Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics

Bachelor's Thesis

Extension of the Control System for the Magnetic Manipulator with a Non-Flat Surface

Author: **Jan Filip**
Supervisor: **Ing. Jiří Zemánek**
Academic Year: **2014/2015**

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

BACHELOR PROJECT ASSIGNMENT

Student: Jan F i l i p

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Bachelor Project: Extension of the Control System for the Magnetic Manipulator with a Non-Flat Surface

Guidelines:

This bachelor thesis is expected to augment the existing experimental platform consisting of an array of coils and a steel ball (or possibly several balls) that moves on a flat surface above the coils. The goal of this work is to design and fabricate a "non-flat" surface that would make the task of steering the ball(s) over the surface even more challenging. For this surface, a feedback controller shall be designed that would guarantee some requirements on the motion of the ball(s). The particular subtasks are:

1. Design and fabricate (you can use a 3D printer) some non-flat surface such as a bowl, U-ramp, bumpy surface with bumps above the coils etc. While designing this surface, you should think about how this combines with the magnetic potential landscape to obtain some interesting and challenging combination.
2. Adjust some existing computer vision algorithm for measuring the position of one or several balls. In particular, strive to obtain the best possible timing performance.
3. Create and verify a mathematical model that captures important features of the kinematics and dynamics of the experimental platform.
4. Design and test a control system that can steer the ball (balls) in a desired way by setting the reference values of currents for the coils. For example, maintain an oscillation of a single ball, synchronized oscillations of two balls, plan and follow a reachable trajectory, control the velocity of a ball exiting the area supported by the coils etc. Consider using dedicated hardware such as Speedgoat, or 32-bit ARM Cortex-M4 Discovery kit to implement the control algorithm.

Bibliography/Sources:

[1] Corke P., Robotics, Vision and Control: Fundamental Algorithms in MATLAB, Springer, 2011
[2] Donalt T., Greenwood, Advanced Dynamics, Cambridge University Press, 2006
[3] Simonian A., Feedback control for planar parallel magnetic manipulation, Diploma thesis, CTU in Prague, 2014
[4] Richter F., Úprava instrumentace pro magnetickou platformu Magman, Bachelor thesis, CTU in Prague, 2014

Bachelor Project Supervisor: Ing. Jiří Zemánek

Valid until: the end of the summer semester of academic year 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 20, 2015

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Jan F i l i p

Studijní program: Kybernetika a robotika (bakalářský)

Obor: Robotika

Název tématu: Rozšíření řídicího systému pro magnetickou manipulaci na zakřiveném povrchu

Pokyny pro vypracování:

Tato bakalářská práce by měla rozšířit existující experimentální platformu, která se skládá z pole cívek a ocelové kuličky (případně několika kuliček) pohybující se na rovném povrchu nad cívkami. Cílem této práce je navrhnout a vyrobit zakřivený povrch, který by udělal úlohu manipulace s kuličkou po povrchu ještě náročnější. Pro tento povrch by měl být následně navržen zpětnovazební regulátor tak, aby mohl zajistit dosažení určitých požadavků na pohyb kuličky (kuliček). Jednotlivé úkoly jsou:

1. Navrhnout a vyrobit (můžete využít 3D tiskárnu) nějaký zakřivený povrch jako například misku, U-rampu, hrbolatý povrch s hrboly nad cívkami apod. Při návrhu vezměte v úvahu vliv magnetického pole, abyste dosáhli zajímavých kombinací tvaru povrchu a magnetického pole.
2. Nasaďte a případně upravte nějaký existující algoritmus počítačového vidění pro měření polohy jedné nebo několika kuliček. Pokuste se dosáhnout co nejkratší doby zpracování obrázku.
3. Vytvořte a verifikujte matematicky model, který bude zachycovat důležité kinematické a dynamické rysy experimentální platformy.
4. Navrhněte a otestujte řídicí systém, který bude umět vést kuličku (několik kuliček) požadovaným způsobem pomocí nastavování proudu tekoucího cívkami. Úlohou může být například udržování oscilací kuličky, synchronizované oscilace dvou kuliček, plánování a sledování dosažitelné trajektorie, řízení rychlosti kuličky při opuštění prostoru nad cívkami apod. Zvažte užití dedikovaného hardware jako Speedgoat nebo ARM Cortex-M4 Discovery kit pro implementaci řídicího algoritmu.

Seznam odborné literatury:

[1] Corke P., Robotics, Vision and Control: Fundamental Algorithms in MATLAB, Springer, 2011
[2] Donalt T., Greenwood, Advanced Dynamics, Cambridge University Press, 2006
[3] Simonian A., Feedback control for planar parallel magnetic manipulation, Diploma thesis, CTU in Prague, 2014
[4] Richter F., Úprava instrumentace pro magnetickou platformu Magman, Bachelor thesis, CTU in Prague, 2014

Vedoucí bakalářské práce: Ing. Jiří Zemánek

Platnost zadání: do konce letního semestru 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 20. 1. 2015

Prohlášení

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

V Praze dne _____

_____
Podpis autora práce

Acknowledgement

Foremost, I would like to express gratitude to my supervisor Ing. Jiří Zemánek for his invaluable advices and support. Next, I would like to acknowledge other members of *AA4CC* group for creating a productive work environment. Last but not least, I would like to thank my family for continuous support throughout my studies.

Abstract

This bachelor's thesis is concerned with the augmentation of the magnetic manipulation platform Magman. Namely, it describes the design and fabrication of a non-planar surface to provide a base for motion control of one or multiple steel balls. Firstly, this thesis documents the surface's design requirements and manufacture process based on 3D printing. Secondly, an exact mathematical model for description of ball's motion dynamics is provided based on Lagrange's equation for nonholonomic systems. The mathematical model is accompanied by a detailed analysis of its dynamic behaviour based on simulations and verification by comparison with experiments on the laboratory platform. Proposed derivation of mathematical model may be applied to a ball rolling on planar, spherical or toroidal surfaces. Finally, this work deals with the design of a feedback controller for manipulation with one or multiple steel balls on the non-flat surface. The feedback controller uses vision-based position measurement and steers the ball by shaping surrounding magnetic field. Achieved results of various control strategies on the real laboratory platform are evaluated.

Keywords: 3D print, Lagrange's equation, nonholonomic constraints, nonlinear systems, estimation, multirate digital control, magnetic manipulation, visual servoing.

Abstrakt

Obsahem této bakalářské práce je rozšíření platformy pro magnetickou manipulaci Magman. V úvodu se tato práce zabývá návrhem a výrobou zakřiveného povrchu, který tvoří základnu pro řízení pohybu jedné či více kuliček. V práci jsou zdokumentovány požadavky a postup výroby zakřiveného povrchu pomocí 3D tisku. Následně tato práce popisuje odvození exaktního matematického modelu pro popis dynamiky odvalující se kuličky založeného na Lagrangeově rovnici pro neholonomní systémy. Odvozený matematický model je doplněn detailní analýzou dynamického chování zakládající se na numerické simulaci. Dále je popsáno úspěšné ověření modelu pomocí srovnávacích experimentů na laboratorní platformě. Získaný matematický model lze aplikovat pro popis valivého pohybu míčku na povrchu rovinném, sférickém i toroidním. Na závěr se tato práce zabývá návrhem zpětnovazebního řízení pohybu jednoho i více míčků pro laboratorní platformu se zakřiveným povrchem. Zpětnovazební smyčka využívá měření polohy v obrazu a tvarování magnetického silového pole k manipulaci objektem. V závěru této práce jsou prezentovány výsledky řízení, kterých bylo na laboratorní platformě dosaženo.

Klíčová slova: 3D tisk, Lagrangeova rovnice, neholonomní omezení, nelineární systémy, odhadování, číslicové řízení na více vzorkovacích frekvencích, magnetická manipulace, polohování na základě obrazu.

Contents

1 Introduction

This bachelor's thesis deals with augmentation of an existing laboratory platform which was developed in AA4CC group in the Department of Control Engineering in the Faculty of Electrical Engineering at Czech Technical University in Prague.

The platform consists of an array of coils, which serve to generate magnetic field. A horizontal flat surface is attached above the coils. By activating the coils and shaping the magnetic field, the platform is capable of planar manipulation with ferromagnetic objects on the surface. Currently, motion of one or multiple steel balls can be controlled. The magnetic platform can be classified as an instance of systems for manipulation by shaping force fields. The platform currently serves as a benchmark system for control algorithms in the research of distributed manipulation.

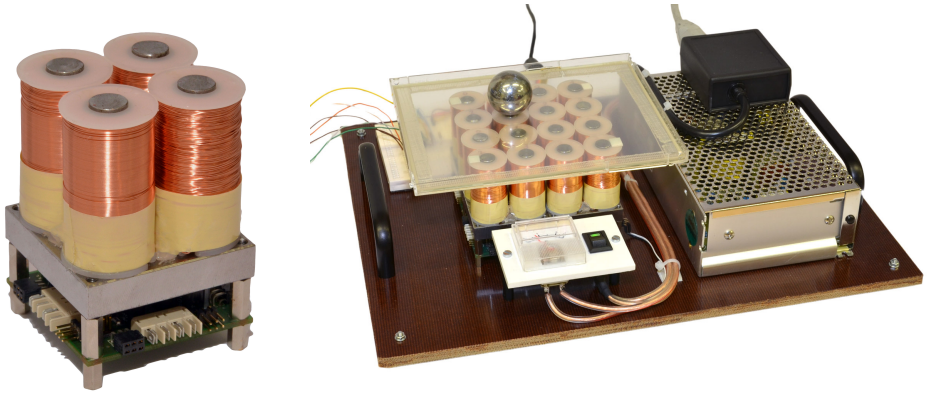
1.1 Motivation

In this bachelor project, we aim to introduce a new concept into the manipulation by shaping force fields. We intend to replace the horizontal plane with a more complex curved surface. By doing so, we will impose a new source of nonlinearity into the already challenging task of manipulation.

However, the main aim of the augmentation is not to further increase the difficulty of ball steering. Our motivation for the augmentation is to examine the effects of added stationary mechanical constraint in conjunction with the manipulation by shaping a force field. By changing the dynamic properties of the mechanical part of the platform, we aim to obtain an interesting combination of behaviour which we will attempt to utilize in order to achieve a higher performance, for example increased velocity of manipulation. Additionally, our motivation is to offer a visually appealing behaviour, such as synchronized oscillations of multiple balls to mimic juggling. Finally, we intend to explore the capabilities of the laboratory platform itself.

1.2 Tasks and outline

In the rest of the first chapter, we will briefly describe the laboratory platform and the mathematical model of its actuators. Next, Chapter 2 describes the first task of this bachelor's thesis, the design and fabrication of a non-flat surface. Chapter 3 is dedicated to the mathematical model of ball's motion dynamics on the designed surface. This chapter provides the derivation of the model, discussion of its behaviour based on simulation results and experimental verification of the mathematical model. Chapter 4 reviews available solutions of position measurement. Subsequently, it describes adjustments of an existing vision-based position measurement in order to be deployed with the non-flat surface. Finally, Chapter 5 describes the development of the control system with the goal of maintaining steady oscillations of a ball on the non-flat surface, including the experimental results achieved on the real platform.



(a) Single module (b) Laboratory platform with a steel ball

Figure 1.1: Photograph of a single module and the laboratory platform consisting of four modules and a horizontal flat surface. Photographs by Ing. Jiří Zemánek.

1.3 Magnetic platform description

In this section, we will provide necessary information concerning the laboratory platform. Based on this description, we will be developing the extension of the platform throughout the subsequent chapter.

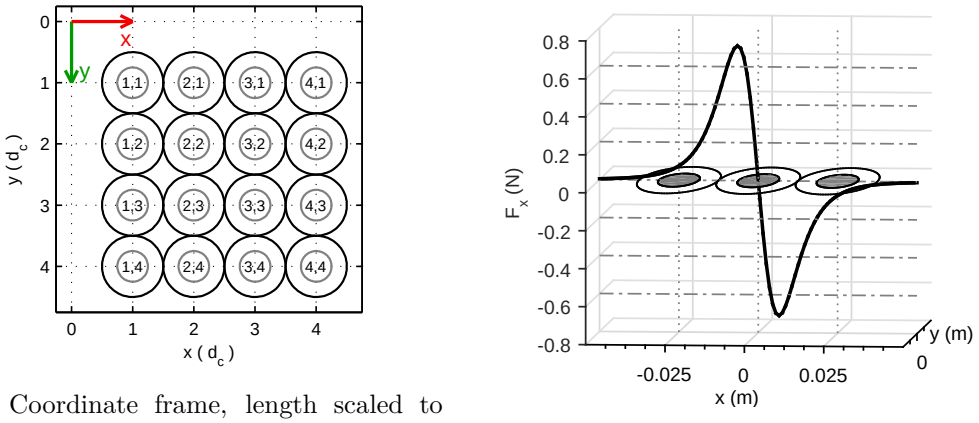
For the sake of simplicity, we can divide the platform into two units: an array of coils and a surface placed above the array. Using the coils, the surrounding magnetic field is shaped, affecting the ferromagnetic object by an attractive force. By setting the current through each coil's winding, the magnetic field can be shaped in order to control the motion of the ferromagnetic object, for example a steel ball.

The platform offers two implementations of position measurement: using a RGB camera, which is observing the whole platform from above, or using a resistive touch foil, which is the top layer of the horizontal planar surface.

The array of coils is assembled from modules, each consisting of four coils with iron cores. From top view, the module is square shaped with dimensions 50×50 mm. Each module is equipped at the bottom with an electronic board including an ARM processor. Electronic circuitry of the module allows for control of the current through each of module's coils using a feedback loop. Single module is depicted in Fig. 1.1a. As each module has its own computing power and ability to communicate with its neighbours, the array unit can be viewed as a network of intelligent actuators. The array is currently assembled from four interconnected modules, which form a 4×4 square array of coils. The laboratory platform is depicted in Fig. 1.1b.

1.4 Mathematical model of actuators

During the design of the curved surface, we shall take into consideration the properties of currently available mathematical model of the coil's magnetic field. Based on its important characteristics, such as maximal force exerted on the ball or maximal achievable velocity of the ball, we intend to design a correctly dimensioned surface. Additionally, we want to ensure that mechanical properties will combine adequately with the magnetic properties of coils according to our requirements. Therefore, let us briefly review the mathematical model of magnetic force created by a single coil, which was presented in [1].



(a) Coordinate frame, length scaled to multiples of coil diameter $d_c = 25$ mm

(b) Force profile of single coil

Figure 1.2: a) coordinate frame with m, n indexes and b) force profile of a fully excited coil according to Eq. (1.4)

The mathematical model has in total 16 inputs which can be written as a 4-by-4 matrix $\mathbf{U}_{m\times n}$. Each element of the matrix represents an amount of excitation of the corresponding coil

$$u(m,n) \in \langle 0,1\rangle, \tag{1.1}$$

where $m, n \in \{1,2,3,4\}$ are indexes of the coil in the matrix, as depicted in Fig. 1.2a. Current i flowing through a coil excited by factor u is scaled by

$$i = u\, i_{\max}, \tag{1.2}$$

where $i_{\max} = 420$ mA is the maximal allowed current. The flow of current through a coil's winding induces the magnetic field. The intensity of the magnetic field is scaled by the value of current approximately according to the scaling function $f(i)$

$$f(i) = \operatorname{sgn}(i)\sqrt{11.1i^2 - 13.6|i|^3}, \quad |i| \le i_{\max}. \tag{1.3}$$

The x, y components of the force exerted on the ball by a single coil placed at the origin of coordinate system can be expressed as

$$\mathbf{F}(i,x,y,z) = \begin{bmatrix} F_x(i,x,y,z) \\ F_y(i,x,y,z) \end{bmatrix} = f^2(i)\frac{c}{(x^2+y^2+z^2)^3}\begin{bmatrix} x \\ y \end{bmatrix}. \tag{1.4}$$

where x, y are the coordinates of the ball's center of mass expressed in the normalized coordinates, meaning that metric lengths are scaled to multiples of coil diameter $d_c = 25$ mm. The top view of the coordinate frame with normalized units is shown in Fig. 1.2a. The force profile of a coil excited by maximal allowed value of current is shown in Fig. 1.2b. Notably, magnetic force has a highly local effect.

The surface is aligned with a xy plane above the coils. Hence, the value of z in this region is constant and approximately equal the height of the ball's center of mass above the coils: $d = 14.5$ mm. Parameter c is aggregate constant which describes material and magnetic properties of the manipulated object, the surrounding and the coil. Values of unknown parameters were acquired by least square fitting of Eq. (1.4) to measurement data. The experiment was conducted using a force gauge. The ball was pushed away from an activated coil and the opposing force was measured.

Although magnetic fields of individual coils overlap and cause mutual interactions, it was experimentally verified in [1] that the interaction is insignificant and

can be neglected. In order to compute the total force exerted at the ball, we sum the contributions of individual coils

$$\begin{aligned} F_x &= \sum_{m=1}^{4}\sum_{n=1}^{4} f^2(i_{m,n}) \frac{-1.85 \cdot 10^{-9}(x-m)}{((x-m)^2+(y-n)^2+d^2)^3}\,, \\ F_y &= \sum_{m=1}^{4}\sum_{n=1}^{4} f^2(i_{m,n}) \frac{-1.85 \cdot 10^{-9}(y-n)}{((x-m)^2+(y-n)^2+d^2)^3}\,, \end{aligned} \tag{1.5}$$

where x, y are the coordinates of the ball's center of mass expressed in the scaled coordinate frame. As seen in Eq. (1.5), the platform has a significant input nonlinearity. In addition, magnetic actuators cannot exert repelling force to the manipulated object. Lastly, the value of the force depends on both the material and the volume of the used ball.

2 | Surface design

This chapter describes design of the surface shape based on properties required for the task of ball's motion control. Presented requirements originate from author's previous work and experiments during the individual project, which is described in [2]. Current requirements were further improved and adjusted to obtain desired properties for the manipulation. Subsequently, this chapter documents the manufacture process of the curved surface, includes both annotated drawings of designed parts and photographs of resulting printed objects.

2.1 Design requirements

Formerly, the surface used in the task of ball servoing was a flat board with a safety bumper frame around its edge. The board was attached horizontally above the array of coils. Magnetic field created by each coil is highly localized and its intensity decreases significantly with the distance. As a result, magnetic force attracting the ball becomes weaker in the absence of coils at the edges of the board. Once the ball leaves the area of actuator's effect, there is no restriction that will force the ball to return. Keeping the ball in the range of actuators used to be one of the tasks of the control algorithm. In the case of its failure, the ball's position may become unreachable by actuators.

Thus, the first requirement is that the new surface will ensure the return of the ball into the actuated area of the platform. In addition, we want to prevent the ball from leaving the platform due to safety reasons. The ball should remain in the designated area to avoid damage of the platform's hardware or any other laboratory equipment. Therefore, we prefer the horizontal cross section of the surface to form a closed curve to avoid the ball falling off sideways. For this particular reason, the shape of an *U-ramp* was abandoned.

Next, we aim to preserve known properties of the coil's magnetic field, whose mathematical model was extensively researched and identified in previous works, namely in [1]. Currently, the model is known only in a horizontal plane above the surface and does not account for the change in the height of the ball. Therefore, one of the proposed shapes, the bumpy surface, was rejected. Bumps were supposed to be locally counteracted by the magnetic potential of activated coils, allowing the ball to travel freely across the bumpy surface. To be able to design such surface, a precise knowledge of the magnetic field in 3D would be necessary. In order to use the mathematical model without loss of precision, the surface should remain planar in the actuated area. The manufactured surface should be placed in approximately same height above the coils as the preceding one. Similarly, the thickness and magnetic properties of the used material should remain unchanged.

In the original state of the platform, the ball lost most of its kinetic energy by colliding with the safety bumper. Since the ball must be maintained in the actuated

area by the controller and any collision with edges of platform results in a loss of velocity, the maximal achievable performance of the system in terms of the velocity is limited. We intend to design the surface to help conserve the energy in the system. The kinetic energy of the ball would be temporarily stored as potential energy due to ascent on the curved parts of the surface, instead of fully dissipating during a collision. The change of surface slope should be sufficiently smooth to avoid bouncing at the transition. Mathematically, we state this requirement so that the function describing the surface is at least of class C^1, meaning that both the function and its derivative is continuous. Besides, we want the ball to be able to move on the curved sides of the platform, possibly around the flat region. Therefore, the horizontal cross section of the surface should form a closed curve without edges, such as an ellipse or a circle.

Lastly, we have to consider the manufacturability of the design. We intend to use a 3D printer (namely Ultimaker 2) for the fabrication. Using a plastic material and custom thickness of horizontal layers, we can easily fulfil the requirement of actuator's model preservation. A particular advantage is that we can achieve a wide variety of shapes with high precision. On the other hand, we have to keep in mind two drawbacks of this manufacturing method. Firstly, as the printed object is built up layer by layer, the print quality decreases with the increasing angle of overhang between consecutive layers (0° being vertical and 90° being horizontal). Therefore, support structures must be incorporated in the design, mainly for fabrication of slightly inclined parts of the surface. Secondly, the available 3D printer has a limited build volume. The final design including support structures must fit into a $(230 \times 225 \times 205)$ mm bounding box.
In conclusion the requirements which the surface design must fulfil are:

- Ensure the return of the controlled object into the actuated area and prevent the escape of the controlled object from the designated area of the platform.
- Preserve properties of the magnetic field created by actuators.
- Improve the conservation of ball's energy.
- Allow tracking of trajectories involving the inclined sides of the surface.
- The design should by manufacturable using available 3D printer.

2.2 Selected shape

In this section, we will describe the shape selected based on the Section 2.1 requirements. We have chosen to use a circular horizontal cross section, which leads to a simpler mathematical description compared to an elliptical cross section. The surface is intended to be centered above the array, meaning that the axis of the surface's rotational symmetry (denoted as z) intersects the center of the square array of coils and the axis is perpendicular to the platform's horizontal plane.

Due to the radial symmetry, the surface can be described by a curve in the ρz radial cross section. The resulting surface is obtained by a full revolution of the curve around the z axis. The radius of the circular flat region was chosen so that its boundary intersects centers of coils in the array's corners. Most of the original actuated area remains unchanged in order to fully utilize available actuators. We will denote the radius of the flat region as R. The flat region holds the equation

$$z(\rho) = 0, \quad \rho \in \langle 0, R \rangle. \tag{2.1}$$

To shape the slopes at the edge of the surface, we have chosen two curves. Either of these could be conveniently described using a mathematical formula. The first is a parabolic segment

$$z = f(\rho) = a(\rho - R)^2, \quad \rho \in \langle R, r_{\text{build}} \rangle, \tag{2.2}$$

where a determines the steepness and r_{build} denotes the radius of the largest circle that fits atop the 3D printer's buildplate. The second curve is composed of a convex segment of a circular arc

$$z = r_{\text{s}} - \sqrt{r_{\text{s}}^2 - (\rho - R)^2}, \quad \rho \in \langle R, r_{\text{build}} \rangle, \tag{2.3}$$

where r_{s} denotes the radius of the circular segment. Both of these curve segments satisfy the requirement to be at least of class C^1 with Eq. (2.1). Each curve has one arbitrary parameter which allows for adjustment of the final shape.

2.3 Platform performance

In this section, we will estimate the platform performance and use it to select the dimensions of the new surface in accordance with the design requirements.

In the previous design in [2], we based the estimation on the maximum velocity v_{max} which the initially motionless ball can gain during one transition over optimally activated array. Only coils contributing positively to the ball's acceleration were activated at every moment of the simulation. Maximum velocity of a ball with mass $m_{\text{b}} = 110\,\text{g}$ and radius $r_{\text{b}} = 15\,\text{mm}$ is

$$v_{\text{max}} = 0.75\,\text{m/s}. \tag{2.4}$$

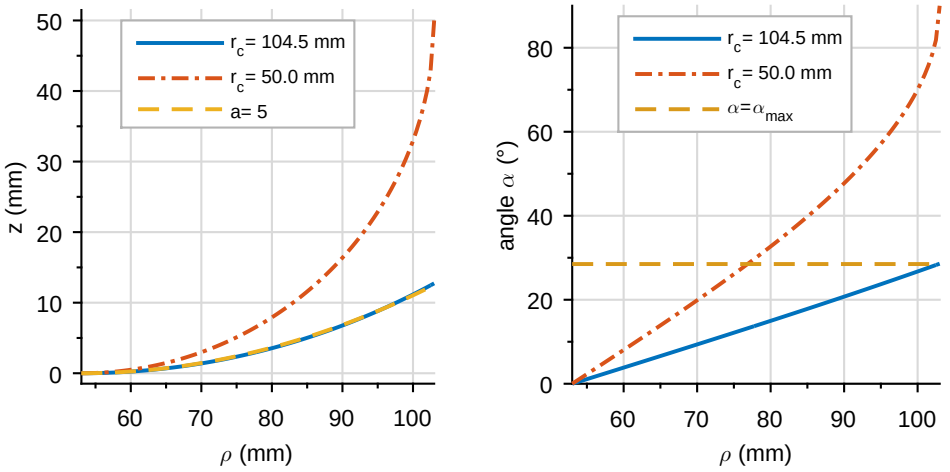
Using the maximal velocity, we evaluated the upper bound of the height reachable during one transition. Assuming that the total kinetic energy of the ball transforms into potential energy

$$m_{\text{b}} g h_{\text{max}} \leq \frac{1}{2}\mathcal{I} v_{\text{max}}^2 \qquad \rightarrow \qquad h_{\text{max}} \leq \frac{\mathcal{I}}{2 m_{\text{b}} g}\, v_{\text{max}}^2 = 40.0\,\text{mm}, \tag{2.5}$$

where $\mathcal{I} = \frac{7}{5}\, m_{\text{b}}$ is generalized inertance, the result of merging the rotational and translational inertia of the ball. Based on this estimate, it seemed that the ball will be likely to escape the surface if the height of slopes is insufficient. Therefore, the first surface was composed of a circular segment with $r_{\text{s}} = 50\,\text{mm}$, which allows for an increase in the height of ball's center $h_{\text{max}} = r_{\text{c}} - r_{\text{b}} = 35\,\text{mm}$ and whose slope angle varies from $0°$ up to $90°$, preventing the escape of the ball.

However, during experiments with the first manufactured surface, one of the previous assumptions appeared to be violated. Namely, the ball was loosing its rolling contact with the surface for angles of surface inclination greater than approximately $30°$. As a result, the ball was sliding off the curved parts, resulting in a decrease of velocity. Moreover, sliding prevented the ball from reaching any significant height and made the curved parts unsuitable for ball's trajectories.

According to these result, we altered the requirements. The surface edges do not necessarily need to be vertical to prevent the escape of the ball. Optionally, a vertical rim may be added around the edge of the surface as a safety measure. Nevertheless, the inclination of the slopes *must* be decreased below $\alpha_{\text{max}} \doteq 30°$.



(a) surface height (b) surface angle α

Figure 2.1: Comparison of the first and current surface design

As previously, we choose the circular arc segment due to its preferable mathematical properties, which will be described further in Section 3.2. The radius r_s will be chosen according to the surface angle $\alpha(\rho)$ which is given as

$$\tan(\alpha) = \frac{\partial z}{\partial \rho} = \frac{\rho - R}{\sqrt{r_\mathrm{s}^2 - (\rho - R)^2}}. \tag{2.6}$$

We choose the maximal angle to be $\alpha_\mathrm{max} = 28° 30'$ at the edge of the surface with $r_\mathrm{build} = 103\,\mathrm{mm}$. Solving Eq. (2.6) for r_s with $\alpha := \alpha_\mathrm{max}$ and $\rho := r_\mathrm{build}$, we obtain

$$r_\mathrm{s} = \sqrt{\frac{(\rho - R)^2}{\tan^2(\alpha)} + (\rho - R)^2} = (\rho - R)\csc(\alpha) \doteq 104.5\,\mathrm{mm}, \tag{2.7}$$

where $\csc(\alpha)$ is the cosecant of α. Similarly, the surface radius can be obtained based on the maximal permitted velocity v_max in the flat region by solving

$$(\rho - R)^2 + (z - r_\mathrm{s})^2 - r_\mathrm{s}^2 = 0, \tag{2.8}$$

where we substitute $\rho := r_\mathrm{build}$ and $z := h_\mathrm{max} = \frac{7}{10g} v_\mathrm{max}^2$. The radius is given as

$$r_\mathrm{s} = \frac{\rho^2 - 2\rho R + R^2 + z^2}{2z}. \tag{2.9}$$

The maximal allowed velocity on the surface with $r_\mathrm{s} = 104.5\,\mathrm{mm}$ is

$$v_\mathrm{max} = \sqrt{\frac{10g}{7}\left(r_\mathrm{s} - \sqrt{r_\mathrm{s}^2 - (r_\mathrm{build} - R)^2}\right)} \doteq 42.2\,\mathrm{cm\,s^{-1}}. \tag{2.10}$$

Comparison of the previous and current design is shown in Fig. 2.1. As seen in Fig. 2.1a, we can achieve almost the same shapes using either a parabola or a circular segment for certain values of a and r_s. Fig. 2.1b shows the angle of surface's inclination in dependency on ρ. The preceding surface reaches α_max for $\rho \approx 77\,\mathrm{mm}$, where its height is still rather insignificant. On the other hand, current surface design will allow us to fully utilize the build volume with surface slopes in permitted bounds. Therefore, considerably greater area will be suitable for the rolling motion of the ball, which will be able to reach a greater height prior to sliding.

Table 2.1: Surface dimensions and properties

Description	Label	Value
Maximal angle of slope	$\alpha_{\max}$	$28° 30'$
Radius of flat region	R	53 mm
Radius of curved side	r_s	104.5 mm
Maximal allowed velocity	$v_{\max}$	42.2 cm/s

The final surface can be equivalently described as if composed by merging of:

1. Circular surface with radius R, centered at the origin of coordinate system.
2. Section of toroidal surface with the major radius R and the minor radius r_s, centered in height r_s above the origin.

Dimensions and properties of the final design are summarized in Tab. 2.1.

2.4 Fabrication

In this section, the fabrication of the surface is documented. Namely, student license of AutoCAD 2014[1] was used to produce 3D models. Subsequently, software MeshMixer[2] was applied to generate customizable support structures for the printing of the surface. Finally, software Cura[3] was used for processing of 3D models into printing instructions, which are more commonly referred to as `gcode`.

2.4.1 Surface

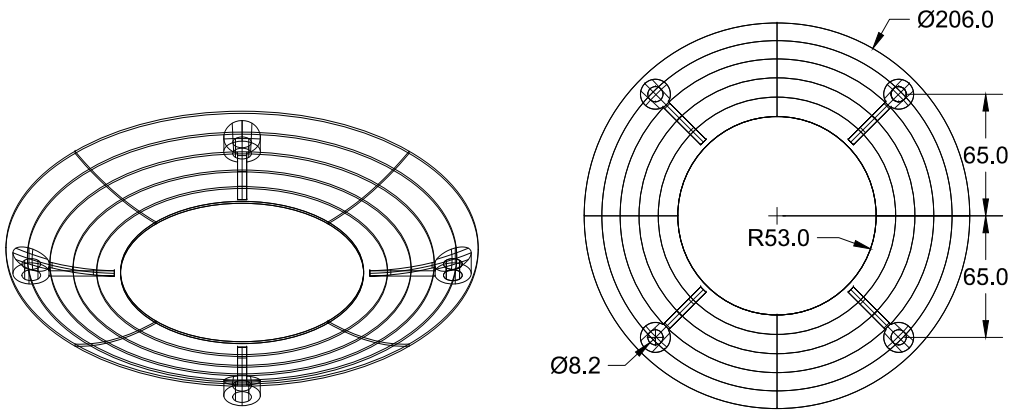
To produce the surface in AutoCAD, we started by drawing the curve in radial cross section such as in Fig. 2.1a. Using `Revolve` command, the curve was turned into a 2D surface object. To add the third dimension, we used the command `Thicken`. The thickness of walls was set to 1 mm and applied along the outer normal direction so that r_s remains unchanged.

To attach the surface to the platform, four mounting holes were added on the bottom side of the surface. These holes have cylindrical shape with 8.2 mm diameter and 5 mm depth. Centers of mounting holes form a square with a 130 mm side. The drawing of the surface is depicted in Fig. 2.2. Additionally, radius of the cylinder forming the flat region was increased to provide supplementary support for slopes below which the height is insufficient to accommodate generated supports.

Next, we generated support structures using Meshmixer 2.0, which allows to configure both the density and dimensions of support struts. Used configuration was based on recommendations in [3]. Namely, we added support structures for overhang angle greater than 45° with *density* option set to 25. Each individual support strut has following diameters: 7 mm base, 3.2 mm post and 0.55 mm tip.

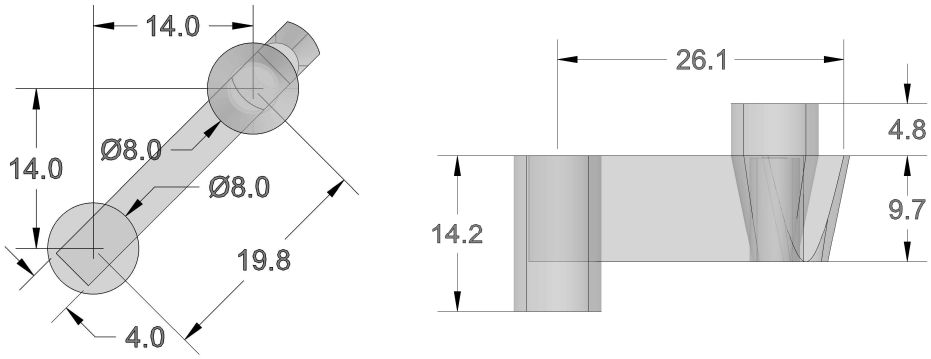
Finally, we used Cura for slicing. Supports generated by MeshMixer are not infilled by default when opened using Cura. To correct this issue, we selected the option combine everything (type A) in the expert settings of Cura. This option ensures that supports are filled completely, including their mutual overlaps. Subsequently, we added a brim to interconnect the supports together and to improve

[1] `http://www.autodesk.com/education/free-software/autocad`
[2] `http://www.meshmixer.com/`
[3] `https://software.ultimaker.com/`



(a) top corner view (b) top view, dimensions in mm

Figure 2.2: Drawing of the surface with added mounting holes



(a) top view (b) side view

Figure 2.3: Drawing of interconnecting part, dimensions in mm

their adhesion to the printer's bed. Additionally, brim prevents the printed object to wrap off the bed during the course of printing and is especially recommended for large flat objects in [4]. Surface was printed using 0.1 mm layer thickness.

2.4.2 Connection structures

According to the location of surface's mounting holes, an interconnecting part was designed. Its purpose is to join the surface with a rack, which was designed during the author's individual project. The rack is firmly mounted on the coil modules and provides mechanical support for the surface to minimize its reactive motion. Connection is created between a pair of opposite mounting holes. Each hole is 8.2 mm in diameter, providing a sufficient tolerance for insertion of a cylindrical peg with 8.0 mm in diameter. Since the mounting holes cannot be placed directly above each other due to lack of space, a pair of pegs is connected by a horizontal link, whose length may be adjusted according to the needs of given surface. The design of interconnecting part is depicted in Fig. 2.3. The connection of the rack and the surface is depicted further in Fig. 2.7.

Figure 2.4: Printed surface prior to removal of support structures

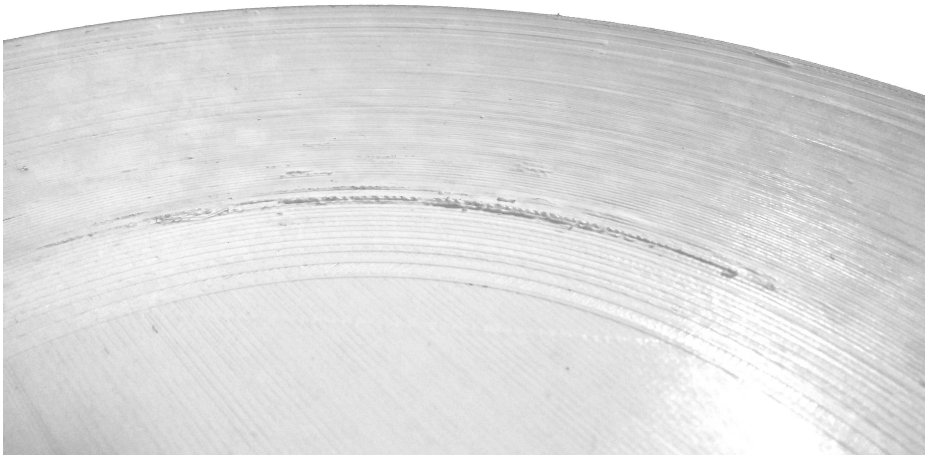


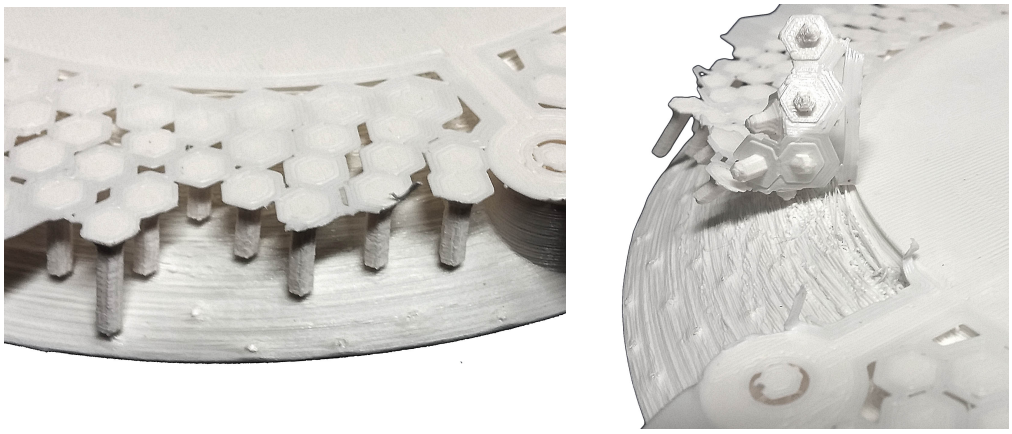
Figure 2.5: Quality of the printed surface with a notable protrusion

2.4.3 Results

Resulting quality of the print was satisfactory, however, two improvements can be made. Firstly, the 1 mm thickness appears rather insufficient for the curved sides. Increasing the thickness would improve the stiffness as well as print quality of the curved sides. Fig. 2.4 shows the quality of the print before removal of support structures. Notably, curved sides appear slightly transparent due to low thickness.

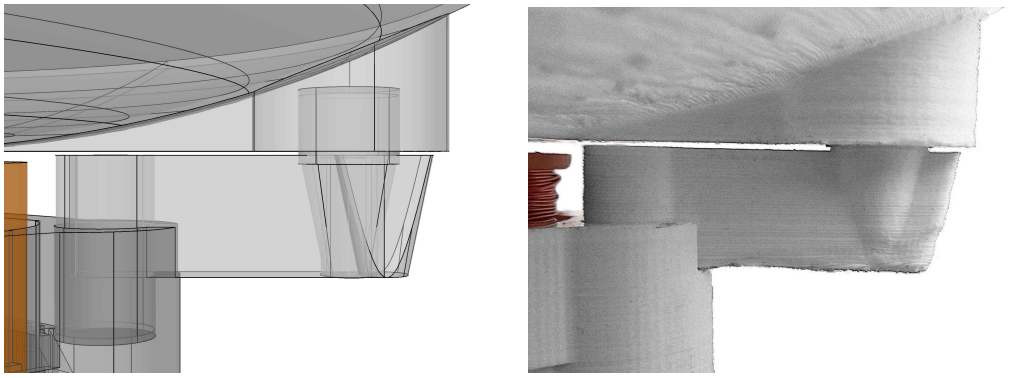
Secondly, the radius of base may be further increased to provide more support for first layers of the curved sides. The surface had minor protrusions in places of insufficient support, which is depicted in detail in Fig. 2.5. Nevertheless, we managed to cut off the protrusions with a razor and brush the surface using a brush paper. It was possible to remove aforementioned flaws, however, suggested improvements may prevent them completely in future.

Supports generated by MeshMixer were easy to cut off with pliers, leaving only a small bump at bottom of the surface, which is shown in Fig. 2.6a. The upper side of the surface was unaffected. Finally, Fig. 2.6b shows the cause of protrusions. Note the loosened threads between the flat base and the first row of supports. Increasing the radius of the base will provide additional support below these overhanging layers.



(a) Structure of the supports (b) Smoothness at the bottom

Figure 2.6: Removal of support structures



(a) Drawing (b) Photograph

Figure 2.7: Connection between the surface and the rack

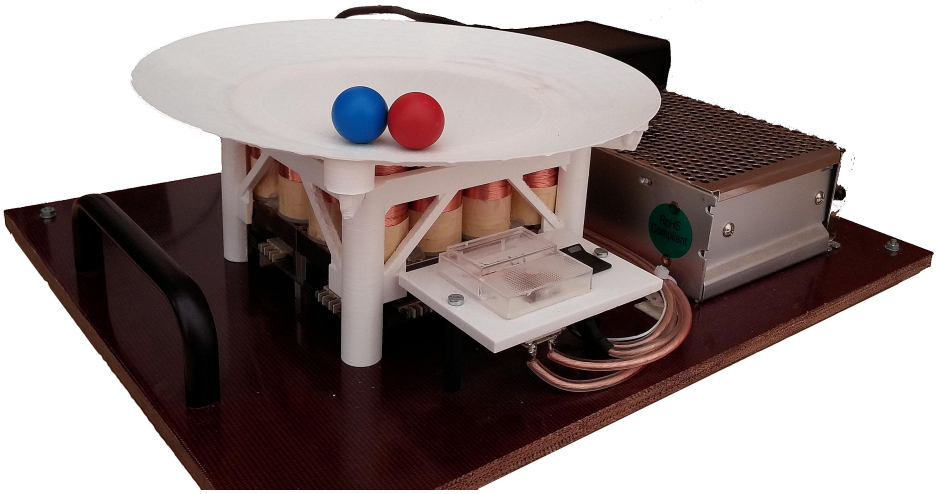


Figure 2.8: Laboratory platform with the curved surface

3 | Mathematical model

In this chapter, we will derive mathematical model of the ball's motion dynamics. The purpose of the mathematical model is numerical simulation and examination of the ball's behaviour on both flat and curved surfaces. We will also use the model as a starting point in the design of a feedback controller, which will be later deployed on the laboratory platform.

Modeled mechanical system consists of a ball and a 3D surface on which the ball can move. In the surface's flat region above the array, the motion of the ball can be influenced by input force as described in Section 1.4. Both the surface and the ball will be described using rigid body mechanics. We consider the ball to be homogeneous, with constant radius r_b and mass m_b. The moment of the ball's inertia with respect to an arbitrary axis going through its center is given as $J_\mathrm{b} = \frac{2}{5}\, m_\mathrm{b} r_\mathrm{b}^2$. We assume that the ball moves on the surface only by rolling without sliding.

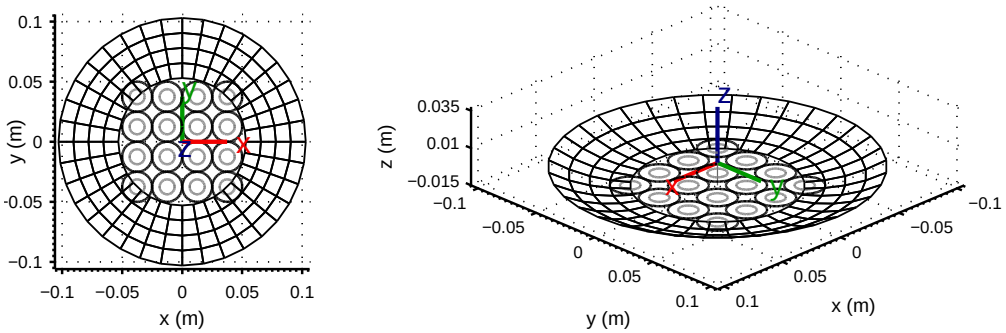
3.1 Pose of the ball

Throughout this section we will use a conventional approach to description of a rigid body in space based mainly on [5]. Additionally, a similar treatment of the topic is provided in other robotic textbooks, most notably in [6] or [7]. First of all, we will choose coordinate frames for the description of the modeled system.

The configuration of a rigid body in three dimensional Euclidean space is fully described by its position and orientation (collectively termed the pose) with respect to a coordinate reference frame. We choose a fixed reference frame with origin O_0 placed in height r_b above the center of the coil array, with z_0 axis pointing upward and the x_0, y_0 axes aligned with array's rows, resp. columns. Fixed frame is chosen so that z coordinate of the ball's center is zero in the flat region. Next, we choose a moving reference frame with origin O_b attached to the center of the ball. Both coordinate frames are right-hand oriented and have orthonormal basis vectors $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0)$, resp. $(\mathbf{x}_\mathrm{b}, \mathbf{y}_\mathrm{b}, \mathbf{z}_\mathrm{b})$. The choice of fixed coordinate frame is depicted in Fig. 3.1. Translational displacement of the ball's center of mass in the fixed frame is equal to the displacement of moving reference frame origin expressed in the fixed reference frame

$$\mathbf{s}_\mathrm{t} = O_\mathrm{b}^0 - O_0^0 = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}, \tag{3.1}$$

where x, y, z denote the components of displacement vector $\mathbf{s}_\mathrm{t}$ along the respective fixed frame axes. The orientation of the ball relative to the fixed coordinate frame can be conveniently denoted by expressing the basis vectors $(\mathbf{x}_\mathrm{b}, \mathbf{y}_\mathrm{b}, \mathbf{z}_\mathrm{b})$ in terms of basis vectors $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0)$. Using a compact notation, this relation is described by a



(a) top view (b) 25° elevated side view

Figure 3.1: Choice of fixed coordinate frame

(3×3) matrix, commonly referred to as rotation matrix

$$\mathbf{R}_{\mathrm{b}}^{0} = \begin{bmatrix} \mathbf{x}_{\mathrm{b}} \cdot \mathbf{x}_0 & \mathbf{y}_{\mathrm{b}} \cdot \mathbf{x}_0 & \mathbf{z}_{\mathrm{b}} \cdot \mathbf{x}_0 \\ \mathbf{x}_{\mathrm{b}} \cdot \mathbf{y}_0 & \mathbf{y}_{\mathrm{b}} \cdot \mathbf{y}_0 & \mathbf{z}_{\mathrm{b}} \cdot \mathbf{y}_0 \\ \mathbf{x}_{\mathrm{b}} \cdot \mathbf{z}_0 & \mathbf{y}_{\mathrm{b}} \cdot \mathbf{z}_0 & \mathbf{z}_{\mathrm{b}} \cdot \mathbf{z}_0 \end{bmatrix}. \tag{3.2}$$

Due to the unit length of basis vectors, the dot product of any two basis vector is the cosine of the angle between them. Therefore, we can refer to components of the matrix as direction cosines. Rotation matrices consist of nine elements which give a redundant description of frame orientation. Columns of a rotation matrix are mutually orthonormal, providing six additional relationships between nine components.

Therefore, only three parameters are required to define the orientation of the body in space. One of the possible minimal representations of orientation is using a set of three Euler angles

$$\boldsymbol{\phi} = \begin{bmatrix} \varphi & \theta & \psi \end{bmatrix}^{\mathsf{T}}, \tag{3.3}$$

which represent an ordered sequence of three consecutive rotations about axes of a moving coordinate frame. In this modeling task, we will use the ZXZ sequence which consists of following elementary rotations:

1. Rotate the reference frame by the angle φ about z axis.
2. Rotate the current frame by the angle θ about x' axis.
3. Rotate the current frame by the angle ψ about z'' axis.

Using elementary rotation matrices

$$\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}, \tag{3.4}$$

the resulting rotation matrix is composed as

$$\mathbf{R}_{zxz}(\boldsymbol{\phi}) = \mathbf{R}_z(\varphi)\mathbf{R}_{x'}(\theta)\mathbf{R}_{z''}(\psi) = \begin{bmatrix} c_\varphi c_\psi - c_\theta s_\varphi s_\psi & -c_\theta c_\psi s_\varphi - c_\varphi s_\psi & s_\theta s_\psi \\ c_\psi s_\varphi + c_\theta c_\varphi s_\psi & c_\theta c_\varphi c_\psi - s_\varphi s_\psi & -c_\varphi s_\theta \\ s_\theta s_\psi & s_\theta c_\psi & c_\theta \end{bmatrix}, \tag{3.5}$$

where s_θ and c_θ are abbreviations of $\sin(\theta)$ and $\cos(\theta)$ respectively, as introduced in [5]. When solving for $\boldsymbol{\phi}$ from a given rotation matrix, this sequence has a singularity. The singularity can be seen if we substitute $\theta_{\mathrm{s}} = 0 + k\pi, k \in \mathbb{N}$ into Eq. (3.5).

In such case, φ and ψ are not defined uniquely. An infinite number of solutions to the problem exists, because only the sum of the first and third angle is constrained by the equation

$$\mathbf{R}_{zxz}(\varphi, \theta_\mathrm{s}, \psi) = \begin{bmatrix} \cos(\varphi+\psi) & -\sin(\varphi+\psi) & 0 \\ \sin(\varphi+\psi) & \cos(\varphi+\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.6}$$

As stated in [7], an Euler angle representation of orientation always exhibits a singularity, regardless of the order of rotations. The singularity is encountered, when the first and last rotation both occur about the same axis. This creates a problem in relating the angular velocity vector of a body to the time derivatives of Euler angles, which are termed Euler rates. Euler rates are true generalized velocities, meaning that upon integration they yield Euler angles and the orientation of the body. Therefore, Euler angles are suitable for use as generalized coordinates in Lagrange's equations. The impact of the singularity on the usability of the model will be further discussed in Section 3.8.

3.2 Independent coordinates

The minimum number of coordinates required to locate a free rigid body in Euclidean space is six. In previous section, we have chosen to describe the position of the ball using the coordinates of its center of mass $\mathbf{s}_\mathrm{t}$ with respect to the fixed frame and the orientation using a ZXZ sequence of Euler angles $\boldsymbol{\phi}$. Combining these together, we obtain the vector of generalized coordinates

$$\mathbf{s} = \begin{bmatrix} x & y & z & \varphi & \theta & \psi \end{bmatrix}^\mathsf{T}, \tag{3.7}$$

which fully specifies the pose of the ball. However, these coordinates are not independent and can be expressed by a smaller set of independent coordinates $\mathbf{q}$.

As the ball can roll only on the physical surface, the position of its center of mass is restricted to a virtual surface of a similar shape above it. This virtual surface is obtained using *surface evolution* by *constant flow*, meaning that the new surface is acquired by moving each point by r_b multiple of the inward-pointing unit normal vector of the physical surface at the given point. Resulting surface has a constant offset r_b from the original one at every point. The same surface can also be obtained by *curvature flow*, which preserves circular shape. These evolutions can be interchanged because of the unique property of both curves of which the surface consists. As stated in [8], straight line and circle are the only two curves with constant *Euclidean curvature*. Therefore, the shape of the curve is preserved upon evolution and the description of the virtual surface can be conveniently obtained in analytical form. This form will be further utilized in the evaluation of the ball's point of contact with the physical surface.

On the contrary, evolving a parabolic curve by constant flow does not result in a parabola and previously mentioned surface evolutions applied to a parabola produce different results. As a result of constant flow evolution, parabolic curve may even collapse, producing a *cusp singularity*. At the point of the singularity, the evolved curve is no longer smooth. Mentioned properties of curves and surfaces are studied in the area of *differential geometry*. Aforementioned book [8] provides not only an introductory description of the topic with applications in image processing, but

also references to purely mathematical books dealing with the differential geometry theory.

The surface on which the center of the ball moves is expressed in the fixed coordinates frame as $z = f(x, y)$

$$f(x,y) = \begin{cases} 0 & \text{if } x^2 + y^2 \in \langle 0, R^2 \rangle, \\ r_\mathrm{e} - \sqrt{r_\mathrm{e}^2 - \left(\sqrt{x^2+y^2} - R\right)^2} & \text{if } x^2 + y^2 \in (R^2, r_\mathrm{build}^2)\,, \\ \text{undefined} & \text{elsewhere,} \end{cases} \tag{3.8}$$

where $r_\mathrm{e} = r_\mathrm{s} - r_\mathrm{b}$ is the effective radius of the virtual surface. Eq. (3.8) represents the relationship between three translational coordinates of $\mathbf{s}$, from which only two are independent. This constraint is *scleronomic*, because it does not depend explicitly on time. Moreover, it is a *geometric* constraint as it depends explicitly only on the position. This constraint can be satisfied by reducing the number of configuration variables by one degree of freedom. Position of the ball's center can be described using following independent coordinates

$$\mathbf{q}_\mathrm{t} = \begin{bmatrix} x & y \end{bmatrix}^\mathsf{T}. \tag{3.9}$$

Joining $\mathbf{q}_\mathrm{t}$ with angular coordinates for minimal representation of the ball's orientation $\mathbf{q}_\mathrm{r} = \boldsymbol{\phi}$, we obtain the vector of generalized independent coordinates

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_\mathrm{t}^\mathsf{T} & \mathbf{q}_\mathrm{r}^\mathsf{T} \end{bmatrix}^\mathsf{T} = \begin{bmatrix} x & y & \varphi & \theta & \psi \end{bmatrix}^\mathsf{T}, \tag{3.10}$$

where indexes t, r denote the translational, resp. rotational, coordinate vectors.

3.3 Nonholonomic constraint

In this section, we will describe the constraint of the ball's motion caused by rolling of the ball. The constraint will be treated using modeling techniques for nonholonomic constrained mechanical systems, which are covered in detail in [9].

Translational velocity of the ball's center of mass with respect to the fixed coordinate frame is given as

$$\mathbf{v}_\mathrm{b}^0 = \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} x & y & f(x,y) \end{bmatrix}^\mathsf{T}. \tag{3.11}$$

We will denote the radial distance of a point on the xy plane from the center of the fixed coordinate frame as $\rho = \sqrt{x^2 + y^2}$. Then, the component of the ball center's velocity along the z axis can be written as

$$\frac{\mathrm{d}f(x,y)}{\mathrm{d}t} = \begin{cases} 0 & \text{if } \rho \in \langle 0, R \rangle, \\ \dfrac{(\rho - R)\,(x\dot{x} + y\dot{y})}{\rho\sqrt{r_\mathrm{e}^2 - (\rho - R)^2}} & \text{if } \rho \in (R, r_\mathrm{build})\,, \\ \text{undefined} & \text{elsewhere.} \end{cases} \tag{3.12}$$

The rolling constraint is described so, that the vector of velocity of ball's center $\mathbf{v}_\mathrm{b}^0$ has the same magnitude but opposite orientation as the vector of the ball's tangential velocity $\mathbf{v}_\mathrm{t}$, which is the result of the ball's rotation. Vector describing the constraint in form $\mathbf{h} = \mathbf{0}$ is

$$\mathbf{h} = \mathbf{v}_\mathrm{b}^0 + \mathbf{v}_\mathrm{t}^0 = \mathbf{v}_\mathrm{b}^0 + \boldsymbol{\omega}_{0\mathrm{b}}^0 \times \mathbf{r}^0, \tag{3.13}$$

where $\boldsymbol{\omega}_{0\mathrm{b}}^{0}$ denotes the angular velocity of the moving frame relative to the fixed frame expressed in the fixed frame and $\mathbf{r}^{0}$ denotes the vector from the center of the ball to its point of contact with the surface expressed in the fixed frame.

To obtain the angular velocity of the body's rotational motion, the equation describing the orthogonality property of the rotation matrix is differentiated with respect to time. Rearranging the result, components of angular velocity $\boldsymbol{\omega}$ can be obtained as respective components of matrix $\mathbf{S}(t)$

$$\mathbf{S}(t) = \frac{\mathrm{d}\mathbf{R}(t)}{\mathrm{d}t}\,\mathbf{R}^{\mathsf{T}}(t) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & -\omega_x & 0 \end{bmatrix}. \tag{3.14}$$

The components of $\boldsymbol{\omega}$ are quasi-velocities which are not integrable, as mentioned in Chapter 4 of [10]. To obtain the orientation as a function of time, angular velocity $\boldsymbol{\omega}$ has to be expressed using Euler rates $\dot{\boldsymbol{\phi}}$, which is done by substituting Eq. (3.5) into Eq. (3.14) and extracting the corresponding components

$$\boldsymbol{\omega}_{0\mathrm{b}}^{0}(\boldsymbol{\phi}, \dot{\boldsymbol{\phi}}) = \begin{bmatrix} \cos(\varphi)\dot{\theta} + \sin(\varphi)\sin(\theta)\dot{\psi} \\ \sin(\varphi)\dot{\theta} - \cos(\varphi)\sin(\theta)\dot{\psi} \\ \dot{\varphi} + \cos(\theta)\dot{\psi} \end{bmatrix} \tag{3.15}$$

The point of contact can be obtained as r_{b} multiple of the outward-oriented unit normal vector at a given point of the surface of the ball center's possible positions

$$\mathbf{r}^{0}(x, y) = r_{\mathrm{b}}\frac{\mathbf{n}}{\|\mathbf{n}\|}, \quad \text{where} \quad \mathbf{n}(x, y) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} & \frac{\partial f(x,y)}{\partial y} & -1 \end{bmatrix}^{\mathsf{T}}, \tag{3.16}$$

Next, we can rewrite the constraint (3.13) in the standard matrix form

$$\mathbf{A}(\mathbf{q}, t)\dot{\mathbf{q}} + \mathbf{B}(\mathbf{q}, t) = \mathbf{0} \in \mathbb{R}^3, \tag{3.17}$$

where each of the three rows corresponds to one constraint. Modeled constraints are scleronomic, *nonholonomic*, because they are not integrable, and *driftless*, because they are linear in velocities and $\mathbf{B}(\mathbf{q}, t) = \mathbf{0}$. Being both driftless and scleronomic, the constraints are said to be *Pfaffian*. Due to the linearity of $\mathbf{h}$ in generalized velocities $\dot{\mathbf{q}}$, we can obtain matrix $\mathbf{A}(\mathbf{q})$ as Jacobian of $\mathbf{h}$ with respect to $\dot{\mathbf{q}}$

$$\mathbf{A}(\mathbf{q}) = \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}} = \begin{bmatrix} \frac{\partial h_1}{\partial x} & \cdots & \frac{\partial h_1}{\partial \psi} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_3}{\partial x} & \cdots & \frac{\partial h_3}{\partial \psi} \end{bmatrix}. \tag{3.18}$$

As $\operatorname{rank}\mathbf{A}(\mathbf{q}) = 2$, we have only two independent nonholonomic constraints. Therefore, in the next steps, we will take only the first two rows of $\mathbf{A}(\mathbf{q})$.

Nonholonomic constraints do not reduce the dimension of the configuration space. It is the the differential motion that is constrained. Any differential displacement $\mathrm{d}\mathbf{q}$ must lie within a three dimensional tangential hyperplane in the configuration space. All admissible velocities are required to belong to the null space of matrix $\mathbf{A}(\mathbf{q})$. Since $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{2\times 5}$, it is possible to find matrix $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{5\times 3}$, which has rank 3 and spans the null space of $\mathbf{A}(\mathbf{q})$ so that

$$\mathbf{A}(\mathbf{q})\mathbf{G}(\mathbf{q}) = \mathbf{0} \in \mathbb{R}^{2\times 3}. \tag{3.19}$$

The columns of $\mathbf{G}(\mathbf{q})$ are linearly independent and form the basis of the null space of $\mathbf{A}(\mathbf{q})$. Comparing Eq. (3.19) with the Pfaffian constraints in the standard form

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \in \mathbb{R}^2, \tag{3.20}$$

it can be seen that the vector of generalized velocities $\dot{\mathbf{q}}$ can be expressed as linear combination of columns of $\mathbf{G}(\mathbf{q})$

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{u}, \tag{3.21}$$

where $\mathbf{u} \in \mathbb{R}^3$ is *kinematic control*, an auxiliary input which ensures that the non-holonomic constraints are met. Since $\mathbf{G}(\mathbf{q})$ forms the basis of the null space, it is not unique and its columns can be combined to obtain different physical interpretations of the kinematic input $\mathbf{u}$.

Given a desired kinematic input $\hat{\mathbf{u}}(\dot{\mathbf{q}})$, whose rows are mutually independent linear combinations of components of $\dot{\mathbf{q}}$, we obtain new matrix $\hat{\mathbf{G}}(\mathbf{q})$ by finding matrix $\mathbf{P}(\mathbf{q})$ such that

$$\dot{\mathbf{q}} = \hat{\mathbf{G}}(\mathbf{q})\hat{\mathbf{u}} = \mathbf{G}(\mathbf{q})\mathbf{P}(\mathbf{q})\hat{\mathbf{u}}. \tag{3.22}$$

Let $\mathbf{O}(\mathbf{q}) = \frac{\partial \hat{\mathbf{u}}}{\partial \dot{\mathbf{q}}}$ be the Jacobian of $\hat{\mathbf{u}}$ with respect to $\dot{\mathbf{q}}$. Then $\mathbf{P}(\mathbf{q})$ is given as

$$\mathbf{P}(\mathbf{q}) = \underbrace{[\mathbf{O}(\mathbf{q})\mathbf{G}(\mathbf{q})]}_{\tilde{\mathbf{G}}(\mathbf{q})}^{-1} \in \mathbb{R}^{3\times 3} \tag{3.23}$$

Each row of $\mathbf{O}(\mathbf{q})$ creates a new row in $\tilde{\mathbf{G}}(\mathbf{q})$ by linearly combining the rows of $\mathbf{G}(\mathbf{q})$. Satisfied that rows of $\mathbf{O}(\mathbf{q})$ are independent, $\tilde{\mathbf{G}}(\mathbf{q})$ is invertible. Postmultiplying the original matrix $\mathbf{G}(\mathbf{q})$ by $\mathbf{P}(\mathbf{q})$, corresponding components in $\hat{\mathbf{G}}(\mathbf{q})$ cancels out, resulting in the desired interpretation of $\hat{\mathbf{u}}$.

Probably the most useful interpretation of kinematic control in our model is by choosing horizontal velocities $\dot{x}, \dot{y}$ and the z component of angular velocity vector $\boldsymbol{\omega}_{0\mathrm{b}}^0$ from Eq. (3.15)

$$\hat{\mathbf{u}} = \begin{bmatrix} x \\ y \\ \dot{\varphi} + \cos(\theta)\dot{\psi} \end{bmatrix}, \qquad \mathbf{O}(\mathbf{q}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cos(\theta) \end{bmatrix}. \tag{3.24}$$

Horizontal velocities can be both measured and influenced by input force. However, ω_z cannot be measured or directly controlled in the current setup. Its influence can be seen after writing the components of the vector product from Eq. (3.13)

$$\boldsymbol{\omega}_{0\mathrm{b}}^0 \times \mathbf{r}^0 = \begin{bmatrix} r_z\omega_y - r_y\omega_z \\ r_x\omega_z - r_z\omega_x \\ r_y\omega_x - r_x\omega_y \end{bmatrix}. \tag{3.25}$$

The trajectory of the ball is affected by ω_z only on non-flat parts of the surface, where r_x and r_y components of $\mathbf{r}^0$ are non-zero.

3.4 Lagrange d'Alambert equation

The equations of motion for the system can be obtained using *Lagrange d'Alambert equation*, which in the case of the modeled system has following form

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{A}(\mathbf{q})^\mathsf{T}\boldsymbol{\lambda} + \mathbf{D}(\mathbf{q})\boldsymbol{\tau}, \tag{3.26}$$

where $\mathcal{L}$ is the *Lagrangian function*, $\boldsymbol{\lambda} \in \mathbb{R}^2$ is vector of Lagrange multipliers and matrix $\mathbf{D}(\mathbf{q}) \in \mathbb{R}^{5\times 2}$ is mapping of generalized input forces $\boldsymbol{\tau}$ to corresponding

generalized coordinates $\mathbf{q}$. Lagrangian $\mathcal{L}$ is given as the difference between kinetic and potential energy expressed in the generalized coordinates

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathcal{U}(\mathbf{q}, t), \tag{3.27}$$

where $\mathcal{T}$ and $\mathcal{U}$ are the total kinetic and potential energy of the mechanism, respectively. In the modeled case, Lagrangian is not explicitly dependent on time. The kinetic energy is given by

$$\mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\, \dot{\mathbf{q}}^{\mathsf{T}} \mathbf{H}(\mathbf{q}) \dot{\mathbf{q}}, \tag{3.28}$$

where $\mathbf{H}(\mathbf{q})$ is the inertia matrix of the system in generalized coordinates. Both translational and rotational inertia matrices are known in dependent coordinates. Next, we need to transform inertia matrices into independent coordinates. As the transformation equations are linear in velocities, it is possible to obtain the transformation matrix as Jacobian of the dependent coordinates with respect to the independent ones. Translational inertia matrix $\mathbf{M}_\mathrm{t}$ in dependent coordinates $\mathbf{s}_\mathrm{t}$ is given as m_b multiple of (3×3) identity matrix. Translational part of kinetic energy is given by

$$\mathcal{T}_\mathrm{t} = \frac{1}{2}\, \mathbf{v}_\mathrm{b}^{\mathsf{T}} \underbrace{\begin{bmatrix} m_\mathrm{b} & 0 & 0 \\ 0 & m_\mathrm{b} & 0 \\ 0 & 0 & m_\mathrm{b} \end{bmatrix}}_{\mathbf{M}_\mathrm{t}} \mathbf{v}_\mathrm{b}. \tag{3.29}$$

The transformation from independent to dependent velocities is described by

$$\mathbf{v}_\mathrm{b} = \mathbf{Q}_\mathrm{t} \dot{\mathbf{q}}_\mathrm{t}, \quad \text{where } \mathbf{Q}_\mathrm{t} = \frac{\partial \mathbf{v}_\mathrm{b}}{\partial \dot{\mathbf{q}}_\mathrm{t}}\,. \tag{3.30}$$

Differentiating Eq. (3.11) according to Eq. (3.30) and substituting into Eq. (3.29), we obtain mass matrix expressed in independent generalized coordinates

$$\mathbf{M}_\mathbf{q} = \mathbf{Q}_\mathrm{t}^{\mathsf{T}} \mathbf{M}_\mathrm{t} \mathbf{Q}_\mathrm{t} \in \mathbb{R}^{2\times 2}. \tag{3.31}$$

Similarly, the tensor of inertia $\mathbf{J}^\mathrm{b}$ with respect to the moving coordinate frame is formed as a J_b multiple of (3×3) identity matrix. Rotational part of kinetic energy is expressed using angular velocity $\boldsymbol{\omega}$ as

$$\mathcal{T}_\mathrm{r} = \frac{1}{2}\, (\boldsymbol{\omega}_{0\mathrm{b}}^0)^{\mathsf{T}} \underbrace{\mathbf{R}_\mathrm{b}^0 \overbrace{(J_\mathrm{b}\mathbf{I})}^{\mathbf{J}^\mathrm{b}} \mathbf{R}_0^\mathrm{b}}_{\mathbf{J}^0} \boldsymbol{\omega}_{0\mathrm{b}}^0. \tag{3.32}$$

Since $\mathbf{J}^\mathrm{b}$ is a scalar matrix and $\mathbf{R}_\mathrm{b}^0 \mathbf{R}_0^\mathrm{b} = \mathbf{I}$, we do not need to consider the change of reference frame orientation in Eq. (3.32). To express the rotational part of kinetic energy in terms of Euler rates $\dot{\boldsymbol{\phi}}$, we use following transformation

$$\boldsymbol{\omega} = \mathbf{Q}_\mathrm{r} \dot{\mathbf{q}}_\mathrm{r}, \quad \text{where } \mathbf{Q}_\mathrm{r} = \frac{\partial \boldsymbol{\omega}(\mathbf{q}_\mathrm{r}, \dot{\mathbf{q}}_\mathrm{r})}{\partial \dot{\mathbf{q}}_\mathrm{r}}\,, \tag{3.33}$$

which is substituted into Eq. (3.32). The matrix of rotational inertia expressed in Euler rates is

$$\mathbf{J}_\mathbf{q} = \mathbf{Q}_\mathrm{r}^{\mathsf{T}} \mathbf{J}^\mathrm{b} \mathbf{Q}_\mathrm{r}. \tag{3.34}$$

The total inertia matrix of the system is expressed in generalized coordinates as

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} \mathbf{M}_\mathbf{q} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_\mathbf{q} \end{bmatrix} \in \mathbb{R}^{5\times 5}. \tag{3.35}$$

Potential energy of the ball $\mathcal{U}(\mathbf{q}_\mathrm{t})$ is given as

$$\mathcal{U}(\mathbf{q}_\mathrm{t}) = m_\mathrm{b} g f(x, y), \tag{3.36}$$

where g is gravitational acceleration. Finally, it is possible to obtain the equations of motion according to Eq. (3.26).

3.5 Reduced order model

The order of the model can be further reduced using the method described in chapter 1 of [9]. The equations derived from Eq. (3.26) written in the compact matrix form are

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{A}(\mathbf{q})^\mathsf{T}\boldsymbol{\lambda} + \mathbf{D}(\mathbf{q})\boldsymbol{\tau}. \tag{3.37}$$

This form is common to most of the mechanical systems and is widely used throughout the field of robotics. Matrix $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})$ can be further rewritten as

$$\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{f}_g(\mathbf{q}), \tag{3.38}$$

where matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ contains Coriolis and centrifugal forces and $\mathbf{f}_g(\mathbf{q})$ is the vector of gravity terms

$$\mathbf{f}_g(\mathbf{q}) = \frac{\partial \mathcal{U}(\mathbf{q})}{\mathbf{q}}\,. \tag{3.39}$$

Elements of matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ are expressed as

$$C_{lj}(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{i=1}^{n} C_{ijk}\dot{q}_i, \tag{3.40}$$

where C_{ijk} are known as *Christoffel symbols* of the first type given as

$$C_{ijk} = \frac{1}{2}\left[\frac{\partial H_{lj}(\mathbf{q})}{\partial q_i} + \frac{\partial H_{li}(\mathbf{q})}{\partial q_j} - \frac{\partial H_{ij}(\mathbf{q})}{\partial q_l}\right] \tag{3.41}$$

where H_{ij} are the elements of matrix $\mathbf{H}$. For more details about Eq. (3.37) please refer to [7].

Based on Eq. (3.21), time derivatives of generalized coordinates can be obtained from known kinematic control. The reduction of order is done by differentiating Eq. (3.21) with respect to time and substituting $\ddot{\mathbf{q}}$ from the result into Eq. (3.37). Next, the differential equations can be once again rearranged to the conventional matrix form

$$\begin{aligned} \bar{\mathbf{M}}(\mathbf{q})\dot{\mathbf{u}} + \bar{\mathbf{N}}(\mathbf{q}, \mathbf{u}) &= \mathbf{G}^\mathsf{T}(\mathbf{q})\mathbf{D}(\mathbf{q})\boldsymbol{\tau}, \\ \dot{\mathbf{q}} &= \mathbf{G}(\mathbf{q})\mathbf{u} \end{aligned} \tag{3.42}$$

where matrices $\bar{\mathbf{M}}$, $\bar{\mathbf{N}}$ were obtained as

$$\begin{aligned} \bar{\mathbf{M}}(\mathbf{q}) &= \mathbf{G}^\mathsf{T}(\mathbf{q})\mathbf{H}(\mathbf{q})\mathbf{G}(\mathbf{q}), \\ \bar{\mathbf{N}}(\mathbf{q}, \mathbf{u}) &= \mathbf{G}^\mathsf{T}(\mathbf{q})\left[\mathbf{H}(\mathbf{q})\dot{\mathbf{G}}^\mathsf{T}(\mathbf{q})\mathbf{u} + \mathbf{N}(\mathbf{q}, \mathbf{G}(\mathbf{q})\mathbf{u})\right]. \end{aligned} \tag{3.43}$$

In the case of horizontal planar surface, Eq. (3.42) can be written in form of eight scalar first order differential equations as

$$\begin{aligned}
\dot{u}_1 &= \frac{5}{7m_\mathrm{b}}\,\tau_x,\\
\dot{u}_2 &= \frac{5}{7m_\mathrm{b}}\,\tau_y,\\
\dot{u}_3 &= 0,\\
\dot{x} &= u_1,\\
\dot{y} &= u_2,\\
\dot{\varphi} &= u_3 + \cot(\varphi)\cos(\theta)\frac{u_1}{r_\mathrm{b}} + \cot(\theta)\sin(\varphi)\frac{u_2}{r_\mathrm{b}}\,,\\
\dot{\theta} &= \sin(\varphi)\frac{u_1}{r_\mathrm{b}} - \cos(\varphi)\frac{u_2}{r_\mathrm{b}}\,,\\
\dot{\psi} &= -\frac{\cos(\varphi)}{\sin(\theta)}\,\frac{u_1}{r_\mathrm{b}} + \frac{\sin(\varphi)}{\sin(\theta)}\,\frac{u_2}{r_\mathrm{b}}\,,
\end{aligned} \tag{3.44}$$

where τ_x and τ_y are generalized forces which were not obtained from potential function $\mathcal{U}(\mathbf{q})$. These generalized forces may account for the input force created by coils as well as dissipative forces due to mechanical friction and *Eddy currents*.

As a result of nonlinearity caused by motion on the curved surface, the equations are rather lengthy in the non-planar case. Therefore, we will handle them in the matrix form of Eq. (3.42) during the implementation of the model for simulation purposes. The inverse of $\bar{\mathbf{M}}(\mathbf{q})$ will be evaluated numerically instead of attempting to obtain it in analytical form. The derivation of mathematical model in the form of a *Wolfram Mathematica* notebook can be found on the attached CD, see Appendix A.

3.6 Numerical simulation

In this section, results of numerical simulation of model behaviour in response to various initial conditions are presented. Additionally, we provide an analysis of the dynamic behaviour based on simulation results.

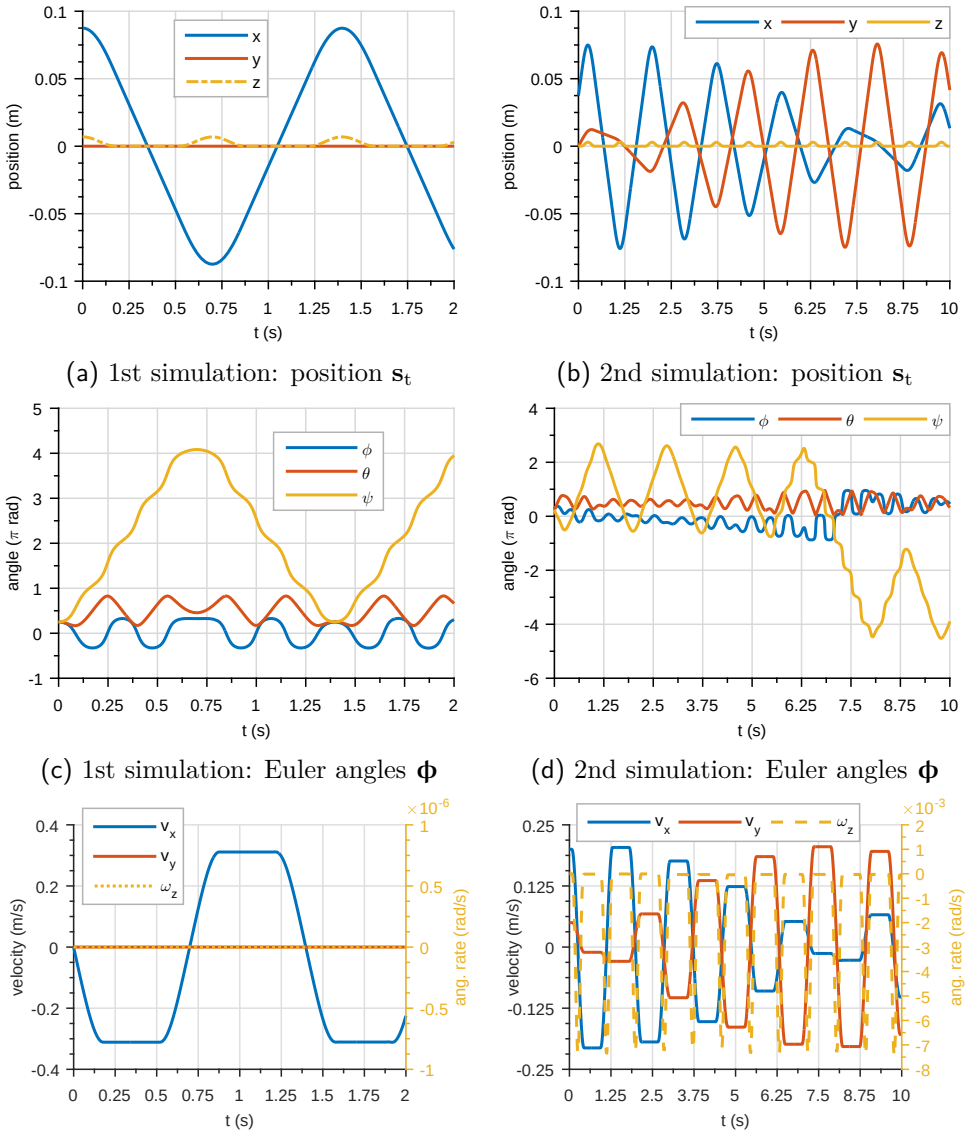
In the first simulation, the ball is placed at the curved side of the surface, with zero initial velocity. Vector of initial conditions $\mathbf{w}_0$ is given as

$$\mathbf{w}_0 = \begin{bmatrix}\mathbf{q}_0^\mathsf{T} & \mathbf{u}_0^\mathsf{T}\end{bmatrix}^\mathsf{T} = \begin{bmatrix}7r_\mathrm{c} & 0 & \frac{\pi}{4} & \frac{\pi}{4} & \frac{\pi}{4} & 0 & 0 & 0\end{bmatrix}^\mathsf{T}. \tag{3.45}$$

After releasing, potential energy of the ball begins to transform into kinetic energy as the height of the ball decreases. The ball travels across the flat part of the surface with constant velocity and reaches again the initial height. Due to radial symmetry of the surface and zero initial velocity, the ball travels directly through the center of the surface, regardless of its initial position on the curved part. Time evolution of the position is depicted in Fig. 3.2a.

During the oscillations along a radial direction, $\omega_z = \omega_{z,0}$ remained zero, as shown in Fig. 3.2e. However, if $\omega_{z0} \neq 0$, the trajectory would have been slightly diverted from the initial direction during each entering of the curved region. As a long term effect, the radial direction of oscillations would slowly rotate with respect to the z axis.

Throughout the movement, θ remained in bounds, as depicted in Fig. 3.2c. Therefore, the results of this simulation can be considered reliable. Note, that the initial value of Euler angles Eq. (3.46) is nonzero in order to avoid the singularity.



(a) 1st simulation: position $\mathbf{s}_t$

(b) 2nd simulation: position $\mathbf{s}_t$

(c) 1st simulation: Euler angles $\boldsymbol{\phi}$

(d) 2nd simulation: Euler angles $\boldsymbol{\phi}$

(e) 1st simulation: kinematic control $\mathbf{u}$

(f) 2nd simulation: kinematic control $\mathbf{u}$

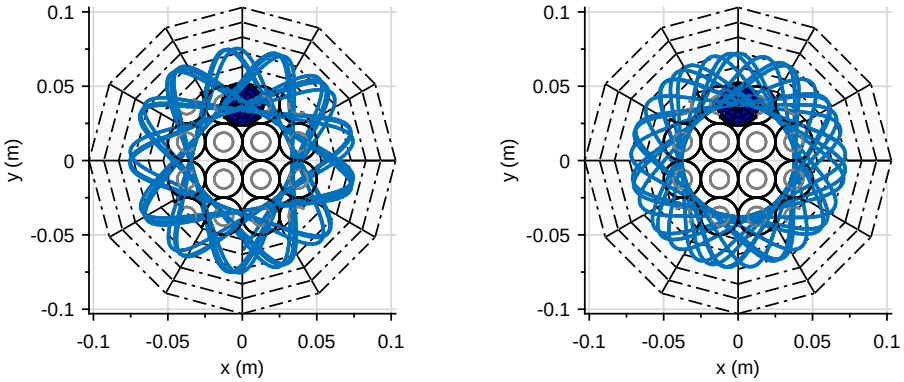
Figure 3.2: Responses to initial conditions. Left resp. right column shows the variables during first resp. second described simulation.

This initial orientation proved to be well chosen to avoid the singularity throughout various simulations. Corresponding figures depict only first two seconds of the simulation, because the behaviour is periodic in this case.

In the second simulation, the ball was placed in the flat region with a nonzero initial velocity. In order to present a characteristic behaviour different from the first simulation, the ball was not placed above the surface's center. If the ball was initially centered, it would periodically oscillate in a constant radial direction, with x, y amplitudes proportional to v_x, v_y, respectively.

Generally, radial oscillations can be obtained, if the ball is placed at a position with respective velocities proportional to its displacement from the center of the surface. Two exceptions from this rule exist, namely the center, where the velocity can be arbitrary but nonzero, and the curved part of the surface, where the velocity can only be zero or in the radial direction.

The second simulation represents the other group of characteristic behaviour, in



(a) $y_0 = 3r_c,\ \dot{\mathbf{q}}_{t,0} = [20\ \ 10]^\mathsf{T}\,\mathrm{cm\,s^{-1}}$ (b) $y_0 = 3r_c,\ \dot{\mathbf{q}}_{t,0} = [20\ \ 0]^\mathsf{T}\,\mathrm{cm\,s^{-1}}$

Figure 3.3: Trajectory in response to initial conditions, simulation of ergodic behaviour of the undamped model for t_{end} =30 s.

which the top view of the trajectory forms a star-shaped curve with various number of vertexes. For the second set of initial conditions

$$\mathbf{w}_0 = \left[3r_c \quad 0 \quad \tfrac{\pi}{4} \quad \tfrac{\pi}{4} \quad \tfrac{\pi}{4} \quad 20\,\mathrm{cm\,s^{-1}} \quad 5\,\mathrm{cm\,s^{-1}} \quad 0\right]^\mathsf{T}, \tag{3.46}$$

it can be seen in Fig. 3.2b that x, y coordinates never cross zero at the same time. Notably, this trajectory shows some periodicity. Similar pattern can be seen in x coordinate near the end of the plotted time as at the beginning in y coordinate. Aforementioned pattern is also apparent in translational velocities in Fig. 3.2f.

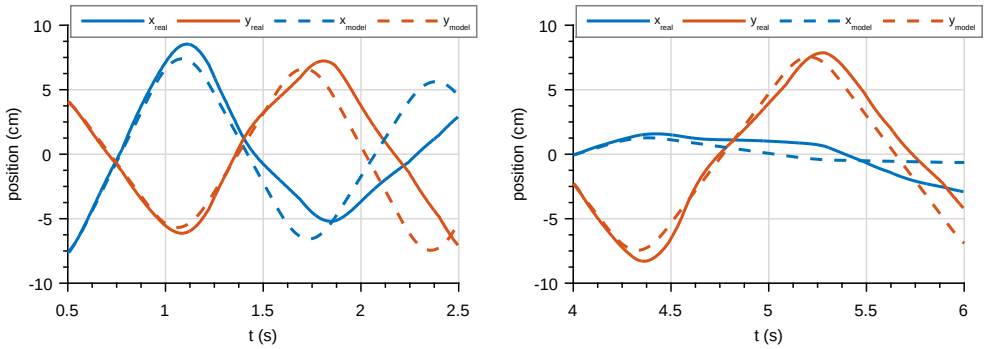
Fig. 3.2d shows the evolution of Euler angles in time. Strictly speaking, θ remained in bounds during the simulation: $\theta \in \langle 0.039;\ 0.958\rangle$. However, as the ball's orientation approached singularity, more rapid changes in Euler angles appeared, most notably at $t \approx 7$ s. Thereafter, the orientation appears to recede from the singularity. The quality of the numerical solution may be decreased. However, the trajectory of the ball's center seem to be unaffected by the fact. The development of ω_z in Fig. 3.2f is similar to the first simulation, meaning that ω_z changes and is nonzero only during motion in the curved area of the surface.

Finally, Fig. 3.3 shows simulation of long term behaviour of the system for the second described type of characteristic behaviour. It can be seen, that there exist a circle around the surface's center, which the ball *never* crosses. The trajectory of the ball forms a star-shaped curve, which in some rare cases may be closed after several transitions. However, in the general case, the trajectory does not close itself completely within several transitions and the whole pattern appears to rotate, which is more notable in Fig. 3.3a.

Nevertheless, the trajectories are closed and periodic, however, the period may be stretched infinitely. The long term behaviour shows a significant sensitivity to slight perturbation of initial conditions, which is one of the characteristics of chaotic systems. The trajectories produced by the the system are for low velocities similar to those of *billiard in a circle*.

3.7 Verification

In this section, we aim to verify the mathematical model by comparing its trajectory with the measured response of the real system to the initial displacement of the ball



(a) 1st comparison (b) 2nd comparison

Figure 3.4: Verification of the mathematical model

on the curved side of the surface. As mentioned in Section 3.6, even small changes in the ball's initial position and velocity are causing the trajectory to divert significantly in the long term view.

Numerous factors contribute to differences between both systems. Namely, the effect of friction was not incorporated to the mathematical model. Additionally, the surface of the real system is neither smooth nor perfectly horizontally aligned. Because of numerous factors contributing to the divergence of trajectories, we would use only short time intervals for comparison. We will set the initial conditions of the model by taking the ball's measured position and estimating the velocity numerically from a few consecutive position measurement samples. By doing so, we can compare the behaviours starting from a random instant of the experiment.

Fig. 3.4 shows the comparisons of the model with the real system in 2 s time ranges. As seen in Fig. 3.4a, behaviours of both systems are almost identical during the first transition over the flat region. The trajectories begin to differ at $t \doteq 1.4$ s where the real trajectory of the ball is turned slightly while crossing the flat region. This fact is most probably caused by the aforementioned nonideal properties of the manufactured surface. Such behaviour occurring on a flat planar surface cannot be easily modeled.

Similarly, divergence of the second trajectory depicted in Fig. 3.4b appears to be mainly caused by a slight but gradual change in the ball's direction. As the change of direction at the curved part of the surface occurs with both slightly different positions and velocities, the difference of trajectories further increases.

In conclusion, meaningful comparison of trajectories is limited to shorter time intervals due to properties of the real system which cannot be modeled. However, the trajectory of the model is close to that of the real system at least during the first second of each presented experimental comparison. Additionally, the behaviour of both system is closely similar during changes of the direction on the curved surface. Therefore, we assume the mathematical model to capture the important dynamic characteristics of the system on both the flat and curved surface. Hence, the model may be used for simulation of the system's short term behaviour with a sufficient precision and for assumptions about the oscillatory behaviour of the real system.

3.8 Singularity of orientation representation

As mentioned in Section 3.1, presented mathematical model suffers from singularity in the representation of orientation. In the singular orientation, Euler angles are

defined ambiguously. This results in rapid changes of Euler rates near singularity. The simulation cannot be started in singular orientation. Moreover, as the ball approaches close to the singularity during the course of the simulation, the quality of the numerical solution is degraded.

Some of the problems may be avoided by changing the initial orientation of the ball. This is possible as the ball is symmetrical, homogeneous and its initial orientation does not influence the trajectory of ball's center. Therefore, the choice of its initial orientation is arbitrary. After the simulation, the range of θ values should be inspected. If the ball orientation approached the singularity significantly, meaning that $\theta \to (0 \text{ or } \pi)$, we should restart the simulation using different initial orientation. Using this approach, it was possible to overcome most of the singularity issues during simulations of responses to various initial conditions. However, as we begin to influence the ball's motion with the input force, the occurrence of singularity becomes more frequent.

3.8.1 Parametrisation using quaternion

To overcome the singularity, we attempted to use a different parametrisation of orientation. Namely the unit *quaternion* $\mathcal{Q} = \{\eta, \boldsymbol{\epsilon}\}$ was used according to the definition in [5]

$$\begin{aligned} \eta &= \cos\left(\frac{\vartheta}{2}\right), \\ \boldsymbol{\epsilon} &= \sin\left(\frac{\vartheta}{2}\right)\mathbf{r}, \end{aligned} \tag{3.47}$$

where η is the scalar part of quaternion, $\boldsymbol{\epsilon}$ is the vector part, ϑ is the angle of rotation and the axis of rotation is specified by vector $\mathbf{r}$. This parametrisation is also called Euler parameters. Unit quaternion is a nonminimal parametrisation of orientation and does not suffer from singularities. However, we attempted to use it to obtain the reduced order model according to [9], which presumes use of independent generalized coordinates. Unit quaternion is parametrized by four parameters, of which only three are independent. The unit norm of quaternion is constrained by the condition

$$\eta^2 + \boldsymbol{\epsilon}^\mathsf{T}\boldsymbol{\epsilon} = 1. \tag{3.48}$$

We used the unit norm constraint to express one of the parameters in terms of the others, namely η as $\eta(\boldsymbol{\epsilon})$, to obtain independent generalized coordinates. We parametrized the rotation matrix using the three independent components of $\boldsymbol{\epsilon}$. Next, the steps of model derivation were relatively similar as when using Euler angles.

However, the performance of the resulting model was compared to the model parametrised by Euler angles. Namely, the ball approached singularity after revolving by angle $\vartheta = \pi$ from the initial position. The singularity was caused by division by zero due to $\eta = 0$ and resulted in an inevitable termination of the simulation.

To conclude, the singularity of orientation representation is inherent when using a minimal representation of orientation. It may be completely overcome only by using a nonminimal representation, however, then the generalized coordinates are not independent and, therefore, not usable with the steps described in [9]. Possibly, this issue may be solved using the Lagrange's equation with dependent coordinates and nonholonomic constraint as described on page 76 in [10] in future. However, we assess the current model parametrised by Euler angles to produce sufficiently reliable results in terms of the trajectory of the ball's center. Its singularity does not significantly restrict the overall usability. Therefore, we decided not to further

pursue the derivation of mathematical model using unit quaternion, which may only be successful if a nonminimal representation with all four parameters is used.

3.9 Simplified model

Let us examine the Eq. (3.44) of ball's motion in the flat region of the surface. It can be seen, that ω_z does not influence the trajectory of the ball's center in the flat region. Moreover, ω_z remains constant in time. If initially $\omega_z(t_0) = 0$, then the kinetic energy of the ball in the flat region can be expressed as

$$\mathcal{T} = \frac{1}{2}\left[\dot{\mathbf{q}}_\mathrm{t}^\mathsf{T}\mathbf{M}_\mathbf{q}\dot{\mathbf{q}}_\mathrm{t} + \boldsymbol{\omega}^\mathsf{T}\mathbf{J}_\mathbf{q}\boldsymbol{\omega}\right] = \frac{1}{2}\, m_\mathrm{b}\left[\dot{x}^2 + \dot{y}^2 + \frac{2}{5}\, r_\mathrm{b}^2\left(\omega_x^2 + \omega_y^2\right)\right]. \tag{3.49}$$

Rearranging Eq. (3.13) and (3.25) for the planar case, we obtain

$$\begin{aligned} \dot{x} - r_\mathrm{b}\omega_y &= 0, \\ \dot{y} + r_\mathrm{b}\omega_x &= 0, \end{aligned} \tag{3.50}$$

because $r_x = r_y = 0$ and $r_z = -r_\mathrm{b}$. Substituting ωs from Eq. (3.50) into Eq. (3.49), the kinetic energy can be rewritten in terms of generalized inertance $\mathcal{I} = \frac{7}{5}\, m_\mathrm{b}$ and translational velocities $\dot{\mathbf{q}}_\mathrm{t}$ as

$$\mathcal{T}_2(\dot{\mathbf{q}}_\mathrm{t}) = \frac{1}{2}\,\mathcal{I}\left(\dot{x}^2 + \dot{y}^2\right). \tag{3.51}$$

Next, we can use this expression for kinetic energy in the Euler-Lagrange's equation in the standard holonomic form

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}_2}{\partial \dot{\mathbf{q}}_\mathrm{t}} - \frac{\partial \mathcal{L}_2}{\partial \mathbf{q}_\mathrm{t}} = \boldsymbol{\tau}, \tag{3.52}$$

where Lagrangian of simplified system is given by

$$\mathcal{L}_2(\mathbf{q}_\mathrm{t}, \dot{\mathbf{q}}_\mathrm{t}) = \mathcal{T}_2(\mathbf{q}_\mathrm{t}, \dot{\mathbf{q}}_\mathrm{t}) - \mathcal{U}(\mathbf{q}_\mathrm{t}). \tag{3.53}$$

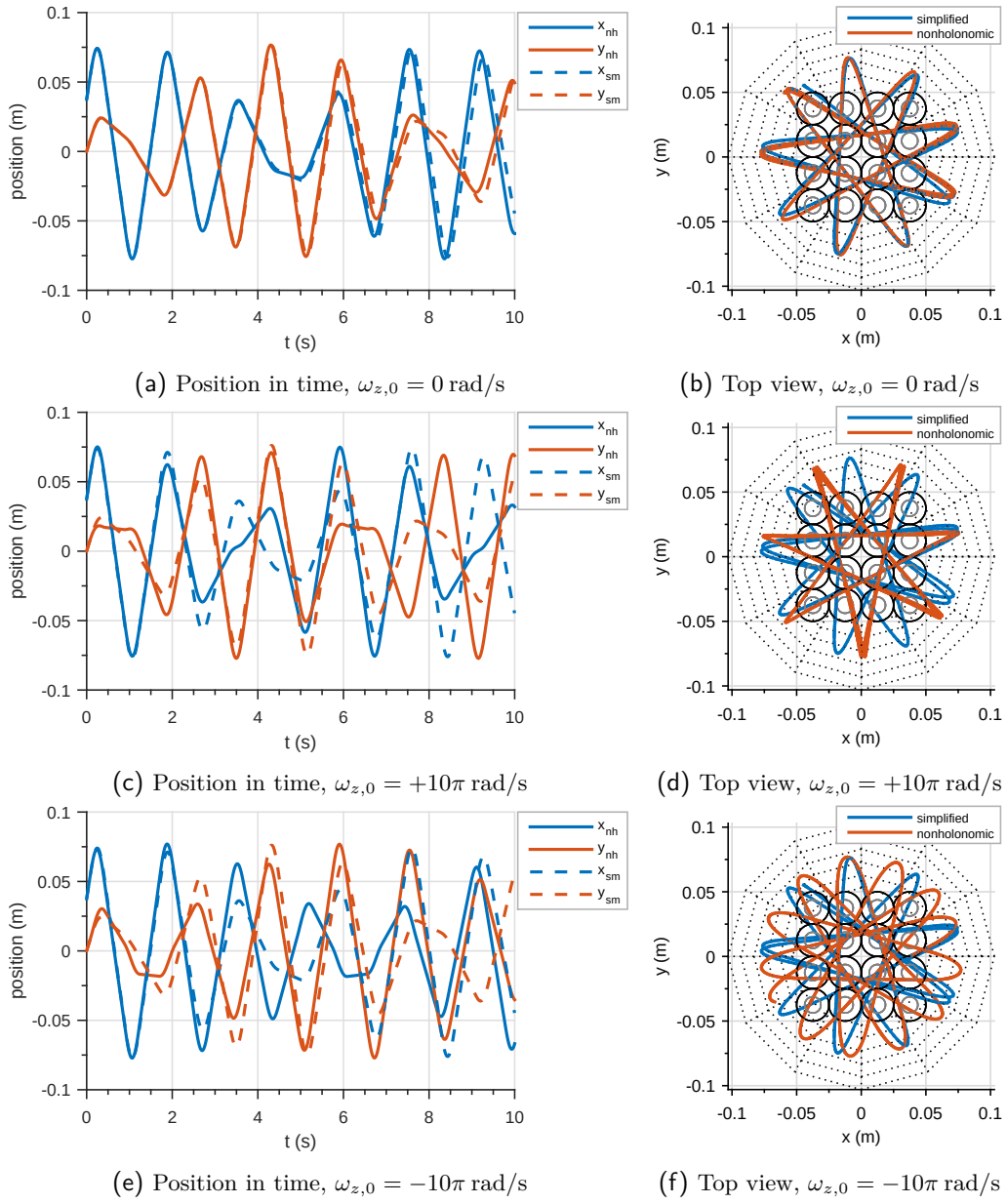
Using this model derivation, the orientation of the ball is neglected. However, obtained model is equivalent to that of Eq. (3.44) in terms of the trajectory of the ball's center on the flat surface. Moreover, as the representation of orientation is omitted, the model does not suffer from singularities. Precisely, this model is not valid on the curved parts of the surface, where r_x and r_y components are nonzero. However, for small angles of surface inclination and negligible ω_z, we may attempt to use this model and compare the simulation results to the exact model with nonholonomic constraints.

Fig. 3.5 shows comparison of responses to initial conditions and trajectories of both mathematical models. In the first case, the initial conditions were

$$\mathbf{w}_0 = \left[3r_\mathrm{c} \quad 0 \quad \tfrac{\pi}{4} \quad \tfrac{\pi}{4} \quad \tfrac{\pi}{4} \quad 20\,\mathrm{cm\,s^{-1}} \quad 10\,\mathrm{cm\,s^{-1}} \quad 0\right]^\mathsf{T}. \tag{3.54}$$

During the first simulation in Fig. 3.5a, a slight deviation between both solutions appears. This comes as the result of change in ω_z in the case of the unsimplified model. On the curved parts of surface ω_z diverts the trajectory by turning it slightly, as shown in Fig. 3.5b. In the next two simulations, we set ω_z to be initially nonzero

$$\mathbf{w}_0 = \left[3r_\mathrm{c} \quad 0 \quad \tfrac{\pi}{4} \quad \tfrac{\pi}{4} \quad \tfrac{\pi}{4} \quad 20\,\mathrm{cm\,s^{-1}} \quad 10\,\mathrm{cm\,s^{-1}} \quad \pm 10\pi\,\mathrm{rad\,s^{-1}}\right]^\mathsf{T}. \tag{3.55}$$



(a) Position in time, $\omega_{z,0} = 0\,\mathrm{rad/s}$

(b) Top view, $\omega_{z,0} = 0\,\mathrm{rad/s}$

(c) Position in time, $\omega_{z,0} = +10\pi\,\mathrm{rad/s}$

(d) Top view, $\omega_{z,0} = +10\pi\,\mathrm{rad/s}$

(e) Position in time, $\omega_{z,0} = -10\pi\,\mathrm{rad/s}$

(f) Top view, $\omega_{z,0} = -10\pi\,\mathrm{rad/s}$

Figure 3.5: Comparison of responses to initial conditions and trajectories of presented models

Depending on its sign relative to the orientation of translational velocity, ω_z makes the trajectory either more pointed (Fig. 3.5d for $\omega_{z,0} = +10\pi\,\mathrm{rad\,s^{-1}}$) or more smoothly curved (Fig. 3.5f for $\omega_{z,0} = -10\pi\,\mathrm{rad\,s^{-1}}$). As seen in Fig. 3.5c, initial nonzero ω_z produces a significant difference in trajectories. However, ω_z is initially close to zero on the laboratory platform. We have not observed the ball to acquire any significant magnitude of ω_z, unless spun by hand.

In conclusion, the simplified model can also be used to describe the ball's motion dynamics with sufficient precision. Its trajectory is almost identical to that of the exact nonholonomic model if $\omega_z = 0$. Due to its simplicity and absence of singularity, this model can be more conveniently used in design of controllers and estimators.

4 Position measurement

In this chapter, we will discuss the deployment of an existing computer vision algorithm for ball's position measurement. We will evaluate its geometric precision with respect to the curved shape of the surface and suggest a correction procedure. Finally, we will describe the model of vision-based position measurement for simulation purposes and additional utilities for setting up the computer vision algorithm.

4.1 Previous solutions

Originally, the platform offered two implementations of position measurement. The first solution was based on a resistive foil. However, resistive foils are available only in flat rectangular shapes, which would not fit our circular curved surface. Moreover, manufactures do not recommend folding the foil which would be necessary to accommodate it to the curved surface. Therefore, this solution is not suitable.

The second solution of position measurement is a RGB camera observing the platform from the top view. The advantage of a camera is the possibility to measure position of several objects simultaneously. On the other hand, vision based position measurement is limited in the maximum sampling frequency. Achieving a reasonable sampling frequency is a critical requirement for proper function of any feedback controller.

We will be using the implementation of camera based position measurement developed in [11]. The algorithm consist of two main steps. Firstly, each pixel is classified based on its color in RGB colorspace. The classification is done using *hue* and *saturation* thresholds transformed to *isosurfaces* in RGB colorspace. Secondly, the classified binary image is summed along rows and columns, forming two one-dimensional signals. These two signals are spatially filtered using a mask of secant line lengths. The mask can be formed by creating an image of a circle with the same value of radius in pixels as the used ball has and subsequently summing the number of circle's pixels in each row or column. Computing correlation of each of the two signals with the mask, the center of the ball is determined at the point of correlation's maximum. In our setup, the surface is made of white plastic. This material further decreases the classification error, because such background is easier to distinguish from the colored ball.

Lastly, the camera calibration is used to determine the position of the ball's center in the world coordinates. The mapping between image plane and platform's horizontal plane is described by a projective transformation called *homography*. However, since the new surface is curved, the scene can no longer be assumed planar. Therefore, using homography introduces a measurement error.

4.2 Measurement error estimation

In this section, we aim to estimate the error introduced by the use of homography. To determine the position of the ball precisely, we would have to reconstruct the ray given by the center of the ball in the image plane using a known camera calibration and then solve the intersection of the ray with the surface in the world coordinate frame. This will lead to solving a *line-torus* intersection in the form of a 4th order polynomial equation. This intersection problem is considered to be complex and time consuming in [12], where various formulations of the line-torus intersection problem are provided. Therefore, attempting to obtain the precise solution will be rather tedious in our application. Instead, we will use simplifying assumptions to obtain an estimate of the ball's position with sufficient precision.

In the laboratory setup, the camera is placed roughly above the center of the surface and aimed down. For simplicity of the estimation, we assume that the camera is centered above the surface and its optical axis is aligned with the axis of surface's rotational symmetry. We can approximately achieve such setting and, therefore, the effect of deviations from this ideal setting is considered negligible.

Next, we can base our estimate upon a 2D pinhole camera model, such as described in the chapter 11 of [13]. Uncommonly, we will consider the 2D model in the radial ρz cross section with z axis pointing downward along the optical axis. As the surface is assumed to be symmetric with respect to the optical axis, the error can be examined based on the radial distance ρ of a given point from the center of the surface. The coordinate ρ_i of an object in the image plane is given as

$$\rho_\mathrm{i} = f\frac{\rho_\mathrm{c}}{z_\mathrm{c}}\,, \tag{4.1}$$

where $\rho_\mathrm{c}, z_\mathrm{c}$ are coordinates of the object expressed with respect to the camera coordinate frame so that z_c denotes the distance of the object along the camera's optical axis and ρ_c is the radial distance perpendicular to it. The focal length of camera is denoted as f. We will estimate the error by evaluating the distance of the original point and the projection of the corresponding image point back to the platform's planar surface. The z_c coordinate of ball's center of mass placed on the curved part of the surface is given as

$$z_\mathrm{c}(\rho_\mathrm{c}) = z_0 - \left(r_\mathrm{e} - \sqrt{r_\mathrm{e}^2 - (\rho_\mathrm{c} - R)^2}\right), \tag{4.2}$$

where z_0 is the distance from camera to the platform plane. We project the point to the image plane using Eq. (4.1) and then project it back to the plane given by $z_\mathrm{c} = z_0$, the position of projection $\hat{\rho}_\mathrm{c}$ is

$$\hat{\rho}_\mathrm{c} = \frac{\rho_\mathrm{c}\, z_0}{z_0 - \left(r_\mathrm{e} - \sqrt{r_\mathrm{e}^2 - (\rho_\mathrm{c} - R)^2}\right)}\,, \tag{4.3}$$

independent of the focal length f. The estimate of measurement error introduced by the use of homography without correction is

$$\mathbf{e}(\rho_\mathrm{c}) = \begin{cases} \begin{bmatrix} 0 & 0 \end{bmatrix}^\mathsf{T} & \text{if } \rho_\mathrm{c} \in \langle 0,\, R\rangle, \\ \begin{bmatrix} \hat{\rho}_\mathrm{c} - \rho_\mathrm{c} & z_0 - z_\mathrm{c}(\rho_\mathrm{c}) \end{bmatrix}^\mathsf{T} & \text{if } \rho_\mathrm{c} \in (R,\, r_\mathrm{build}\rangle, \\ \text{undefined} & \text{elsewhere.} \end{cases} \tag{4.4}$$

Finally, we will evaluate the measurement error for $z_0 = 40\,\mathrm{cm}$. Fig. 4.1 shows reprojection error of the ball's center in the camera coordinate frame. Red arrow points from the real to the measured position of the object.

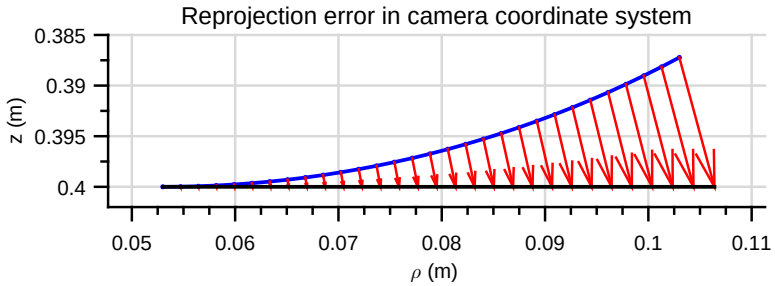


Figure 4.1: Reprojection error

4.3 Error correction

In this section, we will develop a procedure for the correction of position measurement error which will enable us to approximately reconstruct the position of the ball's center in $3D$ space.

We are currently using the implementation where outputs are the $x_\mathrm{n}, y_\mathrm{n}$ coordinates of the ball's center in the platform's horizontal plane expressed in world coordinate frame with lengths normalized to multiples of coil's diameter. As the first step of the correction procedure, these coordinates will be rescaled to metric lengths and denoted as x_m, y_m. As the scene is radially symmetric, we will examine it using cylindrical coordinates derived from the world coordinate frame. The polar coordinates are defined as

$$\begin{aligned} \rho_\mathrm{m} &= \sqrt{x_\mathrm{m}^2 + y_\mathrm{m}^2}, \\ \phi_\mathrm{m} &= \operatorname{atan2}(y_\mathrm{m}, x_\mathrm{m}), \end{aligned} \tag{4.5}$$

where atan2 is four quadrant version of arctangent function. If $\rho_\mathrm{m} \in \langle R, r_\mathrm{build} \rangle$, geometric correction is required and the procedure continues. The position of the ball's center determines a single ray, a line which passes through the camera coordinate frame origin and the measured point. Fixing the angle ϕ_m, the line is expressed in cylindrical coordinates as

$$z_1(\rho_\mathrm{w}) = z_0 - \frac{z_0}{\rho_\mathrm{m}}\,\rho_\mathrm{w}, \tag{4.6}$$

where ρ_w is the parameter of the line. Every point $[\rho_w \;\; z_1(\rho_w)]^\mathsf{T}$ that satisfies Eq. (4.6) projects to the same point in the image plane according to Eq. (4.1). Therefore, this line is the set of all possible positions of the ball's center. Next, we will intersect the line with the equation describing the possible positions of ball's center on the curved side of the surface in the world cylindrical coordinate frame

$$z_2(\rho_\mathrm{w}) = r_\mathrm{e} - \sqrt{r_\mathrm{e}^2 - (\rho_\mathrm{w} - R)^2}. \tag{4.7}$$

The true position of the ball's center is obtained by solving $z_1 = z_2$ for ρ_w. This equation has two solutions from which we choose the one corresponding to the position of the ball's center

$$\rho_\mathrm{w} = \frac{R\rho_\mathrm{m}^2 + \rho_\mathrm{m} z_0^2 - r_\mathrm{e}\rho_\mathrm{m} z_0 \sqrt{\rho_\mathrm{m}^2 z_0 \left[r_\mathrm{e}^2 z_0 + 2r_\mathrm{e}\rho_\mathrm{m}(\rho_\mathrm{m} - R) - z_0(R - \rho_\mathrm{m})^2\right]}}{\rho_\mathrm{m}^2 + z_0^2}. \tag{4.8}$$

Finally, the position of the ball in Cartesian world coordinate frame is given by

$$\begin{bmatrix} x_\mathrm{w} & y_\mathrm{w} & z_\mathrm{w} \end{bmatrix}^\mathsf{T} = \rho_\mathrm{w} \begin{bmatrix} \cos(\phi_\mathrm{m}) & \sin(\phi_\mathrm{m}) & \frac{z_0}{\rho_\mathrm{m}} \end{bmatrix}^\mathsf{T}. \tag{4.9}$$

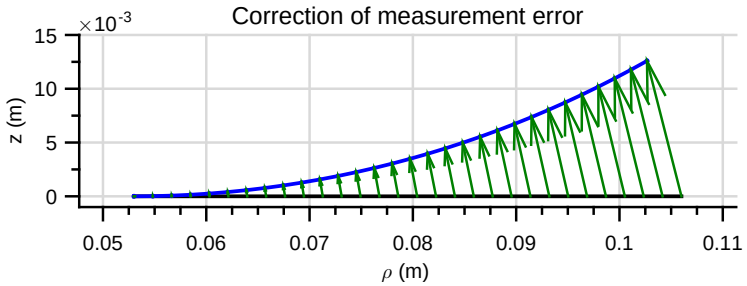


Figure 4.2: Correction of position measurement

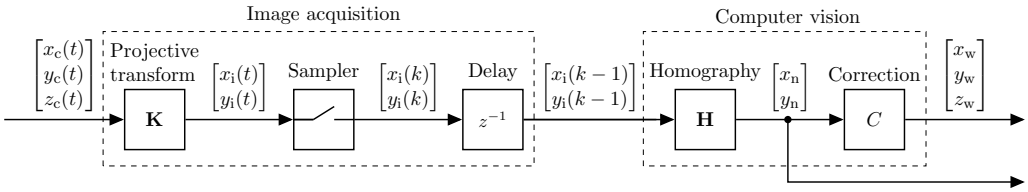


Figure 4.3: Scheme of camera model

This procedure can be used to reconstruct the position in the curved region with a satisfactory precision as we assume the use of homography without correction to be the main source of the error. Other sources of error, such as the deviation of camera from described position, estimation of z_0 and use of simple pinhole camera model, are considered negligible. Fig. 4.2 illustrates the correction of position measurement for camera distance $z_0 = 40$ cm.

4.4 Camera model

In this section, we describe the model of camera used in conjunction with the mathematical model for the simulation of controlling the laboratory platform.

In the Simulink implementation, the output of the ball's motion model is the ball's position in meters. In the camera model, we scale the position to multiples of coil diameter first. This operation corresponds to the projective transform and subsequent homography in the case of the real platform. Next, we add *Zero order hold* block, which accounts for the discrete-time sampling and operates at the same sampling frequency as the camera.

After sampling, the signal is passed to *Unit delay* block. This delay models the time requirements of acquisition, transfer and processing of the image. Specifically, camera's exposure time is set to 7.5 ms, transfer of the image from the framegraber to the Simulink environment takes approximately 14 ms and 4.6 ms is necessary on average for the algorithm to determine the ball's position from the image, according to [11]. Together, the position of the ball is known no sooner than 26.1 ms after the start of the camera exposure. Therefore, this delay will be modeled using the length of one sampling period of the camera, denoted as $T_{\mathrm{s,cam}}$. Finally, the signal is passed to the other parts of the control system, such as a controller or an estimator. The scheme of the camera model is depicted in Fig. 4.3.

In the Simulink model of the camera, we have not implemented projective transform (only scaling of units instead), homography and correction for the purpose of

controller behaviour simulation on the mathematical model. However, these block are considered during experiments with real platform, with projective transform being a natural property of the camera, homography implemented within the algorithm of [11] and the correction used to reconstruct the position in 3D.

4.5 Additional utilities

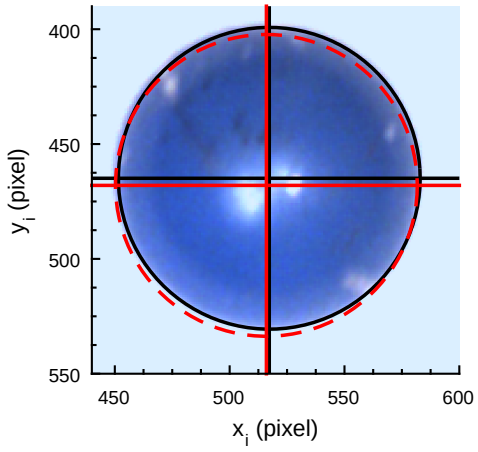
The quality of position measurement depends strongly on the choice of hue and saturation thresholds. As in any classification problem, setting the threshold values is the task of finding an optimal trade-off between the count of false positive and false negative classifications. As the classified image is subsequently filtered to obtain the position, it is difficult to assess the effect of threshold changes based on the binary image alone. One of possible ways is to compare the algorithm's results to *ground truth data*.

Previously, a set of acquired images was annotated by hand to obtain histograms of foreground and background pixels. Because the shape of the object is elemental, the annotation can be done in a fully automated fashion. Matlab provides an implementation of circular object segmentation from an image. The function is named `imfindcircles` and is based on circular *Hough transform*. However, it only accepts single channel images. Therefore, we chose to run the annotation on the saturation channel, in which the ball's outline is easily distinguishable from the background, regardless of the ball's color. Resulting annotation output for a given set of images consists of ball's centers and radii. The average value of the radius can be used for generation of properly sized mask for spatial filtering. Center's in images are determined with high precision and can, therefore, be used for the evaluation of measurement error caused by imperfections of the classification. Comparisons of annotation and detection in RGB images are shown in Fig. 4.4a and 4.4b. Described functions can be found on the attached CD, namely the annotation is implemented in `annotateCirc` and the visualisation of its results in `plotAnnotation`.

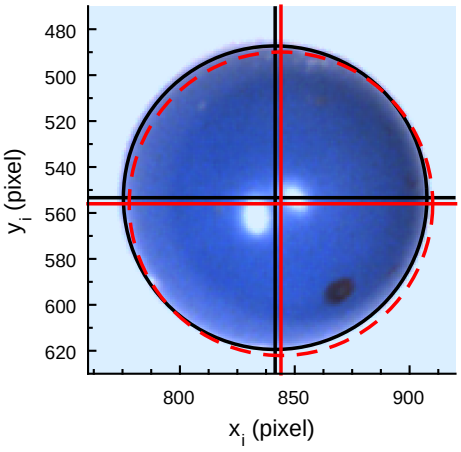
Using a larger set of annotated images, we can statistically evaluate the error of position measurement for given thresholds. Error evaluation can be run using script `detectBall` followed by `detectionError`. Fig. 4.4d shows error vectors determined by subtracting the annotation from the detection. The error was evaluated on a set of one hundred frames of moving blue ball. The average error is $-0.05\,\mathrm{mm}$ in x_m direction and $0.54\,\mathrm{mm}$ in y_m direction for ball with radius $r_\mathrm{b} = 15\,\mathrm{mm}$.

The quality of the classification is deteriorated by several factors, namely the reflection of scene's lighting and scraped paint on the ball's surface which both contribute to false negative classifications, as seen in Fig. 4.4c. The position obtained by spatial filtering may also be slightly biased by greater numbers of false positive pixels located along the cross determined by the ball's position.

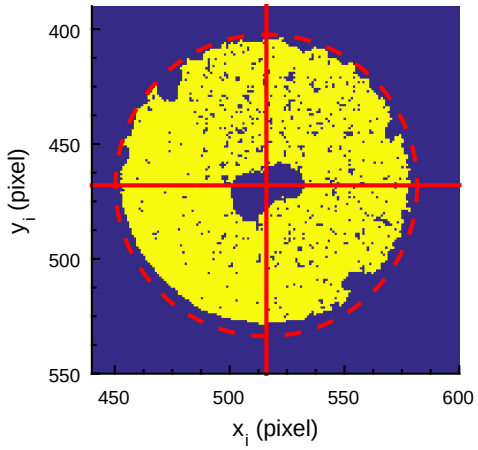
In conclusion, used annotation function is highly time-consuming, but it is well suited for the offline processing, where precision is the main goal. Automated annotation proved to be useful for adjustments of position measurement algorithm when the illumination of the scene or the appearance of the ball was changed. It allows for a quick evaluation of the measurement error of current HSV threshold. Matlab implementation of described functions and scripts are located in the `vision` directory of the attached CD. For more details, see Appendix A.



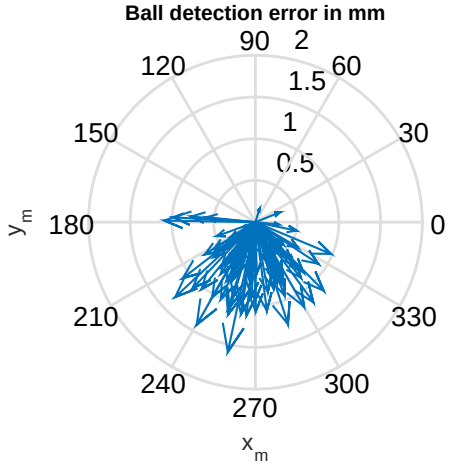
(a) Annotation (black) vs detection (red)



(b) Annotation (black) vs detection (red)



(c) Binary mask from HSV thresholding of RGB image a)



(d) Detection error based on 100 frames of a blue ball with $r_\mathrm{b} = 15\,\mathrm{mm}$

Figure 4.4: Comparison of automated annotations with detections for evaluation of the computer vision algorithm's error.

5 | Feedback control

In this chapter, the design of feedback controller for the laboratory platform is described. The aim of the feedback controller is to achieve steady oscillations of a ball along a chosen radial axis of the platform. Based on the dynamics of the mathematical model, the controller must be able to maintain constant velocity of the ball in the flat region, suppress any displacement perpendicular to the radial direction and prevent the ball falling off the platform to fulfil the task. If not stated otherwise, we will be using a steel ball with mass $m_\mathrm{b} = 110$ g and radius $r_\mathrm{b} = 15$ mm.

From the control point of view, this task belongs to the area of *visual servoing*, which is a term for feedback control based on visual measurement [5]. Key characteristic of visual servoing is that the controlled variables are not measured directly by the sensor. Instead they are obtained from the visual measurement using *image processing* and *computer vision* algorithms. Our particular task can be termed as *single-view*, because only one camera is currently used, and *eye-to-hand*, as the configuration of camera is held fixed.

5.1 Feedback linearization

The design of the feedback controller is based on the *feedback linearization* developed in [1] as its underlying layer. The output of the controller is the vector of desired force $\mathbf{F}_\mathrm{des}$. In the process of feedback linearization, the contributions of individual coils are evaluated according to Eq. (1.4), depending on the coil's relative position to the ball. This steps provides 16 constant (activation-independent) coefficients which are further scaled according to the activation of coils. Next, numerical optimization is used to obtain such values of activation $\mathbf{U}_{m\times n}$ that the error between desired and resulting force as well as the total activation of coil's are minimized.

It is important to mention that the effectiveness of the feedback linearization is strongly dependent not only on the accuracy of the coil's mathematical model, but also on the precise knowledge of the ball's immediate position. This fact, together with visual-based position measurement, limits the performance of the controller, because the position is updated with a relatively low sampling frequency and is delayed due to nonnegligible processing time of every image. As a result, the real value of contribution coefficient is varied. Hence, during activation, the force produced by the coil is different from the requested value. Moreover, the force generated by an activated coil changes with the changes of the ball's position in between the sampling of the control system. This phenomenon is most distinctive during a transition over an activated coil.

Therefore, achieving a reasonably high frequency is vital for the feedback linearization in order to operate properly and yield usable results. The consequence of imprecise linearization can be considered as unmeasurable *input disturbance*, which is added to the desired force. We aim to decrease its effect by providing the feedback

linearization with as accurate entries as possible. Apart from sampling frequency, both the accuracy of camera's calibration and the computer vision algorithm play a key role. Even slight changes in the performance of any of the aforementioned components can have a significant impact on the functioning of the system as whole.

5.2 System description

In previous works, magnetic field was modeled and experimentally identified only in the platform's xy plane. The effect of change in z coordinate on the resultant input force has not been modeled nor identified yet. Therefore, our aim is to influence the ball mainly in the flat region, where we can rely on the results of the feedback linearization.

For the task of controller design we will use the simplified mathematical model of the ball's motion on a horizontal planar surface, as described in Section 3.9. For the verification and simulation of the control system behaviour, we will make use of both mathematical models with incorporated curved sides of the surface. The simplified model of ball's motion dynamics rewritten in state space form is

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \underbrace{\left[\begin{array}{cc:cc} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right]}_{\mathbf{A}} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} + \underbrace{\mathcal{I}^{-1} \left[\begin{array}{c:c} 0 & 0 \\ 1 & 0 \\ \hdashline 0 & 0 \\ 0 & 1 \end{array}\right]}_{\mathbf{B}} \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix}, \tag{5.1}$$

where $\mathbf{x}$ denotes the state vector. Due to unity with previous works, we will design the controller using position normalized to multiples of a coil diameter in meters. The output equation of the system is given as

$$\mathbf{y} = \mathbf{C}\mathbf{x} = \frac{1}{d_\mathrm{c}} \left[\begin{array}{cc:cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \mathbf{x}, \tag{5.2}$$

where d_c denotes the diameter of a coil as well as the distance of centers of two neighbouring coils. Its reciprocal is used for the scaling of the position.

If we neglect the effect of dissipative forces in $\boldsymbol{\tau}$ and consider the components of input force as decoupled, the system can be described by two independent double integrators, each modeling the ball's motion along one axis. In fact, the friction on the laboratory platform is rather insignificant, especially when the ball is not influenced by the input force. Attracting the ball to a coil produces a nonzero component of force in the direction of the surface normal. This force is compensated by the surface's reaction. As a result, the effect of dissipative forces increases. As the friction model of the laboratory platform is rather complex yet does not play a vital role for the controller's design, we will either consider only linear friction or omit the friction completely in some parts of the design.

If not damped, the system of ball's dynamics in plane is linear and marginally stable, with all four poles located in $s = 0$. Additionally, the system is both fully observable and controllable. The behaviour of the ball on the curved sides of the surface is nonlinear and in some respect similar to a spherical pendulum. During the motion on the curved parts, the x, y components of the velocity begin to influence each other. The coupling of x, y components in the curved region has to be considered in order to steer the ball in the desired way.

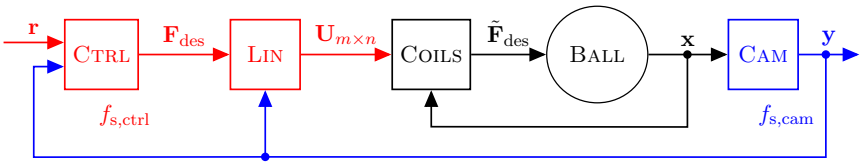


Figure 5.1: Structure of multirate control system. Subsystems and signals are colored according to their sampling frequency. Black color denotes continuous time.
Legend: Ctrl – controller, Lin – Feedback linearization, Cam – Vision-based position measurement

5.3 Multirate control

As mentioned in Section 5.1, the performance of the control system relies on the precision and the frequency of the position measurement. However, with current hardware equipment and implementation of the computer vision algorithm, we are *not* able to reach a sampling frequency higher than approximately $f_{\mathrm{s,cam}} = 45\,\mathrm{Hz}$, due to reasons described in Section 4.4.

Nevertheless, the performance may be further improved by use of *multirate digital control*, that is by use of more than one sampling frequency. In the case of the laboratory platform, such approach is possible as the frequency of sending the control input to the modules can reach up to $f_{\mathrm{s,ctrl}} \approx 300\,\mathrm{Hz}$.

Moreover, multirate control is supported by Simulink as one of possible implementations of concurrent execution. Using the *rate based tasks* option, the Simulink model can be implicitly partitioned by setting a sampling frequency for each subsystem. The transition of signal between subsystems operating at different rates can be handled automatically in this setting. The scheme of multirate control applied to the laboratory system is depicted in Fig. 5.1.

The sampling frequency of the controller is chosen to be a positive integer multiple of the camera sampling frequency: $f_{\mathrm{s,ctrl}} = n f_{\mathrm{s,cam}},\ n \in \mathbb{N}$. Using suggested scheme, the controller can additionally change its output control value $(n-1)$ times in between two consecutive position measurements.

During early experiments we used only the camera sampling frequency in the whole control system. Using $f_{\mathrm{s,cam}} \leq 30\,\mathrm{Hz}$ proved to be insufficient for the control of an oscillating ball. Often, the controller was unable to recover from a feedback linearization inaccuracy and consequently destabilized the system.

At $f_{\mathrm{s,cam}} = 45\,\mathrm{Hz}$ we first managed to maintain controlled oscillations, however, the system was very sensitive to disturbances and required a longer period of time before the oscillations became steady. By incorporating the multirate control alone, as shown in Fig. 5.1, we achieved a significant improvement. This fact is attributed to the dynamics of the controller operating with the input of slower rate.

The sampling frequency of the controller in the multirate control scheme was set to the maximum positive integer multiple of the camera sampling which complies the restriction of maximum communication frequency with the modules

$$f_{\mathrm{s,ctrl}} = 6 f_{\mathrm{s,cam}} = 270\,\mathrm{Hz}. \tag{5.3}$$

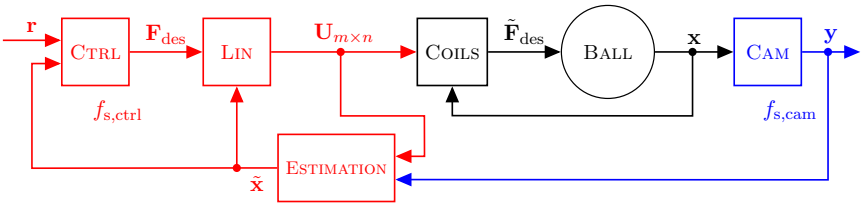


Figure 5.2: Structure of multirate control with position estimation.

5.4 Estimation

In order to improve the control performance by further utilizing the multirate control, an estimator of state was designed. Its task is to estimate the development of the ball's position in between the measurement samples. The estimator is operating at the faster frequency $f_{\mathrm{s,ctrl}}$ and uses the measured position $\mathbf{y}$ sampled at $f_{\mathrm{s,cam}}$ and the activation of coils $\mathbf{U}_{m\times n}$ sampled at $f_{\mathrm{s,ctrl}}$ as its inputs.

Based on the known mathematical model of the ball's motion dynamics and the magnetic force at the corresponding position, we can calculate the estimate $\hat{\mathbf{x}}$ of the real position $\mathbf{x}$. Reducing the sampling period of the estimation, we can obtain multiple estimates of position in between two consecutive measurements. In each sampling period of the controller and observer, the estimate of position may be used to adjust the activation of the coils in an attempt to match the resulting force to the requested. Similarly, the estimate of position may be used in the position feedback to update the tracking error, allowing for a new command on the desired force. The position measurement is used for periodical correction of the estimate to compensate for the model uncertainties. The scheme of the system with added estimator is shown in Fig. 5.2.

5.4.1 Linear observer

Since the system is fully observable, we base the estimator on the well-known discrete time linear observer. First, let us write down the state space form of the continuous system of Eq. (5.1) with a zero order hold at its input and a sampler at the output

$$\mathbf{x}(k+1) = \underbrace{\left[\begin{array}{cc:cc} 1 & T_{\mathrm{s}} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 1 & T_{\mathrm{s}} \\ 0 & 0 & 0 & 1 \end{array}\right]}_{\mathbf{\Phi}=e^{\mathbf{A}T_{\mathrm{s}}}} \mathbf{x}(k) + \underbrace{\mathcal{I}^{-1}\left[\begin{array}{cc:cc} \frac{T_{\mathrm{s}}^2}{2} & 0 \\ T_{\mathrm{s}} & 0 \\ \hdashline 0 & \frac{T_{\mathrm{s}}^2}{2} \\ 0 & T_{\mathrm{s}} \end{array}\right]}_{\mathbf{\Gamma}=\mathbf{B}\int_0^{T_{\mathrm{s}}} e^{\mathbf{A}\nu}\mathrm{d}\nu} \boldsymbol{\tau}(k), \tag{5.4}$$

where $T_{\mathrm{s}} = 1/f_{\mathrm{s,ctrl}}$ is the used sampling time.

We will use the *prediction estimator* in the form described in [14]. The estimate of the state vector is obtained using

$$\hat{\mathbf{x}}(k+1) = \mathbf{\Phi}\hat{\mathbf{x}}(k) + \mathbf{\Gamma}\hat{\boldsymbol{\tau}}(k) + \mathbf{L}(\mathbf{y}(k) - \hat{\mathbf{y}}(k)), \tag{5.5}$$

where $\hat{\mathbf{y}}(k) = \mathbf{C}\hat{\mathbf{x}}(k)$ is the observer's output, $\mathbf{L}$ is the matrix of output injection and $\hat{\boldsymbol{\tau}}(k)$ is the estimated value of input force based on the activation of coils $\mathbf{U}_{m\times n}$ and position estimate $\hat{\mathbf{x}}(k)$. The matrix of observation error's dynamics is

$$\mathbf{\Phi}_{\mathrm{obsv}} = \mathbf{\Phi} - \mathbf{LC}. \tag{5.6}$$

Provided that all eigenvalues of $\mathbf{\Phi}_{\text{obsv}}$ are within the unit circle, estimation error is asymptotically stable. Moreover, eigenvalues of $\mathbf{\Phi}_{\text{obsv}}$ may be arbitrarily chosen by setting the gains in matrix $\mathbf{L}$. For our application, we placed the observer eigenvalues into stable yet relatively slow locations

$$\boldsymbol{\lambda}_{\text{obsv}} = \begin{bmatrix} 0.9 + 0.05i \\ 0.9 - 0.05i \\ 0.9 + 0.05i \\ 0.9 - 0.05i \end{bmatrix}, \quad \text{by choosing} \quad \mathbf{L} = \begin{bmatrix} 0.005 & 0 \\ 0.07 & 0 \\ 0 & 0.005 \\ 0 & 0.07 \end{bmatrix}. \tag{5.7}$$

Gains in matrix $\mathbf{L}$ are relative low, but sufficient to stabilize the estimate of the state. Hereby, the estimation relies more on the model instead of the position measurement, which is held constant in between the samples and would otherwise restrict the desired development of estimate during this period. However, due to slower poles of the observer, measured output of the system has a lower influence on the estimate of the state. As a result, the quality of estimate suffers more from unmodeled disturbances.

5.4.2 Periodically reset model of the system

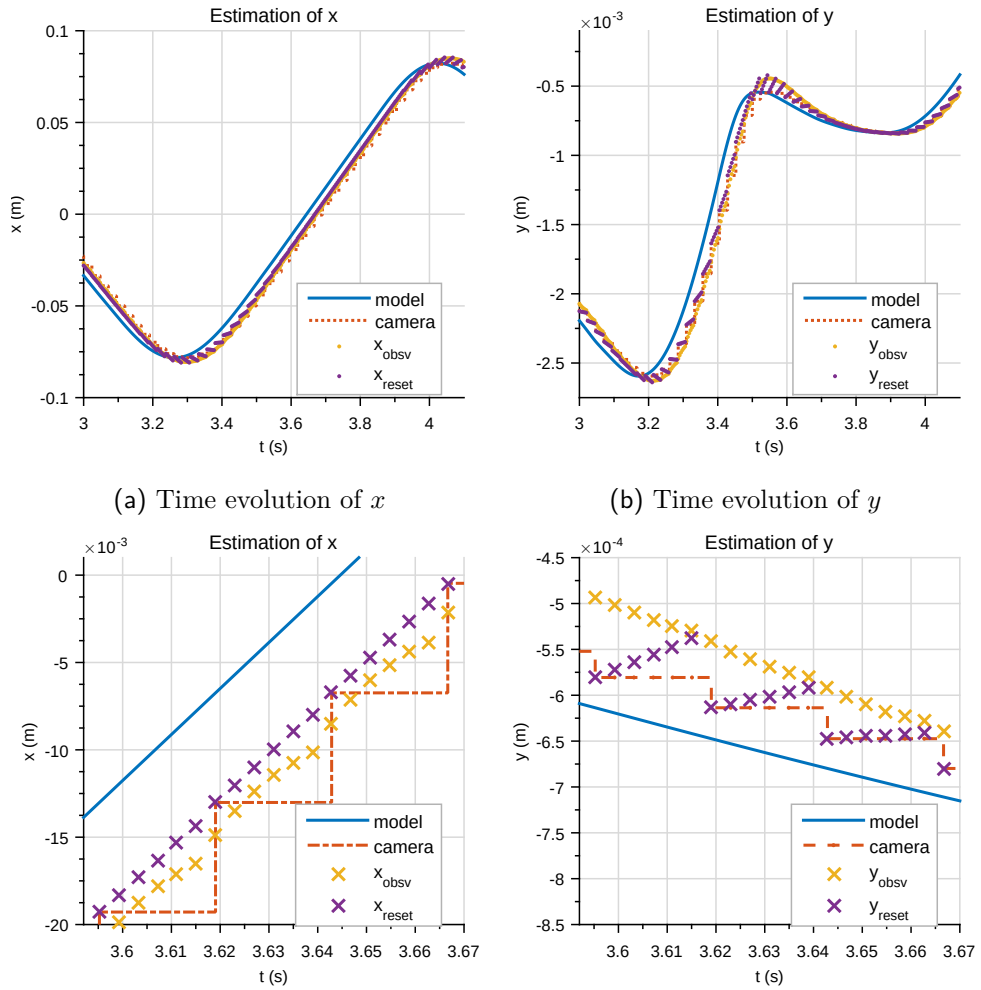
Another possible strategy of estimation further extends the previous solution. In between two position measurements, the estimation will be based only on the mathematical model. We will be simulating the dynamics of the discretized plant using Eq. (5.4). When a new measurement of position arrives, we will reset the estimate. The new value of the state vector will be composed of the currently measured position and the velocity estimated by the observer. We use the observer's velocity estimate, because it proves to be more precise and responsive in the simulation, however, it requires the observer to be operating in parallel in the control scheme. Mathematically, the operation of this estimator can be written as

$$\hat{\mathbf{x}}_{\text{sim}}(k) = \begin{cases} \mathbf{\Phi}\hat{\mathbf{x}}_{\text{sim}}(k-1) + \mathbf{\Gamma}\hat{\boldsymbol{\tau}}_{\text{sim}}(k-1) & \text{for } t_k \neq nT_{\text{s,cam}},\ n \in \mathbb{N}, \\ \begin{bmatrix} x_{\text{m}}(k) & \hat{v}_{\text{x}}(k) & y_{\text{m}}(k) & \hat{v}_{\text{y}}(k) \end{bmatrix}^{\mathsf{T}} & \text{for } t_k = nT_{\text{s,cam}},\ n \in \mathbb{N}, \end{cases} \tag{5.8}$$

where $x_{\text{m}}(k),\ y_{\text{m}}(k)$ are current position measurements and $\hat{v}_{\text{x}}(k),\ \hat{v}_{\text{y}}(k)$ denote the the previous velocity estimates of the observer. This estimate can be used in the time step t_k since it either predicts the state based on the previous state and input or uses the current measurement together with available velocity predictions.

5.4.3 Simulation

In this section, the simulation results of both suggested estimation schemes are presented. The simulation was executed using the mathematical model of Section 3.9. Discretization and the unit delay caused by camera is used in accordance with the description in Section 4.4. Simulated model also includes a controller of oscillations, which will be described later in Section 5.5. It is used here to illustrate the behaviour of the estimators incorporated in the complete control scheme. Currently, the controller is using the measured position as its input. In the simulation, the ball was initially placed at a random position in the flat region. The aim of the controller was to regulate y coordinate while simultaneously forcing x coordinate to oscillations. At the same time , the aim of the estimators is to produce values as close as possible to those of mathematical model, which we consider as the reference.



(a) Time evolution of x

(b) Time evolution of y

(c) Detailed view of x estimates

(d) Detailed view of y estimates

Figure 5.3: Simulation of position estimation. Upper figures depict the long term behaviour, lower figures provide a close-up view of individual samples. Index obsv resp. reset denotes the observer's resp. the reset estimator's position estimate.

For comparison, we selected data from the moment when oscillations are becoming steady. Fig. 5.3a shows the x coordinate of the ball during one transition across the surface. Fig. 5.3b depicts the y coordinate of the ball, which is being regulated during the transition. From these two views, it can be seen that the estimate based on periodically reset model produces more accurate results during the motion across the flat region compared to results of the linear observer. However, velocity estimate of the reset estimator suffers more from the change of model and disturbances, producing a ripple of the position estimate, which is especially notable, when the motion's direction changes. Examining the simulation from a closer view, Fig. 5.3c shows a reduction of the delay by the reset estimator. During the motion over the flat part of the surface, this prediction produces a fine extrapolation of the measurements. Linear observer provides a similar yet slightly more delayed estimates of model's position. On the other hand, the observer is less susceptible to changes in the model during the motion on curved sides, which were not incorporated in the estimators to reduce the computation time. Nevertheless, the estimates will mainly

Table 5.1: PID controller gains for $m_{\mathrm{b}} = 110\,\mathrm{g}$

Parameter	Tested interval	Final value
K_{P}	$\langle 0.02;\ 0.10 \rangle$	0.065
K_{I}	$\langle 0.005;\ 0.03 \rangle$	0.025
K_{D}	$\langle 0.005;\ 0.02 \rangle$	0.015
N	$\langle 10;\ 50 \rangle$	30

be used during the transition over the flat region, where we can actually influence the ball by the input force. As can be seen in Fig. 5.3d, in some cases the estimation may produce less accurate result compared to the original measurement. However, such error is only temporary and is corrected in the course of few measurement time steps. Both estimators offer a usable form of position extrapolation between current and forthcoming measurement. However, the compensation of the full one-period camera delay is not implemented in the estimation yet.

Both of the estimators were applied during the testing of controllers on the laboratory platform and proved to be usable in the feedback loop. However, due to the difficult repeatability of experiments and large number of factors, it is hard to assess which estimator results in the better overall performance in the control task.

5.5 Controller

This section deals with the design of the controller for the laboratory platform. In Section 5.4, we described two possible approaches leading to an improvement of position measurement. Subsequently, we will use the estimate of state $\hat{\mathbf{x}}$ to control the real state of system $\mathbf{x}$.

5.5.1 PID control

As the main building block of the controller, we chose to use the well-known PID in the parallel form with the control τ given as

$$\tau = \left(K_{\mathrm{P}} + K_{\mathrm{I}} \frac{1}{s} + K_{\mathrm{D}} \frac{N}{1 + N\frac{1}{s}} \right) \epsilon, \tag{5.9}$$

where ϵ is the tracking error and N is the filter coefficient of the derivative component. First, we attempted to design the controller gains based on the mathematical model. However, due to negligence of friction, produced control input was insufficient to steer the ball according to the given command. Subsequently, the controller gains were tuned empirically. The system was observed and the gains were manually adjusted to produce a reasonable behaviour of the controlled system. Resulting set of gains and intervals, over which the manual tuning was performed, are presented in Tab. 5.1.

Due to the limitation of the platform on maximal exerted force, which were illustrated in Fig. 1.2b, the output saturation of the controller was set to $\mathbf{F}_{\mathrm{des}} = \pm 0.2\,\mathrm{N}$. To compensate for the output saturation, *clamping* was selected as the used *anti windup* method.

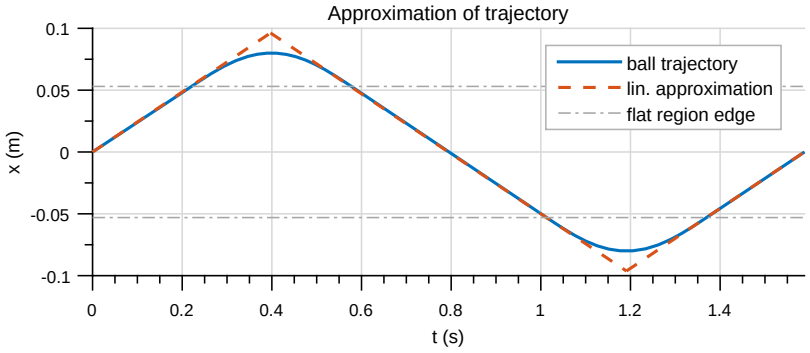


Figure 5.4: Linear approximation of the trajectory used as the reference of PID_x

5.5.2 Reference tracking

The next task is to extend the control system to fulfil the task of maintaining controlled oscillations of the manipulated object. The first proposed solution to the task is based upon the controller described in Subsection 5.5.1.

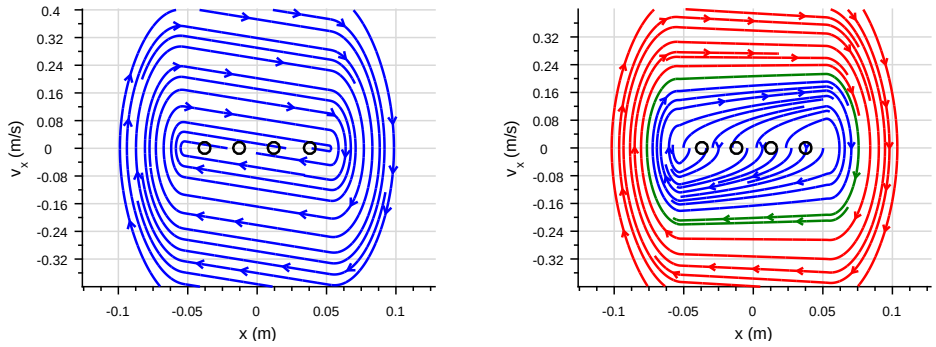
As we aim to produce controlled oscillations along a radial axis, any displacement perpendicular to it must be regulated to reduce the coupling effect at curved sides. Let us select the x axis as the direction of oscillations. Then, the y position of the ball will be controlled by the PID_y with the reference set to zero.

Next, we attempt to generate a reference command for the controller PID_x of the x coordinate to produce steady controlled oscillations. We will base the reference on the ideal behaviour of the mathematical model and feed it to the PID_x controller in order to produce the same behaviour on the real platform.

Let us suppose that the ball is initially placed in the center of the platform with nonzero velocity v_x. In the absence of the friction, the ball will oscillate, maintaining its velocity. Theoretically, the ball may exit the platform's surface if its velocity exceeds $v_{\mathrm{exit}} \doteq 43\,\mathrm{cm\,s^{-1}}$. During steady oscillations, we require the ball to maintain a constant velocity in the flat region and to remain on the surface.

For the choice of initial velocity of $v_x = 24\,\mathrm{cm\,s^{-1}}$, the oscillatory trajectory is depicted in Fig. 5.4. During experiments, this velocity appeared well controllable with respect to the sampling and limitations of actuators to the maximal exerted force. Additionally, this velocity offers a sufficient safety margin from v_{exit}. Moreover, it is possible to approximate the trajectory using a piece-wise linear function with a slope corresponding to desired velocity and the period same as that of the original trajectory. Such approximation was found experimentally and is depicted in Fig. 5.4 for comparison.

One shortcoming of this strategy is that it does not take into account the initial position of the ball. It relies on the controller that the ball will be eventually forced to track the reference. Once the oscillations become steady, the controller is suitable for their maintenance. However, it may require some additional time before the oscillations become synchronized with the reference at the beginning of the experiment. Reference is completely independent of the ball's current position, which has both benefits and drawbacks. It allows to dictate the position of the ball, which may be useful for control of multiple balls. On the other hand, some trajectories may not be immediately accessible by using available input force. Another drawback is the excessive consumption of energy due to alternating acceleration and deceleration whenever the ball's tracking is slightly off.



(a) without x controller (b) with x controller

Figure 5.5: Analysis of control strategy in state space

This controller scheme mimics the ball's oscillatory behaviour based on known mathematical model. However, due to uncertainties in the model and input disturbances, additional effort must be used to fulfil the given position command.

5.5.3 Natural oscillations

Another possible approach is by not enforcing any reference in the x coordinate at all. Instead, proposed controller will evaluate the current phase and magnitude of the oscillatory motion based on x, v_x. According to these, it will selects an appropriate control input which will contribute to achieve and maintain steady oscillations.

In this scheme, y coordinate is controlled in the same way as previously, but PID_x is replaced by a controller, whose logic can be described by following steps:

1. Evaluate the direction of motion: $v_{\mathrm{dir}} = \mathrm{sign}(v_x)$.
2. Evaluate the total magnitude of velocity: $\|v\| = \sqrt{v_x^2 + v_y^2}$.
3. Attempt to apply force $F_{\mathrm{des},x} = \begin{cases} v_{\mathrm{dir}} F_{\mathrm{m}} & \text{if} \quad \|v\| < v_{\max}, \\ 0 & \text{otherwise,} \end{cases}$

where $v_{\max}$ is the maximal allowed velocity due to safety reasons and F_{m} is a constant and reasonably small amount of force applied to contribute to oscillations. F_{m} is designed to compensate for the losses due to linear friction in the range of allowed velocities, which means $F_{\mathrm{m}} \leq b v_{\max}$, where b is unknown coefficient of linear friction. Numeric value of F_{m} was assessed experimentally: $F_{\mathrm{m}} = 0.025$ N.

This control strategy can be illustrated using phase portraits. Fig. 5.5a depicts the phase portrait of the system including viscous friction with estimated $b = 0.1$ kg/s and assuming ideal regulation of $y = 0$. The ball decelerates due to friction and eventually stops at a fixed point where $v_x = 0$ and $x \in \langle -R; R \rangle$. On the other hand, by adding the described controller, we change the dynamics of the system as depicted in Fig. 5.5b. If the ball's velocity is low (blue), it is accelerated by the input force. On the contrary, if the velocity is too high (red), the ball is decelerated due to friction. Therefore, the velocity of the ball stabilizes at $v_x \approx \frac{F_{\mathrm{m}}}{b}$. There are no longer any fixed points, however, a stable periodic orbit is added (approximated by green). This trajectory is the limit cycle of the system.

Thanks to utilizing the plant's natural oscillations, the controller demands a significantly lower activation of coils. Moreover, it is more efficient and less sensitive,

when steering the ball from an arbitrary initial condition to steady oscillations. On the other hand, the possibility of direct position control was sacrificed.

5.6 Experiments

First of all, let us examine a response of the ball to initial displacement on the laboratory platform. In this experiment, ball's initial height was measured to be $0.3\,d_\mathrm{c}$, as seen in Fig. 5.6b. Since the surface is neither perfectly horizontally aligned nor completely smooth, the ball gains velocity also in the direction perpendicular to its initial radial direction. As a result, the trajectory is turned at the opposite side of the surface. This phenomenon occurs throughout the experiment. Moreover, ball's motion is slowly damped as depicted in Fig. 5.6a. The lower the velocity of the ball, the greater the effects of the surface's unevenness. Parts of this response were used for verification of the mathematical model in Section 3.7. Here, we use it to illustrate the measured long term behaviour of the uncontrolled plant. Comparing the response with ideal radial oscillations of the mathematical model, it can be seen that an external control of both direction and magnitude of the oscillations is necessary.

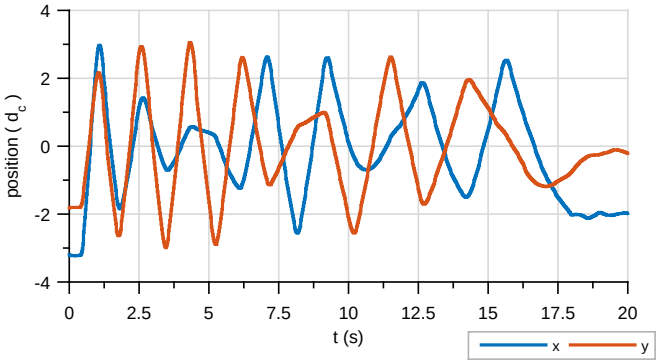
Second set of figures presents measurements of early experiments with control based on reference tracking. Namely, Fig. 5.6c shows the time development, where the ball's initial response is opposite from the command. However, the controller eventually managed to track the reference with only minor displacements in y.

The third row of figures presents yet another set of measurements with reference tracking used. Fig. 5.6e shows lesser steady state error in y regulation in comparison to Fig. 5.6c. Moreover, the system had a better performance in initializing the oscillations. These improvements are contributed to the use of estimation and an additional change in the feedback linearization. We chose not to activate coils that are closer than r_c to the ball's center. The motivation for this change is that both the magnitude and the orientation of the coil's force contribution change significantly with a relatively small changes of the ball's position in this imminent region. Therefore, we do not use coils that are too close to reduce the effect of input disturbances. Note that the error of y regulation right after the start of the experiment caused a notable deviation of the trajectory in Fig. 5.6f. If we aim to extend this concept of control to multiple balls, this phenomenon complicates the initial settling and avoiding collisions between controlled objects.

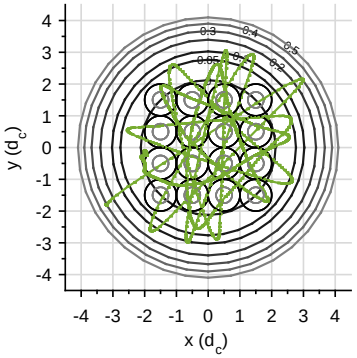
The last row of figures depicts the trajectory obtained by the use of second described control scheme, that is supporting of natural oscillations. At the beginning of experiment, the ball is gradually accelerated into desired oscillations, which become steady at $t \approx 6\,\mathrm{s}$. A more gradual increase in the v_x velocity is notable as the slope of x position in Fig. 5.6g. This control achieves the lowest error in y, with only insignificant changes during the change of motion direction. Resulting trajectory depicted in Fig. 5.6h is the most compact from presented experiments. If such trajectories are obtained for multiple balls, the possible collisions will be prevented with the only exception in surroundings of the surface's center and the initialization phase of oscillations. However, as the control scheme lacks direct control over the ball's position, avoiding such collisions is yet another task to overcome.

5.7 Extension for multiple balls

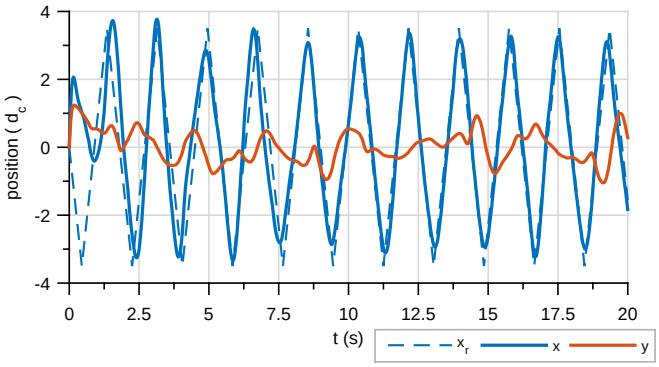
In this section, we describe the steps and experiments conducted in order to extend the control system for multiple balls. Currently, the extension for multiple balls is



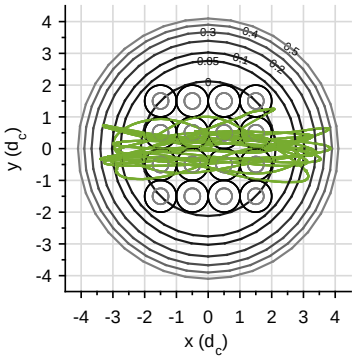
(a) Response to initial displacement



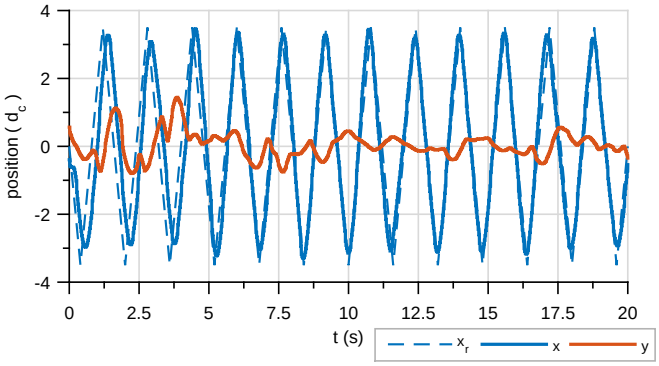
(b) Top view of trajectory (a)



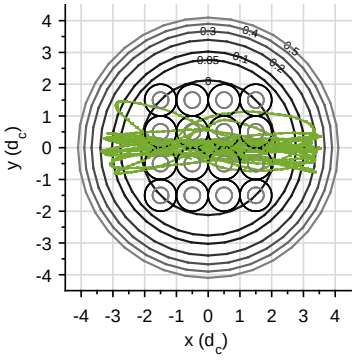
(c) Reference tracking (1)



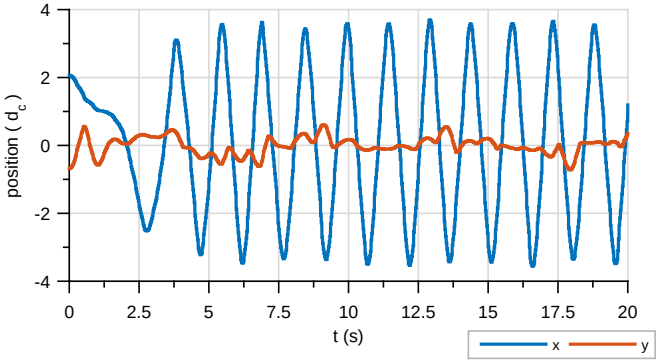
(d) Top view of trajectory (c)



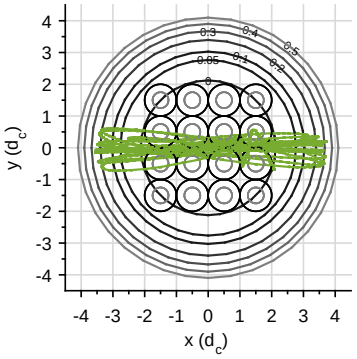
(e) Reference tracking (2)



(f) Top view of trajectory (e)



(g) Oscillations enforcing



(h) Top view of trajectory (g)

Figure 5.6: Experiments with maintaining oscillations of a single ball on the curved surface of the laboratory platform. Measured data.

not finished, however, we made considerable improvements worth documenting.

As a demonstration experiment, we intend to achieve synchronized oscillations of two balls, for instance the first ball oscillating along x axis whereas the second is oscillating along y axis. By synchronized we mean that when the first ball is crossing the center of the platform, the second ball is at the highest point of its trajectory on the curved side of the surface. The phase shift between their oscillations is one quarter of the period. Apart from maintaining steady oscillations, the controller must prevent possible collisions between the two objects. Interestingly, if the two manipulated objects are close, they are attracted to each other, which violates the mathematical model of the input force and complicates the control. In experiments with multiple balls, we chose not to activate only the coils that are closer than $r_\mathrm{c}/5$ to the ball's center, because the presence of multiple balls may be too restricting on the number of coils which may be activated to avoid input disturbances. By doing so, we intend to avoid activating coils whose real force contribution would be significantly different from the computed value. Lastly, use of multiple balls increases the processing time necessary for visual position measurement. Therefore, the sampling frequency of the camera was decreased to $f_{\mathrm{s,cam}} = 35\,\mathrm{Hz}$ to provide sufficient time to finish all computations within each sampling cycle.

Initially, we attempted to extend the concept of reference tracking, which was described in Subsection 5.5.2. It allows for direct control over the ball's position, meaning that we can easily transfer the requirement of synchronization to corresponding reference commands. This scheme was implemented and experimentally tested for two balls of $m_\mathrm{b} = 110\,\mathrm{g}$. The experiment was unsuccessful, which is attributed to two factors. Firstly, it was not possible to avoid mutual collisions during the initial settling of the oscillations. Secondly, the controller was not able to suppress the initial tracking error sufficiently quickly as in experiments with a single ball. As a result, the controller's command was too aggressive due to high tracking error. The command was alternately reaching the saturation limits and eventually destabilized the plant.

Results based on the extension of second control scheme, which was described in Subsection 5.5.3, were slightly superior. However, the controller was only able to successfully maintain oscillations of one ball while simultaneously regulating the other at a constant position. The available force seemed to be insufficient to cover the demands of the controller. Additionally, as this scheme does not allow for direct position control, it was not feasible to avoid collisions of the balls as well as the mutual attractive interaction between them.

Subsequently, we decided to use smaller balls. Originally, the size of the used ball was chosen with respect to the minimal pressure necessary to measure the position using the resistive touch foil. However, as we measure the position visually, the use of smaller balls is not restricted. The smaller ball has radius $r_\mathrm{b} = 10\,\mathrm{mm}$ and mass $m_{\mathrm{b},2} \doteq 39\,\mathrm{g}$. To adjust the controller to the new controlled object, we scaled its control output proportionally to the inverse of masses ratio

$$\mathbf{F}_{\mathrm{des},2} = \mathbf{F}_{\mathrm{des},1} \frac{m_{\mathrm{b},2}}{m_{\mathrm{b},1}}\,, \tag{5.10}$$

where index 1 denotes the original ball and index 2 denotes the lighter replacement. Additionally, the mathematical model of magnetic field was adjusted. The magnetic force was approximately scaled by 0.75 to account for the change of ball's volume and height in magnetic field. Scaling is based on measured data provided by J. Zemánek.

Consequently, the average activation of coils was decreased to approximately 35% during the control of smaller ball's oscillations, improving the performance signifi-

cantly. Additionally, the effect of mutual attractive interaction between two smaller balls was decreased. In conclusion, the platform should be able to provide sufficient force for the control of oscillations of multiple small balls.

Finally, we conducted the experiment of controlling oscillations of two smaller balls using the natural oscillations controller. To provide the phase shift between the oscillations, one of the balls was initially held and released by hand. Although the scheme seems highly effective and achieves steady timing of oscillations for a single ball, we accomplished only several transitions before the balls collided. Without a more direct control over the positions and relative distance between the balls, the collision is inevitable due to mutual interactions, input disturbances caused by lower sampling frequency and imprecision by releasing the second ball manually. Importantly, balls have a different period due to different velocities at the beginning of the experiment. This fact plays a key role and significantly contributes to a collision before the steady oscillations may be possibly reached.

5.7.1 Suggested improvements

In the future work, the control scheme based on natural oscillations may be further extended to provide a better control over positions of multiple balls, yet retaining the advantage of not having a direct position reference command. A cascade controller may be implemented. Currently, we can only exert a constant force in the direction of the velocity. Provided that we have a good estimate of the ball's velocity, P or PD controller in the inner velocity feedback loop will provide similar results and further extend the possibilities of the current implementation. In the outer loop, an intelligent controller may be evaluating the current phase of oscillations and relative positions to avoid possible collisions by setting the reference velocities of both balls, for example accelerate one and simultaneously decelerate the other to avoid a collision. The cascade controller may provide a possibility to decompose the problem, letting the inner loop to maintain the oscillations and outer loop to avoid collisions. This control scheme was neither experimentally tested nor implemented due to insufficient time. It remains as a proposal for future improvements.

5.8 Hardware implementation

During the work on this thesis, we considered the possibilities of implementing the control system using a dedicate hardware. However, in our application the use of dedicated hardware was limited by the available method of position measurement.

From the two suggested hardware options, only Speedgoat rapid prototyping platform offers BitFlow NEON-CLB framegrabber for Camera Link compatible cameras. Simultaneously, it provides adequate computing power to run computer vision algorithms at a sufficient rate. However, as stated in [11], block From Video Device of Image Acquisition Toolbox does not support code generation for external target. Its possible replacement, BitFlow CameraLink block of the xPC Target library, does not support hardware Bayer interpolation, which is in our application necessary for a reasonable timing performance.

The information in [11] may have become outdated since. However, we decided not to inquire into the current state of code generation support of Image Acquisition Toolbox. Instead, we focused our attention on the development of a functioning controller and estimator to fulfil the main obligatory tasks of the thesis.

6 | Conclusion

The major tasks of the thesis were to design and fabricate a non-flat surface for the Magnetic manipulation platform, to derive a mathematical model of the ball's motion dynamics on the surface and to design a feedback controller, which will utilize the curved surface to achieve an interesting and challenging behaviour.

In the thesis, we managed to design and manufacture a curved surface, which allows to reach higher velocity of the controlled object due to improved conservation of the energy in the system. A significant step has been made toward mastering the surface's manufacturing process. We achieved a notably better smoothness of the surface, compared to the previous work. Moreover, the design is properly adapted to the dynamics of a rolling ball, prevents excessive sliding and offers a challenging control task by increasing the performance in terms of velocity.

Subsequently, the dynamic behaviour of the mechanical system, consisting of a rolling ball on the curved surface, was mathematically described using Lagrange d'Alambert equation in the fundamental nonholonomic form. The mathematical model provided useful simulation results, explaining the behaviour of the ball rolling on the non-flat surface. In Section 3.7, the validity of the mathematical model was verified based on the comparison with experimental measurements of platform's response to initial conditions. Additionally, a simplified mathematical model was proposed. Using the simulation results, it was shown in Section 3.9 that both models produce almost identical responses in the initial conditions characteristic for our application. Both models capture important features of the kinematics and dynamics of the experimental platform with a reasonable accuracy.

Presented simplified mathematical model also provided a valuable starting point in the design of feedback controllers. Based on the model, we computed initial values of the PID controller gains. Subsequently, the controller gains were adjusted empirically to compensate for unmodeled friction. Mathematical model was also frequently employed for simulation and verification of new control schemes prior to deployment on the laboratory platform. Moreover, we were able to design two estimators for the real laboratory platform based on the simplified mathematical model. Both of the estimators were successfully deployed during the control task.

Next, we fulfilled the selected demonstration task of maintaining steady oscillations of a single ball along a chosen radial axis. This task posed a challenge due to reaching a higher top velocity of the controlled object and the delay introduced by the vision-based position measurement. Together with the position-sensitive feedback linearization, the task is demanding in the terms of required sampling frequency. The timing limitations of the camera and image transfer were overcome by using a multirate control scheme. We utilized the estimators to extrapolate the position of the ball in between the position measurements. By doing so, we provided a finer-grained entries for the feedback linearization layer and allowed the controller to run effectively at a faster sampling frequency. As a result, the force exerted by

coils appeared to match the requested force more closely, reducing the occurrence of input disturbances and allowing for a more precise steering.

Finally, we attempted to extend the control scheme to maintain oscillations of multiple balls. However, controlling multiple objects at relatively high velocities simultaneously proved to be a problem of much greater complexity. Simple extensions of the control schemes, which are sufficient for a single ball, were unable to cope with the mutual interactions and collision avoidance.

6.1 Future work

To make further improvements in the task of controlling synchronized oscillations of multiple balls, we proposed a possible approach in Subsection 5.7.1. The approach is based on the use of a cascade control scheme. Such control scheme may improve the collision avoidance by providing a more direct control over positions of the balls. Yet it will retain the advantages of the previous control scheme of Subsection 5.5.3 which controls the oscillations based on the ball's velocity only. Additionally, compensation of the camera's delay may be implemented in future to further improve the estimation.

Next, the possibilities of implementing the control system on a dedicated hardware may be once again inquired. Namely, the contemporary state of code generation support of Image Acquisition Toolbox may be surveyed. If supported, it will allow to run the Simulink implementation of the control system using the Speedgoat rapid prototyping platform.

Finally, we may inquire into possibilities of using a high speed camera with a built-in FPGA board. The FPGA camera will allow a significant increase in the frequency of the position measurement not only by running the computer vision algorithms on an FPGA, but also by avoiding the lengthy transfer of the whole image to the PC.

Bibliography

[1] J. Zemánek and Z. Hurák. Experimental modular platform for distributed planar manipulation by shaping magnetic field. *unpublished*, 2015.

[2] J. Filip. Extension of magnetic platform for manipulation on a curved surface. Individual projet, Czech Technical University in Prague, 2015.

[3] S. J. Oliver. Meshmixer 2.0: Best newcomer in a supporting role? Online: `http://www.extrudable.me/2013/12/28/meshmixer-2-0-best-newcomer-in-a-supporting-role/`. [Cited on March 21st 2015].

[4] Ultimaker B.V. Cura user manual. Online: `https://ultimaker.com/download/170/Cura_User-Manual_v1.0.pdf`. [Cited on March 20th 2015].

[5] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer, 2009.

[6] M. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley and Sons, New York, 2006.

[7] B. Siciliano and O. Khabit, editors. *Springer Handbook of Robotics*. Springer, 2008.

[8] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2009.

[9] Z. Qu. *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*. Springer, 2009.

[10] Donald T. Greenwood. *Advanced Dynamics*. Cambridge University Press, 2006.

[11] A. Simonian. Feedback control for planar parallel magnetic manipulation. Master's thesis, Czech Technical University in Prague, May 2014.

[12] V. Skála. Line-torus intersection for ray tracing: Alternative formulations. *WSEAS Transaction on Computers*, July 2013.

[13] P. Corke. *Robotics, Vision and Control*, volume 73 of *Springer Tracts in Advanced Robotics*. Springer, 2011. Chapter Image Formation.

[14] G. F. Franklin, D. Powell, and M. L. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley Longman Publishing Co., Inc., 3rd edition, 1997.

Appendix A

Content of the attached CD

Directory	Content
autocad	Drawings of 3D models in AutoCAD Drawing `.dwg` format.
printable	3D models exported for printing in `.stl` format.
mathematica	Wolfram Mathematica notebooks containing derivation of mathematical model, including generation of Matlab code in .m format.
matlab	Matlab&Simulink implementations of controllers, estimators and mathematical models, `surfaceDesign.m` of Section 2.3.
vision	Utilities for computer vision algorithm configuration, Matlab scripts of Section 4.5, HSV thresholding planes.
images	Annotated RGB frames shown in Section 4.5.
measure	Measurements of experiments conducted on the laboratory platform
visualisation	Matlab scripts and functions used for visualisation.
thesis	Text of the thesis in `.pdf` format.