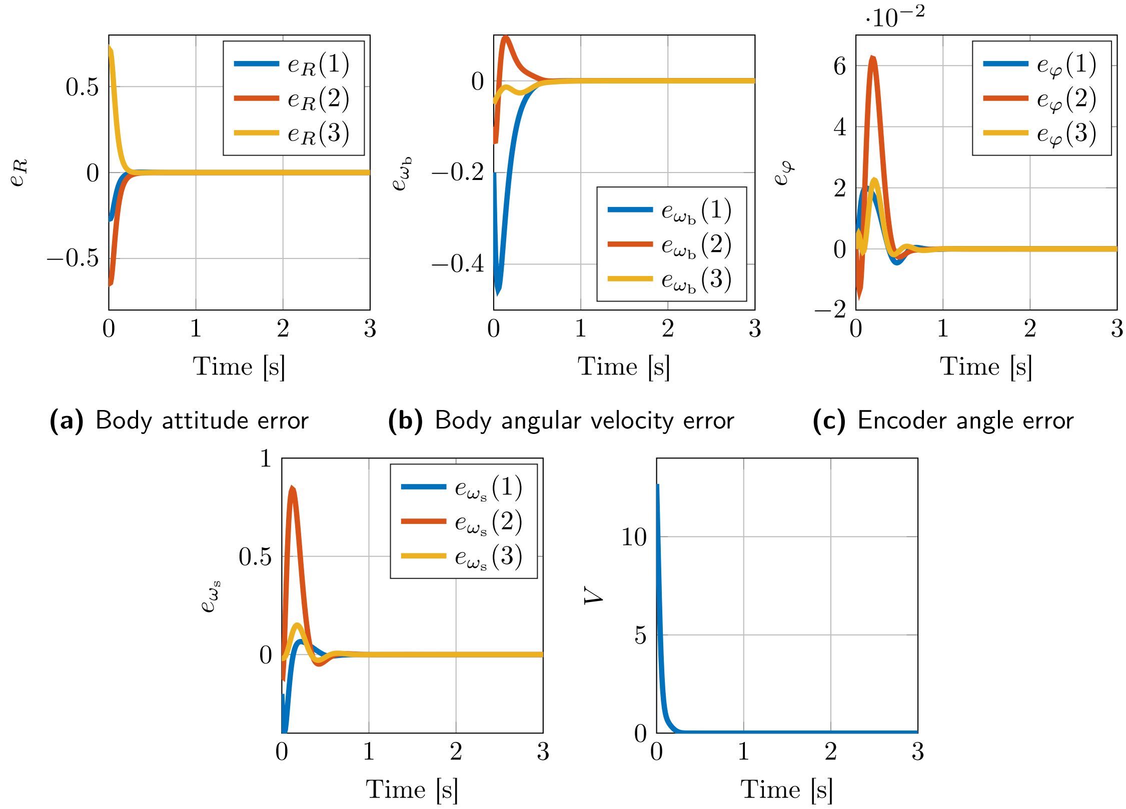
$$\|F_b\| \leq K_b \|x_1\|$$
$$\|F_s\| \leq K_s \|x_3\|$$

and assuming that the following assumption holds for all $t \geq 0$

$$x_1^T L_b x_1 > K_b \|x_1\|^2$$
$$x_3^T L_s x_3 > K_s \|x_3\|^2$$

6-3-1 Simulation results

In this section the effectiveness of the proposed observer is shown in simulation. The initial rotation matrix is again initialised at a rotation with $\theta_1 = -\frac{40}{180}\pi$, $\theta_2 = \frac{60}{180}\pi$, $\theta_3 = -\frac{110}{180}\pi$, where the same parametrisation as in Section 6-2-3 is used. . The initial angular velocities of the body are chosen to be: $\omega_{\mathrm{b}}^{0,0}(0) = \begin{bmatrix} 0.2 & 0.1 & 0.05 \end{bmatrix}^T$ rad/s and the angular velocities of the sphere are chosen to be $\omega_{\mathrm{s}}^{0,0}(0) = \begin{bmatrix} -0.2 & 0.1 & -0.02 \end{bmatrix}^T$. The estimated states are initialised at zero. The observer parameters are chosen to be $L_1 = 10$, $\alpha = 20$, $L_2 = I$, $L_3 = 10I$, $L_4 = 20I$. The error functions e_R and e_ω over time are shown in Figure 6-2. It can be seen that the error functions and the candidate Lyapunov function indeed converge to zero and as a result the estimates converge to their true states.



(a) Body attitude error **(b)** Body angular velocity error **(c)** Encoder angle error

(d) Sphere angular velocity error **(e)** Canditate Lyapunov function

Figure 6-3: Error functions of the geometric observer for the ballbot

6-4 Discussion

The proposed observer has some advantages and disadvantages. First of all, the design of the observer is relative simple and uses the error functions for $SO(3)$ and tries to capture the underlying manifolds as much as possible, without parametrising the differential manifolds or using local charts.

However, the downside is that the assumptions and conditions that guarantee stability are be difficult to prove. Furthermore, the observer does not account for measurement noise such as for example an Extended Kalman filter or a Unscented Kalman filter. However, the performance of an Extended Kalman filter requires the computation of Jacobians and its performance decreases for high non-linearities. On the other hand, the Unscented Kalman filter is able to cope with high non-linearities, but requires a good initialisation of the initial states and no global convergence is not guaranteed.

Chapter 7

Implementation

In this chapter an attempt is made to control the real system. First of all the parameters of the system are identified. Secondly a linear approach is proposed for the implementation in the robot. However, the implementation of this linear approach was unsuccessful. This chapter will be concluded by a detailed discussion on the implementation.

7-1 Parameters

The derived model is subjected to certain systems parameters, such as for example the mass of the body. This section is devoted to identifying and measuring these parameters for the ballbot at Alten Mechatronics.

All model parameters, their description and value are shown in Table 7-1. The remainder of this section is devoted to highlight the more involved measurements or calculations of these parameters.

Table 7-1: Model parameters

Parameter	Description	Value	Unit
g	Gravitational acceleration	9.81	$\mathrm{m/s^2}$
m_s	Mass of the sphere	3.07	kg
m_b	Mass of the body	12.97	kg
j_s	Moment of inertia of the sphere in Ψ_s	0.0255	$\mathrm{kg \cdot m^2}$
$j_{\mathrm{b},x}$	Moment of inertia of the robot in Ψ_b, around the x axis	1.400	$\mathrm{kg \cdot m^2}$
$j_{\mathrm{b},y}$	Moment of inertia of the robot in Ψ_b, around the y axis	1.406	$\mathrm{kg \cdot m^2}$
$j_{\mathrm{b},z}$	Moment of inertia of the robot in Ψ_b, around the z axis	0.1831	$\mathrm{kg \cdot m^2}$
p_b^r	Translation from Ψ_b to Ψ_r	$\begin{bmatrix} 0 & 0 & 0.4 \end{bmatrix}^T$	m
r_s	Radius of the sphere	0.115	m
r_w	Radius of the omni-wheels	0.050	m

7-1-1 Inertia of the sphere

The ball in the ballbot of Alten is a medicine ball of approximately 3 kg. Furthermore, it is known that the ball is hollow with a thickness of between 6 and 8 mm. For a spherical shell, with inner radius r_i and outer radius r_o, the moment of inertia is given by:

$$I = \frac{2m}{5}\left(\frac{r_o^5 - r_i^5}{r_o^3 - r_i^3}\right) \tag{7-1}$$

7-1-2 Inertia of the body

For a swinging downward pendulum, it is relative easy to derive the moment of inertia. The equations of motion for a downward pendulum is given by:

$$j_i\ddot{\phi}_i = -mgl_i\sin(\phi_i) \tag{7-2}$$

Here $i \in \{x, y, z\}$ denotes the axis of rotation and l_i the distance between the center of mass and the axis of rotation. Assuming small angles, we can approach the sine as $\sin(\phi_i) \approx \phi_i$, resulting in following second-order linear differential equation:

$$\ddot{\phi}_i + K\phi_i = 0 \tag{7-3}$$

with

$$K = \frac{mgl_i}{j_i} \tag{7-4}$$

Where m, g, and l are assumed to be known. The solution of this differential equation is

$$\phi(t) = e^{i\sqrt{K}t} \tag{7-5}$$

Note that in this equation i denotes the complex number, and not the axis of rotation. Now using Euler's formula it is possible to rewrite the complex exponential as a sinusoid:

$$\phi_i = \cos\left(\sqrt{K}t\right) + i\sin\left(\sqrt{K}t\right) \tag{7-6}$$

From this notation, it can be observed that the angular velocity is $\omega_i = \sqrt{K}$ rad/s. Finally, it is possible to find a relation between the period T and the moment of inertia:

$$T = 2\pi\sqrt{\frac{j_i}{mgl_i}} \tag{7-7}$$

So far we have derived a simple relation between the period and moment of inertia of a swinging pendulum for small angles. By suspending the top of the robot on a bar and letting the robot swing in a fixed direction, as shown in Figure 7-1, it is possible to measure the period of the angle. Henceforth it is possible to derive the moment of inertia using equation (7-7).

Note that the measured inertia is around the axis of rotation and not around the center of mass of the body. However, it is possible to relate the found inertia with the

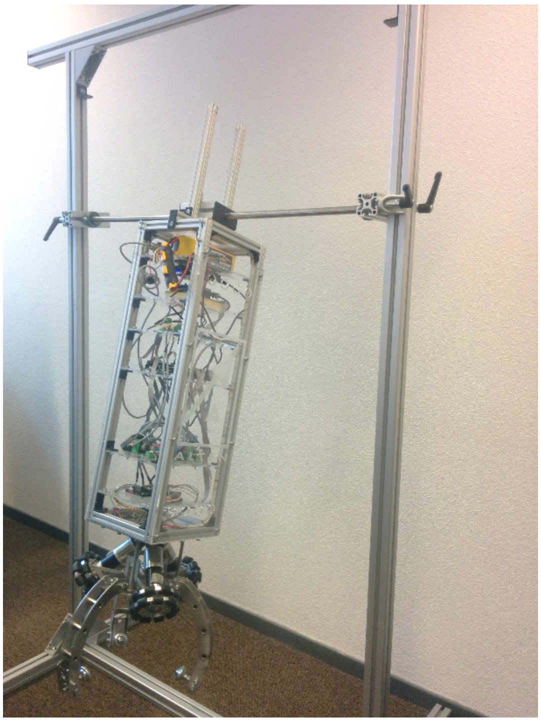


Figure 7-1: Inertia measurement test. The top of the robot is suspended to a bar and oscillates around the stable equilibrium point.

moment of inertia around the body frame, using the relations between inertia tensor and different frames. However, equivalent the parallel axis theorem can be used:

$$j_i = j_{\mathrm{b},i} + m_{\mathrm{b}} l_i^2 \tag{7-8}$$

Alternatively to determining the period by hand, the `fit()` function from MATLAB is used to fit a sine of the form $a_i \sin(b_i x + c_i)$, where b_i is equal to $\sqrt{K}$. Then using

$$j_i = \frac{mgl_i}{b_i^2}$$

the inertia is determined. An example of a measurement and fit is shown in Figure 7-2. Note that while there is a slight mismatch in amplitude, the frequency of both signals is equal, which is the variable that is used to determine the inertia. The resulting body inertias are shown in Table 7-2.

Table 7-2: Body inertia measurements [kg$\cdot$ m^2]

Parameter	Measurements			Average
$j_{\mathrm{b},x}$	1.390	1.408	1.402	1.400
$j_{\mathrm{b},y}$	1.413	1.405	1.402	1.406
$j_{\mathrm{b},z}$	0.1823	0.1830	0.1840	0.1831

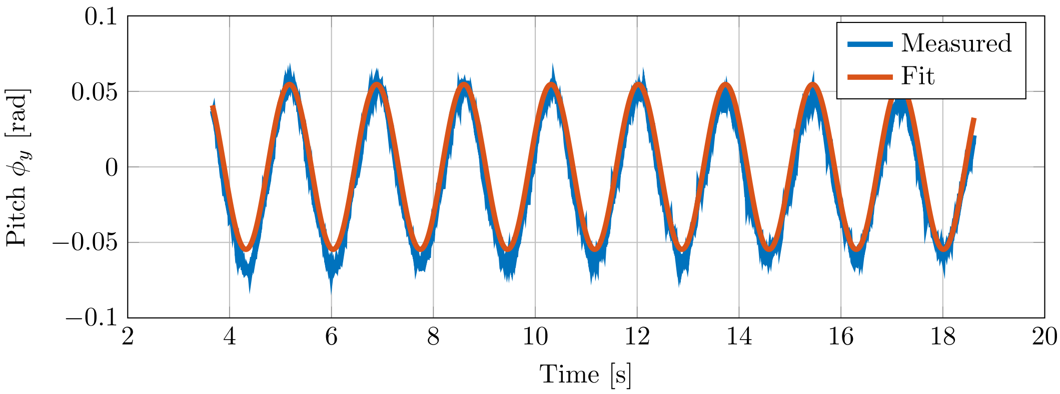


Figure 7-2: Example of the measured pitch angle (blue) and corresponding fitted sinus (red).

7-2 Linear approach

In the preceding work on the ball-balancing robot at Alten Mechatronics [7] an attempt was made to control the robot using a linear controller. However, due to multiple reasons this controller failed to properly stabilise the system.

First of all, the body of the robot was not constrained to the ball. As a result the body could fall of the sphere. Therefore a ball-gripper has been added to the robot that constraints the ball to the omni-wheels.

Secondly, a combination of measurement noise and process noise resulted in instability of the system. In order to counter this a Kalman filter will be applied to the system.

Before the implementation of the geometric algorithms in Chapter 5 and 6, it is chosen to first implement the improved linear control algorithm proposed in the preceding work [7]. The motivation is to start a more 'simple' control algorithm design before implementing the more complex geometric control methods. Furthermore, in literature this linear approach has already been proven successful for similar robots.

7-2-1 Linearisation

Before the equations of motion are linearised, state variables will be chosen. In the modelling chapter it was chosen to model the system using rotation matrices and translations. For the LQR controller it is chosen to parametrise these rotation matrices with Euler angles. More specifically, the Tait-Bryan angles $\theta = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3\end{bmatrix}^T$ are used, which represent the sequential rotations $x - y' - z''$. The rotation matrix R_{b}^0 is given by:

$$R_{\mathrm{b}}^0 = R_{\mathrm{z''}}(\theta_3) R_{\mathrm{y'}}(\theta_2) R_{\mathrm{x}}(\theta_1) \tag{7-9}$$

with

$$R_{\mathrm{x}}(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{bmatrix} \quad R_{\mathrm{y'}}(\theta_2) = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \quad R_{\mathrm{z''}}(\theta_3) = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The derivatives of the Euler angles can be related to the angular velocity $\omega_{\text{b}}^{0,0}$:

$$\begin{aligned}\tilde{\omega}_{\text{b}}^{0,0} &= \dot{R}_{\text{b}}^{0} R_{0}^{\text{b}} \\ \omega_{\text{b}}^{0,0} &= \begin{bmatrix} \dot{\theta}_1 \cos\theta_2 \cos\theta_3 - \dot{\theta}_2 \sin\theta_3 \\ \dot{\theta}_2 \cos\theta_3 + \dot{\theta}_1 \cos\theta_2 \sin\theta_3 \\ \dot{\theta}_3 - \dot{\theta}_1 \sin\theta_2 \end{bmatrix}\end{aligned}$$

For the linear controller, the equations of motion are linearised around $\theta = 0$, $\dot{\theta} = 0$. Linearising $\omega_{\text{b}}^{0,0}$ around the same operating point yields

$$\dot{\theta} \approx \omega_{\text{b}}^{0,0}$$

Finally, it is desired to control the position of the robot, and not necessarily the angular velocity and rotation of the sphere. Hence, $v_{\text{k}}^{0,0} \in \mathbb{R}^2$ and $p_{\text{k}}^{0} \in \mathbb{R}^2$ are used to parametrise the equations of motion used for the synthesis of the controller. The relation between the linear velocities and acceleration are obtained directly from the rolling constraint, using

$$\omega_{\text{s,x}}^{0,0} = -\frac{v_{\text{k,y}}^{0,0}}{r_s} \tag{7-10}$$

$$\omega_{\text{s,y}}^{0,0} = \frac{v_{\text{k,x}}^{0,0}}{r_s} \tag{7-11}$$

Furthermore, for simplicity $\omega_{\text{s,z}}^{0,0}$ is assumed to be zero.

Given the above transformations, the states used for the linearised model are given by

$$x = \begin{bmatrix} v_{\text{k}}^{0,0} \\ \omega_{\text{b}}^{0,0} \\ p_{\text{k}}^{0} \\ \theta \end{bmatrix}, \qquad \dot{x} = \begin{bmatrix} \dot{v}_{\text{k}}^{0,0} \\ \dot{\omega}_{\text{b}}^{0,0} \\ v_{\text{k}}^{0,0} \\ \omega_{\text{b}}^{0,0} \end{bmatrix} \tag{7-12}$$

and the input is given by $u = \tau$.

The linear model is linearised around the upright equilibrium, i.e.

$$x_e = 0_{10\times 1}, \;\; u_e = 0_{3\times 1} \tag{7-13}$$

Given the nonlinear equations from the previous section and the new states x, the dynamics can again be expressed as

$$\dot{x} = f(x, u)$$

The linear system matrices are found using a first order Taylor expansion and are given by:

$$A = \left.\frac{(\partial f(x,u))}{\partial x}\right|_{x_e,u_e} \qquad B = \left.\frac{\partial\,(f(x,u))}{\partial u}\right|_{x_e,u_e}$$

7-2-2 LQR controller

In this section a linear controllers are designed, namely a LQR controller. The use of an LQR controller has been done before for ballbots in e.g. [5, 7]. The LQR controller is synthesised using the linearised equations of motion. For simplicity, the Q and R matrices are chosen equal to the matrices found in the preceding work [7]. Using MATLAB, the optimal gain matrix K for the linearised system can be found. The final control law is then given by

$$u = -Kx \tag{7-14}$$

7-2-3 Observer design

In this section a observer is constructed to estimate the system states, given the measurements y. The Euler angles θ are directly computed from the accelerometer and magnetometer or the constructed rotation matrix R_y as described in subsection 6-1-4:

$$\theta_1 = \arctan\left(\frac{R_y(3,2)}{R_y(3,3)}\right), \qquad \theta_2 = \arcsin\left(-R_y(3,1)\right), \qquad \theta_3 = \arcsin\left(\frac{R_y(2,1)}{\cos\theta_2}\right)$$

Therefore, the measurements y is given by:

$$\begin{aligned} y &= \begin{bmatrix}\omega_{\mathrm{b}}^{\mathrm{b},0} & \theta & \phi\end{bmatrix}^T \\ &= \begin{bmatrix}R_0^{\mathrm{b}}\omega_{\mathrm{b}}^{0,0} & \theta & \phi\end{bmatrix}^T \end{aligned}$$

From the previous section, the states for the linear model of the controller were defined as

$$x_c = \begin{bmatrix} v_{\mathrm{k}}^{0,0} \\ \omega_{\mathrm{b}}^{0,0} \\ p_{\mathrm{k}}^{0} \\ \theta \end{bmatrix}, \qquad \dot{x}_c = \begin{bmatrix} \dot{v}_{\mathrm{k}}^{0,0} \\ \dot{\omega}_{\mathrm{b}}^{0,0} \\ v_{\mathrm{k}}^{0,0} \\ \omega_{\mathrm{b}}^{0,0} \end{bmatrix}$$

However, it is not possible to use these states directly in the observer design. First of all, the measured encoder angles can only be related to the state variables through its derivative. Therefore the states are augmented with the encoder angles. Secondly, the angular velocities of the omni-wheels $\dot{\phi}$ are a function of the angular velocity of the sphere $\omega_{\mathrm{s}}^{0,0}$, which cannot be computed from the chosen states. Therefore also the angular velocity $\omega_{\mathrm{s,z}}^{0,0}$ is included in the observer states. Here the following state transformation holds:

$$\begin{aligned} \begin{bmatrix} v_{\mathrm{k}}^{0,0} \\ \omega_{\mathrm{s,z}}^{0,0} \end{bmatrix} &= \begin{bmatrix} 0 & r_s & 0 \\ -r_s & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \omega_{\mathrm{s}}^{0,0} \\ \omega_{\mathrm{s}}^{0,0} &= \begin{bmatrix} 0 & -\frac{1}{r_s} & 0 \\ \frac{1}{r_s} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{\mathrm{k}}^{0,0} \\ \omega_{\mathrm{s,z}}^{0,0} \end{bmatrix} \end{aligned} \tag{7-15}$$

Finally, the absolute position is not observable [3] and is therefore excluded from the observable states.

The velocity in x and y direction $v_{\mathrm{k}}^{0,0} \in \mathbb{R}^2$ and $\omega_{\mathrm{s,z}}^{0,0}$ are related to the velocity of the omni-wheels using equation (6-9) and (7-15):

$$\dot{\phi} = \frac{r_s}{r_w} \begin{bmatrix} -\frac{1}{2}\sqrt{2} & 0 & \frac{1}{2}\sqrt{2} \\ \frac{1}{4}\sqrt{2} & -\frac{1}{4}\sqrt{6} & \frac{1}{2}\sqrt{2} \\ \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{6} & \frac{1}{2}\sqrt{2} \end{bmatrix} \left(R_{\mathrm{b}}^{0}\right)^T \begin{bmatrix} 0 & -\frac{1}{r_s} & 0 \\ \frac{1}{r_s} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{\mathrm{k}}^{0,0} \\ \omega_{\mathrm{s,z}}^{0,0} \end{bmatrix} \tag{7-16}$$

The resulting observer states are:

$$x_o = \begin{bmatrix} v_{\mathrm{k}}^{0,0} \\ \omega_{\mathrm{s,z}}^{0,0} \\ \omega_{\mathrm{b}}^{0,0} \\ \theta \\ \phi \end{bmatrix}, \qquad \dot{x}_o = \begin{bmatrix} \dot{v}_{\mathrm{k}}^{0,0} \\ \dot{\omega}_{\mathrm{s,z}}^{0,0} \\ \dot{\omega}_{\mathrm{b}}^{0,0} \\ \omega_{\mathrm{b}}^{0,0} \\ \dot{\phi} \end{bmatrix} \tag{7-17}$$

The equations of motion $\dot{x}_o = f(x_o, u)$ are linearised around the upright equilibrium, i.e.

$$x_e = 0_{12\times1}, \quad u_e = 0_{3\times1} \tag{7-18}$$

such that the following form is obtained:

$$\dot{x}_o = A_o x_o + B_o u \tag{7-19}$$

$$y = C_o x_o + D_o u \tag{7-20}$$

The final observer takes the form

$$\dot{\hat{x}}_o = A_o \hat{x}_o + B_o u + L\left(y - C_o \hat{x}_o\right) \tag{7-21}$$

Where L represents the observer gain, which can be designed by e.g. pole placement or through the design of a Kalman filter.

The estimates of the controller states $\hat{x}_c$ can be related to the estimated observer states $\hat{x}_o$, where the position can be obtained through the integration of the estimated velocity:

$$\hat{x}_c = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 \end{bmatrix} \hat{x}_o + \int \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \hat{x}_o \, dt \tag{7-22}$$

However, since the absolute position is not observable, it is very likely that the estimated position will drift from the true position. In Figure 7-3 an example of this drift is shown when the system is initialised for some arbitrary non-zero angles and the observer states initialised at zero. This is an issue that cannot be solved with the current sensors, which was also found in [3]. However, exact position control is often not required.

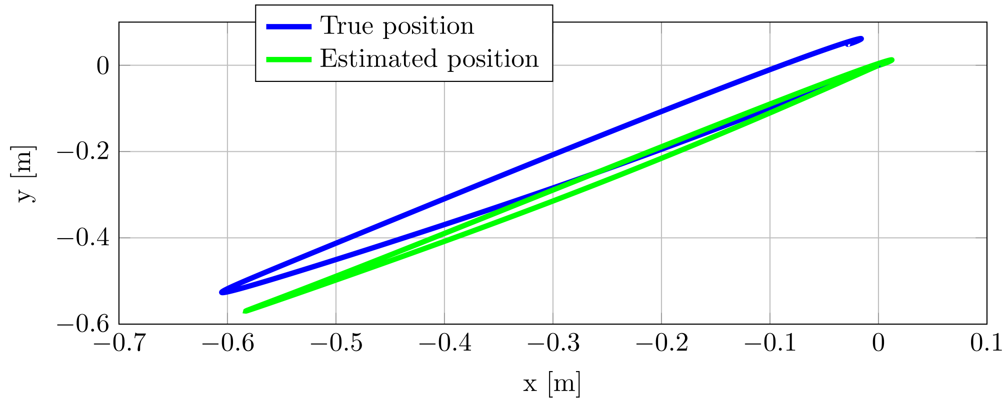


Figure 7-3: Example of the position drift, due to the observability of the position.

7-2-4 Discretisation

The controller and observer described in previous sections are expressed in continuous time. However, the measurements received on the master board are sampled with a frequency of 200 Hz, hence measurements are received in discrete time. In this section the discretisation of the controller and observer is briefly discussed. For the LQR controller, the system matrices are converted from discrete time to continuous time using the `c2d()` command in MATLAB. This command discretises the system using the Zero-order hold (ZOH) method at a frequency of 200 Hz. It is chosen to use the ZOH method, since the controller input to the motors will also be zero-order hold. The corresponding controller gain K is found using the MATLAB function `dlqr()`. The discrete observer gain will be found by designing a discrete time Kalman filter, as will be discussed in the next subsection. Finally, the relation between the estimated controller states and estimated observer states becomes

$$\hat{x}_c(k) = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ h\,I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 \end{bmatrix} \hat{x}_o(k) + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \hat{x}_c(k-1) \tag{7-23}$$

where h denotes the sampling time.

7-2-5 Kalman filter

In reality the system and measurements are subjected to noise, resulting in an perturbed estimate and subsequent a noisy input, hence a Kalman filter will be used to estimate the states with a minimum variance. In this subsection a Kalman filter is designed for a linear time-invariant system. Since the system is time-invariant, there is a stationary solution to the Kalman-filter gain.

The linear equations of motion the system subjected to zero-mean white noise w and v is given by

$$x_o(k+1) = Ax_o(k) + Bu(k) + w(k) \tag{7-24}$$
$$y(k) = Cx_o(k) + v(k) \tag{7-25}$$

Furthermore, the covariance matrix of w and v is given by:

$$E\begin{bmatrix} w(k) \\ v(k) \end{bmatrix} \begin{bmatrix} w(j)v(j) \end{bmatrix} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \Delta(k-j)$$

Initially, it will be assumed that the process noise and measurement noise are uncorrelated, i.e. $S = 0$.

Here R is assumed to be positive definite. Finally, the covariance between of the estimation error is denoted by $P(k|k-1)$:

$$P(k|k-1) = E[(x(k) - \hat{x}(k|k-1))(x(kI - \hat{x}(k|k-1)^T] \tag{7-26}$$

This covariance matrix which will converge to a constant stationary value P since the system is time-invariant and satisfies the discrete algebraic Ricatti equation:

$$P = APA^T + Q - APC^T \left(CPC^T + R\right)^{-1} \left(APC^T\right)^T \tag{7-27}$$

The solution to this discrete algebraic Ricatti equation is found using the MATLAB command `dare()`. The Kalman gain L is then found by

$$L = APC^T \left(CPC^T + R\right)^{-1} \tag{7-28}$$

The discrete time Kalman filter is then given by

$$\hat{x}_o(k+1) = A_o\hat{x}_o(k) + B_o u(k) + L\left(y(k) - C_o\hat{x}_o(k)\right) \tag{7-29}$$

Note that these state matrices are different from the continuous time state matrices in equation (7-21) and are obtained using the `c2d()` command as described before.

The covariance matrix R is identified by measuring the sensor values while the robot is idle and determining the covariance. The resulting covariance matrix R is

$$R = \mathrm{diag}\left(\begin{bmatrix} 0.07203 & 0.07204 & 0.121 & 1.015\cdot 10^{-4} & 5.894\cdot 10^{-5} & 5.784\cdot 10^{-5} & 0 & 0 & 0 \end{bmatrix}\right) \tag{7-30}$$

Furthermore, for the initial implementation, the covariance matrix of the process noise Q is set to be (close to) zero.

7-3 Discussion

Unfortunately, the proposed linear control algorithm has not been implemented successfully on the robot yet. This section will discuss the main challenges, setbacks and resolved issues in the robot.

The main components of the robot consists of a master board, sensor board, Raspberry Pi and motor drivers. The master board and sensor board are both STM32F3 Discovery boards. Both boards are equipped with a gyroscope, accelerometer and magnetometer. The sensor board is dedicated to read out the sensor values and sends it through a SPI connection to the master board. Besides the measurements of the sensor board, the master board also receives the encoder angles from the motor drivers. The master board is dedicated to compute the control algorithm and sends the desired motor input to the motor drivers. Furthermore a logging data is sent over an UART connection to the Raspberry Pi.

The master board, sensor board and Raspberry Pi are programmed in C. The designed observer and controller in MATLAB are converted to C code using the MATLAB code generator. The resulting functions are then called in the master board to compute the desired output, given the sensor values.

On the master board and sensor board FreeRTOS is used to operate the system in real time. The main loop on the master board has a frequency of 200 Hz.

Debugging Debugging the software on the current platform is challenging, because on-board debugging and the logged files are unreliable methods for this specific system.

Let us first discuss the on-board debugging. The system depends on time-critical scheduling and communication between the main boards. The master board waits for measurement data to be received before it computes the control algorithm and sends new data to the motor. By on-board debugging of the master board, it is possible to pause the code at certain breakpoints. However, by pausing the code on the master board, the synchronisation between the master board and other boards is broken and no useful data will be received any more. Furthermore, by pausing the real time operating system the scheduler is interupted, resulting in unreliable results. Thus, while on-board debugging of the master board is possible, it will not provide reliable data.

Now let us discuss the use of logged data to debug the software. The data is logged at the Raspberry Pi and is done by sending relevant data from the master board to the Raspberry Pi through an UART connection. However, there are three downsides to this method. First of all, the amount of data that can be send to the Raspberry Pi is limited by the amount of bytes the UART can send per second. The linear system consist already of 23 unique states which are all stored as floats. Especially when rotation matrices are used, each consisting of 9 floats, the send rate of the UART can become serious limiting factor. Secondly, the more data is send over the UART, the more likely it will be that data can become corrupted or will be missing. As a result it will be difficult to track whether unexpected data is a result of the corrupted UART connection or if something went wrong in the sensor or master board. The third problem is that if the master board should fail to send data to the Raspberry Pi, no data can be consulted for debugging.

A more robust method of debugging the robot is a logic analyser of using the on-board LEDs to signal the current state of the robot.

For example, the communication between the sensor board and master board sometimes failed. This bug occurred especially often if the robot was subjected to jerky movements. Since the master board waits for all sensor data to be received upon continuing the main loop, the robot did not update the motor driver and did not sent data to the Raspberry Pi for logging. Moreover, on-board debugging is not a viable option, as it inherently breaks the synchronisation. The source of the bug was found using the on-board LEDs to signal the exact location of the bug.

Resolved issues In this paragraph some of the resolved issues are listed.

- **The maximum stack size of the controller was capped to a certain maximum**. When implementing the new control algorithm, the amount of variables and therefore also the amount of used floats in the master board main loop increased. This resulted in a hard fault on the master board. This issue was be resolved by increasing the maximum stack size.
- **The communication between the sensor board and master board failed to update.** As a result not all events were passed in order for the main loop to compute the desired motor torque and the motor drivers were not updated again. As a result the motors would maintain a constant torque. This problem was resolved by using the sensors on the master board and by passing the sensor board. This results in a more robust hardware architecture.

Open issues Up till now only an attempt is made to implement the linear control. However, some issues remain that prevent the successful implementation within the time scope of this thesis.

- **Singularities in the computation of the Euler angles from the accelerometer and magnetometer**. Currently a singularity occurs in the computation of the Euler angles if $v_{m,x} = v_{m,y} = 0$, despite the fact that all used functions are well defined. Note that this bug is circumvented when using the geometric controller, which could use the accelerometer and magnetometer vectors directly to measure the attitude.
- **Sensor calibration.** The measurement data of the gyroscope is subjected to a bias. Currently this bias is removed using the first measurement to compensate this bias. However, it would be preferred to calibrate this bias when the system is initialised. Finally, also the initialisation and calibration of the accelerometer and magnetometer should be improved.
- **Tuning of the LQR controller and Kalman filter.** So far the robot did not manage to operate robustly. However, if the above issues are resolved, the LQR controller and Kalman filter need to be tuned.

Chapter 8

Conclusions and recommendations

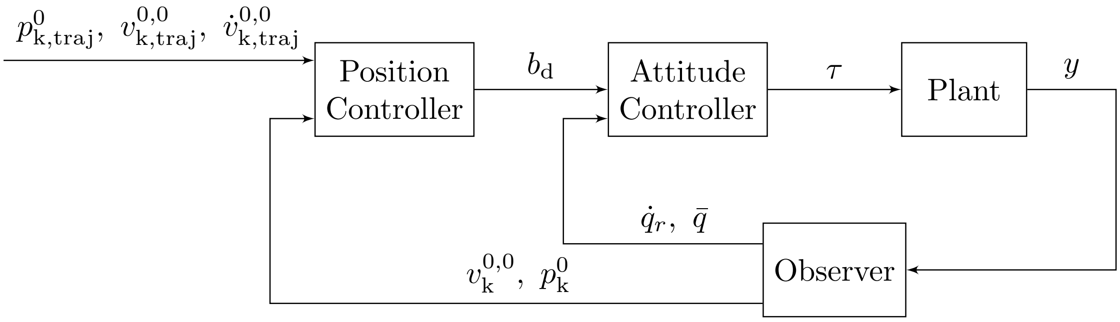


Figure 8-1: Controller structure

This work can roughly be split into a theoretical and practical part. The theoretical part revolved around a geometric, i.e. coordinate-free approach to the control of a ball-balancing robot. To the best knowledge of the author, this approach is novel for ball-balancing robots. Figure 8-1 gives an overview of the total controller structure, consisting of a coordinate-free model, position controller, geometric attitude controller and geometric observer. Furthermore, the control objectives were to control the attitude the control the position of the robot. Furthermore, it was desired to reject disturbances and to be robust with respect to model uncertainties.

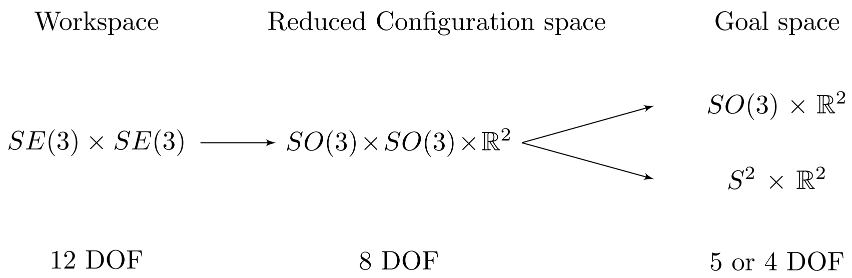


Figure 8-2: Different spaces of the ball-balancing robot

In Chapter 4 a geometric model was derived using screw theory. The workspace of the robot is given by $SE(3) \times SE(3)$. The resulting model is expressed on the configuration space $SO(3) \times SO(3) \times \mathbb{R}^2$, while the dynamics can be fully described on the reduced configuration space $SO(3) \times SO(3)$. The goal space resulting from the control objectives differs from the configuration space and is either on $SO(3) \times \mathbb{R}^2$ or $S^2 \times \mathbb{R}^2$, depending whether it is desired to track a certain heading direction. The change in spaces is shown in Figure 8-2.

The system is under-actuated, i.e. the degrees of freedom of the system is larger than the amount of actuators. For the considered system with the 3 omni-wheel actuation there are 3 actuators and 6 degrees of freedom. Hence the overall system is under-actuated. However, either the attitude $SO(3)$ of the body or the attitude of the sphere $SO(3)$ can be considered fully actuated.

To control the attitude of the body, two methodologies were proposed, a computed torque controller and a sliding mode controller. For both controller two types of attitude tracking were distinguished, i.e. tracking on S^2 or $SO(3)$. The control laws were designed using the special nonlinear error functions for the attitude on either of the two manifolds.

The asymptotic stability of the proposed controllers was proven using Lyapunov theory, assuming bounds on the initial conditions and the domain of operation. The domain of operation of all controllers was defined to be the upper halve of a sphere, i.e. $b_{3,\mathrm{z}} \geq 0$. For the geometric controller a bound on the initial attitude and angular velocity was imposed, while for the sliding mode controller only a bound on the initial attitude was imposed. Finally, the set of viable initial attitudes was found to be larger for tracking on S^2 than for $SO(3)$.

The effectiveness of the proposed controllers was shown in simulation. The differences between tracking on S^2 and $SO(3)$ was shown, being that the controller for S^2 recovers the system with a rotation around one axis, while the trajectory for controller on $SO(3)$ also consists with an additional rotation to correct for the desired heading direction. This results in a straight position trajectory for tracking on S^2 and a curved position trajectory for tracking on $SO(3)$.

The performance of the computed torque controller and sliding mode controller in the nominal case are very similar. However, the is most apparent when the system is subjected to model uncertainties and/or unmodelled friction. It was shown that the computed torque controller was not robust to both types of uncertainties. Sliding mode controller on the other hand was able to robustly control the system, provided a high switching gain η. However, the downside of the sliding mode controller is the possibility of chattering, i.e. a high frequency switching around the sliding surface. This could be resolved by replacing the sign function by a saturation function.

Position control of the robot was established using an outer-loop that provided a desired attitude, given a desired position or trajectory. The position controller assumed the body of the robot to be static, i.e. $\dot{\omega}_{\mathrm{b}}^{0,0} \approx \omega_{\mathrm{b}}^{0,0} \approx 0$. In order to satisfy this condition, the controller parameters of the position and attitude controller are chosen relatively low. It was shown that the control algorithm was able to a) stabilise the robot at a certain fixed point, b) control the from point to point and c) was able to recover from pushes to the system.

Exact trajectory tracking could not be achieved. The reasons are due to the under-actuated nature of the system and the assumptions on the dynamics in the position controller.

An almost global geometric observer was proposed that used the same error functions on the differential manifolds as the geometric controllers. The benefit of this observer was argued to be its simplicity and the fact that no parametrisation of the system dynamics was required. It was shown that the observer was able to converge to the measured states. However, the imposed assumptions on the system might be difficult to prove.

The goal of the practical part was to implement the proposed methods on the physical robot at Alten Mechatronics. Before the implementation of the geometric approach, it was chosen to start with the implementation of a discrete time linear approach consisting of an LQR controller and a Kalman filter. However, due software and hardware issues the implementation of the linear algorithm is not yet successfully realised and is left as future work.

8-1 Recommendations for future research

In this section recommendation for future research is done.

First of all, for the theoretical part, the following recommendations are made.

- It was shown that the computed torque controller was not robust to parametric uncertainties. This could be improved by introducing an adaptive control law which estimates the uncertain parameters.
- The proposed implementation of the position controller assumed slow system dynamics. It would be interesting to include the full dynamics in the trajectory planner to significantly improve the performance.
- The control structure of ball-balancing robots are often two-loop structures. A possible research area would be to design one geometric controller to control both position and attitude.
- The overall stability of the position and attitude controller was not proven. The investigation of the overall stability could result in bounds on the controller parameters, which would decrease the need of manual tuning of the controller parameters.
- The assumptions on the geometric observer design can be hard to prove. It would be interesting to investigate relaxations on the used assumptions.
- The proposed geometric observer did not account for measurement noise. For future research it could be interesting to improve the observer such that the observer gains are variable and result in a minimum variance unbiased estimate of the states.

Regarding the implementation, the following recommendations are made:

- The omni-wheels of robot consists of two rows. As the wheel rolls over the ball, the contact point of the omni-wheel switches between the two rows. Due to imperfections in the omni-wheels, this results in a vibration of the system. This vibration could be countered through the use of a filter, such as for example a Kalman filter. However, the frequency of the vibration is depended on the angular velocity of the wheels, which is constantly changing. Furthermore, there are three omni-wheels, all rotating at a different velocity and hence inducing different vibrations. This results in a challenging filtering problem. The performance of the Kalman filter could be improved by replacing the omni-wheels with single row omni-wheels, which are for example used in Rezero [5] and BallIp [2].

- The current structure of the embedded systems is not robust. As discussed before, the connection between the boards can fail and/or corrupt data. This could be improved by replacing the current embedded hardware by one master board and sensor module with logging directly on the master board. Such a master board could be a Raspberry Pi. By eliminating the communication line between master board and logging board, the change of missing or corrupted data is reduced. Furthermore, by replacing the current master board by a Raspberry Pi would increase the processing power substantially.

Due to time constraints it was not possible to verify the geometric controller and observer on an experimental platform. This is left as future work.

Bibliography

[1] U. Nagarajan, G. Kantor, and R. Hollis, "The ballbot: An omnidirectional balancing mobile robot," *The International Journal of Robotics Research*, p. 0278364913509126, 2013.

[2] M. Kumaga and T. Ochiai, "Development of a robot balanced on a ball: Application of passive motion to transport," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 4106–4111, IEEE, 2009.

[3] L. Hertig, D. Schindler, M. Bloesch, C. D. Remy, and R. Siegwart, "Unified state estimation for a ballbot," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 2471–2476, IEEE, 2013.

[4] M. Spong, *Underactuated mechanical systems*, vol. 230 of *Lecture Notes in Control and Information Sciences*, book section 9, pp. 135–150. Springer Berlin Heidelberg, 1998.

[5] P. Fankhauser, *Modeling and Control of a Ballbot.* Thesis, Eidgenössische Technische Hochschule Zürich, 2010.

[6] A. N. Inal, O. Morgul, U. Saranli, Ieee, and J. Robotics Society of, *A 3D Dynamic Model of a Spherical Wheeled Self-Balancing Robot*, pp. 5381–5386. IEEE International Conference on Intelligent Robots and Systems, New York: Ieee, 2012.

[7] K. Van der Blonk, *Modeling and Control of a Ball-Balancing robot.* M.s. thesis, University of Twente, 2014.

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

[9] C. K. Chan, C. C. Tsai, and Ieee, "Direct adaptive recurrent interval type 2 fuzzy neural networks control using for a ball robot with a four-motor inverse-mouse

ball drive," in *2013 International Conference on Advanced Robotics and Intelligent Systems*, (New York), pp. 5–10, IEEE, 2013.

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

[11] U. Nagarajan and R. Hollis, "Shape space planner for shape-accelerated balancing mobile robots," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1323–1341, 2013.

[12] M. Kumagai and T. Ochiai, "Development of a robot balancing on a ball," in *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, pp. 433–438, IEEE, 2008.

[13] L. Ching-Wen, T. Ching-Chih, L. Yi-Yu, and C. Cheng-Kai, "Dynamic modeling and sliding-mode control of a ball robot with inverse mouse-ball drive," in *SICE Annual Conference, 2008*, pp. 2951–2955, 2008.

[14] C.-C. Tsai, M.-H. Juang, C.-K. Chan, C.-W. Liao, and S.-J. Chan, "Self-balancing and position control using multi-loop approach for ball robots," in *System Science and Engineering (ICSSE), 2010 International Conference on*, pp. 251–256, IEEE, 2010.

[15] C.-K. Chan and C.-C. Tsai, "Intelligent backstepping sliding-mode control using recurrent interval type 2 fuzzy neural networks for a ball robot with a four-motor inverse-mouse ball drive," in *SICE Annual Conference (SICE), 2012 Proceedings of*, pp. 1281–1286, IEEE, 2012.

[16] C. Cheng-Kai and T. Ching-Chih, "Intelligent backstepping sliding-mode control using recurrent interval type 2 fuzzy neural networks for a ball-riding robot," in *Fuzzy Theory and it's Applications (iFUZZY), 2012 International Conference on*, pp. 169–174, 2012.

[17] A. Lotfiani, M. Keshmiri, and M. Danesh, "Dynamic analysis and control synthesis of a spherical wheeled robot (ballbot)," in *Robotics and Mechatronics (ICRoM), 2013 First RSI/ISM International Conference on*, pp. 481–486, IEEE, 2013.

[18] W. Wang, J. Yi, D. Zhao, and D. Liu, "Design of a stable sliding-mode controller for a class of second-order underactuated systems," in *IEE Proceedings: Control Theory and Applications*, vol. 151, pp. 683–690, 2004. Export Date: 14 November 2014.

[19] Y.-F. Peng, C.-H. Chiu, W.-R. Tsai, and M.-H. Chou, "Design of an omnidirectional spherical robot: Using fuzzy control," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, pp. 18–20, 2009.

[20] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation.* CRC press, 1994.

[21] H. K. Khalil and J. Grizzle, *Nonlinear systems*, vol. 3. Prentice hall New Jersey, 1996.

[22] F. Bullo, *Geometric control of mechanical systems*, vol. 49. Springer Science & Business Media, 2005.

[23] R. Ortega and E. Garcia-Canseco, "Interconnection and damping assignment passivity-based control: A survey," *European Journal of control*, vol. 10, no. 5, pp. 432–450, 2004.

[24] F. Bullo and R. M. Murray, "Tracking for fully actuated mechanical systems: a geometric framework," *Automatica*, vol. 35, no. 1, pp. 17–34, 1999.

[25] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 5420–5425, 2010.

[26] F. Grasser, A. D'Arrigo, S. Colombi, and A. C. Rufer, "Joe: A mobile, inverted pendulum," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, pp. 107–114, 2002.

[27] H. G. Nguyen, J. Morrell, K. Mullens, A. Burmeister, S. Miles, N. Farrington, K. Thomas, and D. W. Gage, "Segway robotic mobility platform," in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5609, pp. 207–220, 2004.

[28] R. Mahony, T. Hamel, and J. M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[29] M. Figueiroa, A. Moutinho, and J. Azinheira, "Attitude estimation in so(3): A comparative case study," in *Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on*, pp. 79–84, May 2014.

[30] H. Khalil, *Nonlinear Control*. Always Learning, Pearson Education, Limited, 2014.

[31] N. Chaturvedi and N. McClamroch, "Attitude stabilization of the inverted 3d pendulum on tso(3) with control saturation," in *Decision and Control, 2007 46th IEEE Conference on*, pp. 1910–1915, Dec 2007.

Glossary

List of Acronyms

CMU	Carnegie Mellon University
TGU	Tohoku Gakuin University
ETH	Eidgenössische Technische Hochshule Zürich
NCHU	National Chung Hsing University
ZOH	Zero-order hold

List of Symbols

$\bar{\cdot}$	Flatten operator: $\mathbb{R}^{n\times m} \rightarrow \mathbb{R}^{nm\times 1}$
$\hat{\cdot}$	Estimated variable
$(\cdot)^\vee$	Inverse tilde operator
$\tilde{\cdot}$	Tilde operator
$Ad_{H_{\mathrm{a}}^{\mathrm{b}}}$	Adjoint matrix for coordinate transformations a twist in frame Ψ_{a} to frame Ψ_{b}
b_3	Last column of the body rotation matrix R_{b}^0
$C(q, \dot{q})$	Coriolis matrix
D_c	Coulomb friction constant
D_v	Viscous friction constant
$D^{\mathrm{a,b}}$	Total friction acting on body a, expressed in frame b
e_R	Attitude error on the special orthogonal group $SO(3)$
e_{b_3}	Attitude error function on the 2-sphere group S^2
e_φ	Error of the travelled encoder angle
e_ω	Angular velocity error
e_ω^{a}	Angular velocity error expressed in frame Ψ_{a}
F_{ext}	External forces and torques acting on the system

$f^{\mathrm{b,a}}$	the forces acting on frame Ψ_{a} expressed in frame Ψ_{b}
$G(q)$	Gravity matrix
g	Gravitational acceleration [m/s^2]
$H_{\mathrm{a}}^{\mathrm{b}}$	Homogeneous matrix from frame Ψ_{a} to Ψ_{b}
$\mathcal{I}^{\mathrm{b,a}}$	Inertia tensor of Ψ_{a} expressed in frame Ψ_{b}
J	Geometric Jacobian
J_{i}	Moment of inertia matrix of body i
$j_{\mathrm{s},i}$	Moment of inertia around the i axis [kg $\cdot$ m^2]
$\mathcal{L}$	Lagrangian
$M(q)$	Inertia matrix i
m_{b}	Mass of the body [kg]
$\tau^{\mathrm{b,a}}$	The torques acting on frame Ψ_{a} expressed in frame Ψ_{b}
m_{s}	Mass of the ball [kg]
Ψ_{a}	Frame a
P^{i}	A three dimensional point expressed in frame Ψ_{i} augmented with 1. $P^{\mathrm{i}} \in \mathbb{R}^4$
$\Phi(R_{\mathrm{a}}^{\mathrm{b}})$	Map relating the flattened derivative of the rotation matrix to the angular velocity
φ	Encoder angle
$p_{\mathrm{a}}^{\mathrm{b}}$	Translation from frame Ψ_{a} to Ψ_{b}
R_e	Error rotation matrix
$R_{\mathrm{a}}^{\mathrm{b}}$	Rotation matrix from frame Ψ_{a} to Ψ_{b}
r	Vector between the center of the sphere and the contact point with the floor
r_s	Radius of the ball [m]
r_w	Radius of the omniwheels [m]
$SE(3)$	Euclidean group
$SO(3)$	Special orthogonal group
S^2	2-sphere manifold
θ	Euler angle
$T_{\mathrm{a}}^{\mathrm{c,b}}$	Twist from Ψ_{a} with respect to Ψ_{b} expressed in Ψ_{c}
τ	Motor torque
V	Lyapunov function
$v_{\mathrm{a}}^{\mathrm{c,b}}$	Linear velocity from Ψ_{a} with respect to Ψ_{b} expressed in Ψ_{c}
v_a	Accelerometer measurement $\in \mathbb{R}^3$
v_m	Magnetometer measurement $\in \mathbb{R}^3$
$W^{\mathrm{b,a}}$	Wrench acting on frame Ψ_{a} expressed in frame Ψ_{b}
$\omega_{\mathrm{a}}^{\mathrm{c,b}}$	Angular velocity from Ψ_{a} with respect to Ψ_{b} expressed in Ψ_{c}