Design and Development of a Multi-Mode, Two-Wheeled, Self-Balancing Service Platform

MAS500 - Master's Thesis

Spring 2019

Vidar Kittilsland

Sondre Lien

Supervised by:

Morten Ottestad

University of Agder, 2019
Faculty of Engineering and Science
Department of Engineering Sciences

This project is carried out as a part of the education at the University of Agder and is therefore approved as a part of this education. However, this does not imply that the University answers for the methods that are used or the conclusions that are drawn.

Acknowledgements

This master's thesis is carried out as the final part of the master's program in mechatronics at the University Of Agder.

We would like to especially thank our supervisor Morten Ottestad for supporting and guiding us through this project. We would also like to thank him for giving us the opportunity to work with such an interesting solution to a service platform and contributing with ideas for the realization of a prototype design.

We have also recieved some great help from Carl Thomas Duus, Roy Werner Folgerø, Harald Sauvik and the other lab engineers. A well deserved thank you is therefore appropriate for their support with 3D-printing, metal workings like CNC operation and practical suggestions regarding construction.

24/05-2019

Vidar Kittilsland

Vidar Kittilsland

Sondre Lien

Sondre Lien

Abstract

This thesis covers the design and development of a multi-mode, two-wheeled, self-balancing service platform. The purpose of such a platform is to assist people with transportation tasks in crowded spaces like airports by consuming as little space as possible and minimizing its overall footprint. This is solved by constructing a robot that can transition between bicycle and differential drive configurations and keep its balance in both configurations as well as in the transition process between them. A two-wheeled, self-balancing robot is an inherently unstable system that can be based on the theory of an inverted pendulum. The unstable system is stabilized using a sophisticated state space control approach that utilizes the mathematical models of the different dynamic systems of the control modes to provide full state feedback.

The mechanical structure is designed and built mainly from steel profiles, 3D-printed parts, and laser cut parts to satisfy the robot's requirements to rigidity and mechanical properties. A custom made battery pack is also designed to minimize the weight of the robot while simultaneously enabling the robot to run for a desired amount of time without recharging.

The result is a robot that balances satisfyingly in both bicycle and differential drive configurations and presents a proof-of-concept prototype that shows that products such as this robot are highly feasible. Due to time limitations and limited access to proper equipment, some theory in this thesis was not executed in practice but is considered as potential further work on the service platform.

For video of the resulting system, please see the URL below.
https://youtu.be/KidiPGkNSt0

Keywords: Machine design, Control design, Simulation, MATLAB, Simulink, LabVIEW, SOLIDWORKS, State-space, Stabilization, Full State-feedback, Kinematics

Contents

List of Figures

List of Tables

1 Introduction

The use of robots and robotic equipment is increasing every day and is becoming an important element of everyday life [9]. New robots continue to make their entry to provide newer and better solutions to mundane challenges that arise. An important aspect is that the robots should not interfere with their environment beyond their respective tasks. Hence, the overall footprint of the robots should always be as low as possible.

This report describes the design and development of a two-wheeled multi-purpose robot that minimizes its footprint by being able to keep its balance in three different control modes; as a high-velocity bicycle robot, a low-velocity bicycle robot, and a differential drive robot. These balancing modes are illustrated in Figures 1, 2 and 3 respectively.

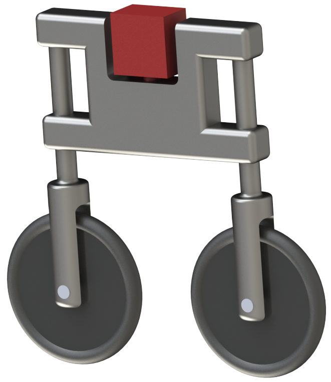


Figure 1: Bicycle, still to slow

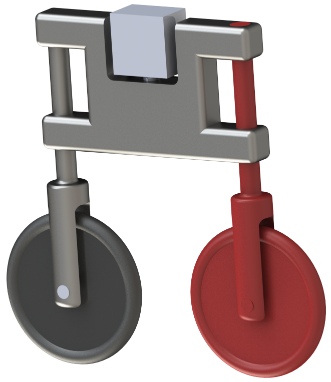


Figure 2: Bicycle, fast

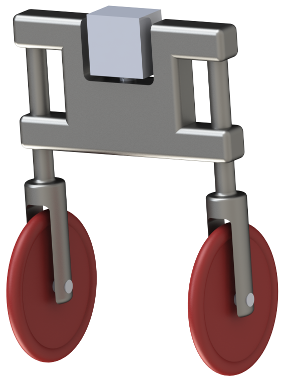


Figure 3: Differential drive

Note that these figures are only sketches made to illustrate the different control modes, and do not represent the final mechanical design. The parts in red represent the active parts used to keep the robot stable; The slow bicycle uses an independent balancing mechanism, the fast bicycle balances by altering the steering angle of its front wheel while moving with a certain velocity, and the differential drive utilizes its two wheels to keep its balance similar to a segway. In all modes, the dynamic systems are inherently unstable and require sophisticated control structures to compensate for the unstable poles of the system.

The main advantage of having these separate modes is that the robot would inherit the high maneuverability of the differential drive robot, and the speed possibilities of the bicycle while minimizing the space consumption in tight areas. This speed and maneuverability combination could, for instance, be an advantage at crowded places like airports.

1.1 Motivation

A small footprint service robot that can transport people or items with both high speed and maneuverability has the potential to increase efficiency especially in large places where the frequency of dynamic obstacles is high, like the previously stated example; airports. Such a design that combines a fully self-balancing bicycle and differential drive would push the limits of what a service robot can do and therefore increase the allowed workplace conditions.

1.2 Thesis Statement

The thesis statement includes four main features that should be included in the service robot.

1. The robot should be self-balancing in differential drive mode.
2. The robot should be self-balancing in bicycle mode.
3. The robot should be able to transform between the two configurations while keeping the self balancing feature.
4. The robot should be able to be controlled remotely.

It is also specified that this robot is built to explore the feasibility of such a service platform, and to present a ”Proof-of-Concept Prototype”. However, the relatively unexplored design and control of a transforming self-balancing bicycle consumed more time than estimated, so the low-speed control of both configurations which are paramount for the transitioning between the configurations are focused on. This results in less focus on the bicycle control for fast movements. This is also partly due to the necessity of using a sort of treadmill to ensure the safety of both the service robot and people around it during the integration and function testing at high speeds. Since there was no time to set up such a device, it can be part of the future development. The theory covering the kinematics and dynamic models for fast bicycle control is, however, included in the report.

1.3 Project Management

Throughout the thesis period, a well-organized schedule is followed to be able to finish the different tasks and meet the requirements set in advance. As mechatronics engineers, the V-model illustrated in Figure 4, represents an ideal development process where the different tasks are iteratively built and improved.

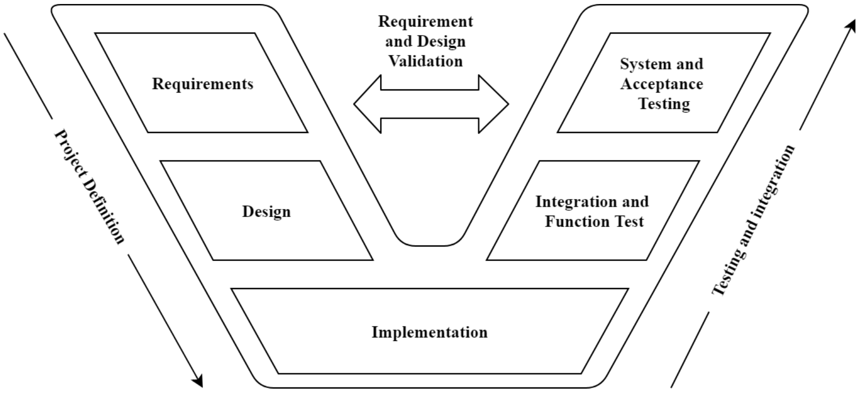

Requirement and Design Validation
Requirements
System and Acceptance Testing
Design
Integration and Function Test
Implementation
Project Definition
Testing and integration

Figure 4: V-model

A design is built based on system requirements and implemented thereafter. The design is then integrated, tested, and re-evaluated if changes are needed to meet the requirements. During such a process, a re-design is done multiple times until the results are satisfactory. A Gantt diagram, illustrated in Figure 5, is also set up before the thesis period with rough estimates of the time duration of the different tasks.

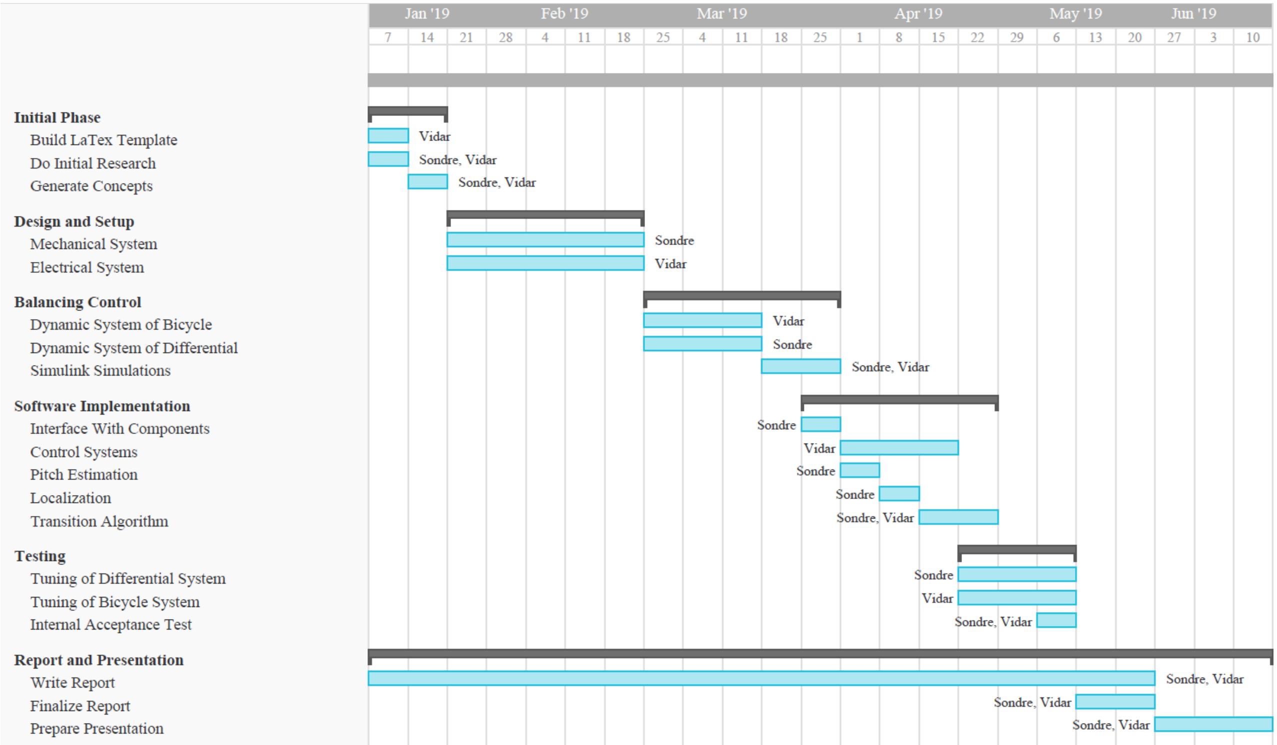


Figure 5: Gantt Diagram

2 System Description

The service platform is designed and built from various parts and components in order to meet the system requirements. This chapter will go into detail on why and how these different components are implemented on the robot.

2.1 Concept

Since this thesis does not revolve around product development but the design of a proof-of-concept prototype, a brief concept selection is incorporated in this report opposed to an elaborate and extensive process. The main requirements of the concept selection are to generate a concept that can easily transform between bicycle and differential locomotion, and build a system that includes all the necessary properties needed for it to keep its balance in both configurations. See Figure 6.

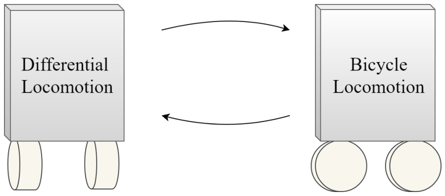


Figure 6: Locomotion Configurations

2.1.1 Concept Generation

Figure 7 presents a concept chart that shows different elements that the main concept can be build from. The elements are set up based on the knowledge of the group participants on how to solve the problems within each topic.

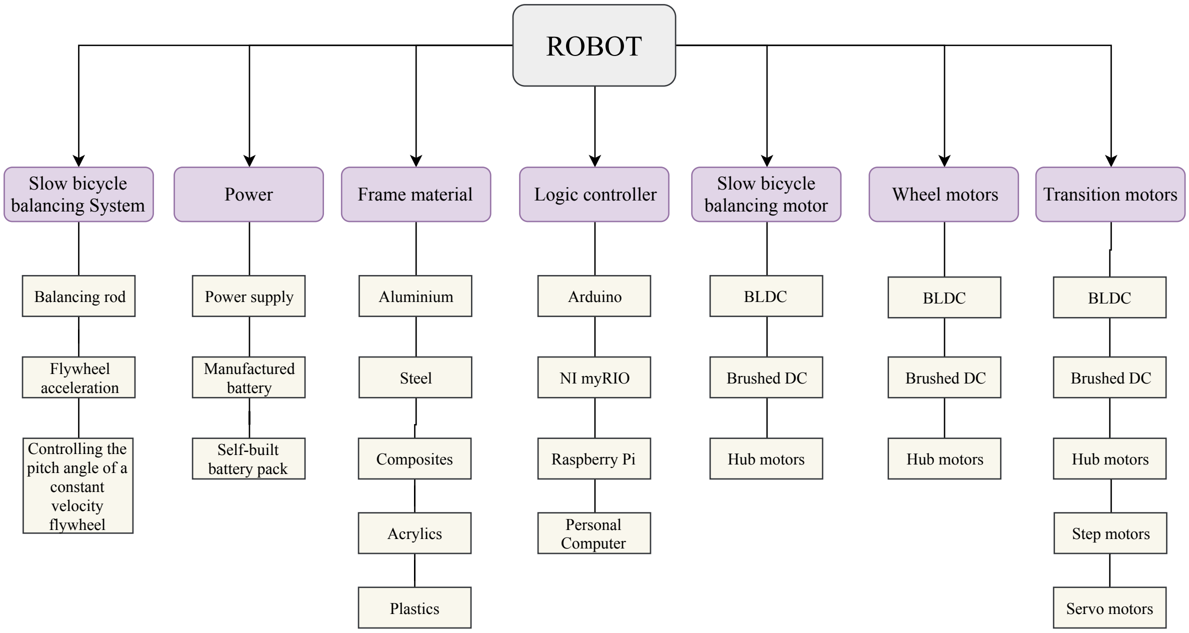


Figure 7: Concept Chart

Where the transition motors ensure the transformation between the two locomotion configurations, as well as controlling the steering angles in the bicycle configurations. Figures 8, 9 and 10 show the working principles of the different bicycle balancing systems from Figure 7.

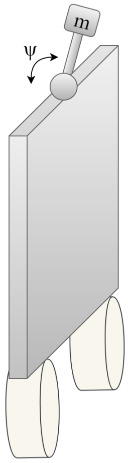


Figure 8: Balancing Rod

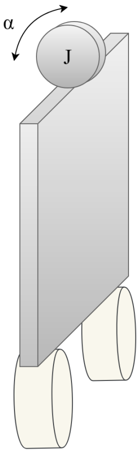


Figure 9: Inertial Acceleration

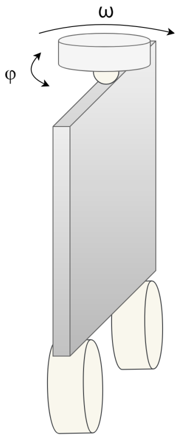


Figure 10: Flywheel Angle

The balancing rod stabilizes the robot by controlling the angle, ψ, of a rod with a certain mass, m, using a single motor. Inertial acceleration utilizes acceleration, α, of a motorized inertia wheel, J, to counter the moment of the robot's angular displacement. And flywheel angle control benefit from the principles of conservation of angular momentum by altering the angle, ϕ, of a flywheel rotating with a constant velocity, ω.

2.1.2 Concept Selection

The selections listed in Table 1 of conceptual elements were done to obtain the main robot concept:

Table 1: Selected concept elements

Topic	**Element**	**Justification**
Slow bicycle balancing system	Inertial acceleration	Only requires one motor, and low space consumption
Power	Self-built battery pack	The battery can be designed to fit the robot with desired weight, size, current, charge and voltage
Material	Steel	Inexpensive, availability, easily welded and rigid
Logic controller	myRIO	Previous experience by the project participants, Many I/O and easy to troubleshoot
Slow bicyle balancing motor	BLDC motor	High efficiency and torque/size ratio
Wheel motor	Hub motors	Motors within the wheels reduce the complexity of the system
Transition motors	Servo motors	Compact and easily controllable

2.2 Electrical Design

2.2.1 Components

The selected components to be implemented on the robot are listed in Table 2.

Table 2: Selected components

Topic	Component	Qty	Description
Battery cells	Li-ion cells	18	Connectable by nickel strips
Logic controller	NI myRIO-1900	1	Embedded controller
Inertia motor	Maytech MTO6374-HA	1	BLDC motor, 190 kv
Hub motors	FLJ 10in	2	BLDC hub motor
BLDC controller	VESC	3	Universal BLDC controller
Servo motors	Dynamixel MX-64	2	High powered smart actuator
Inertial measurement unit	MPU9250	1	Gyroscope and accelerometer
Encoders	HN3806-AB-600N	2	2400 quad. pulses per revolution
Battery management system	JBD-HP8S001	1	6S - 3,7V - 60A
Step-down converter	OSKJ Buck converter	1	DC 7-32V to 0.8-28V

Note that a step-down converter is added to the component list to provide a 14.0V power supply to the MyRIO and Dynamixel servo motors.

2.2.2 Battery

The battery is built from Li-ion cells and embedded in a 3D printed protective casing together with a battery management system to ensure uniform charge and discharge for all the cells. These cells are connected through point welding of nickel strips to fulfill the electrical requirements of the system. Detailed calculations surrounding these requirements are presented later in the report. The battery supplies the different components such as a MyRIO logic controller, brushless DC motors, and inertial measurement unit with power. Figures 11 and 12 show the designed battery embedded in the protective shielding.

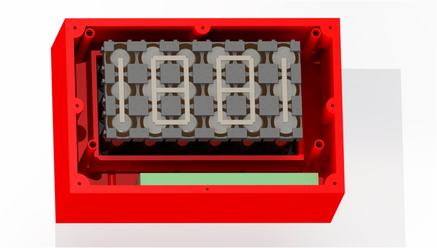


Figure 11: Battery assembly, including the BMS in green

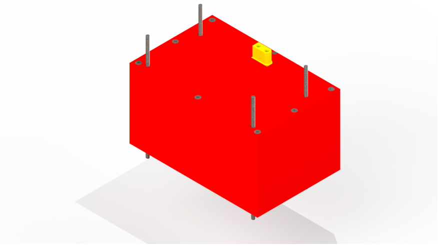


Figure 12: Finished Battery package

2.2.3 Wiring Schematics

Figure 13 shows the complete wiring schematics of the electrical components implemented on the robot.

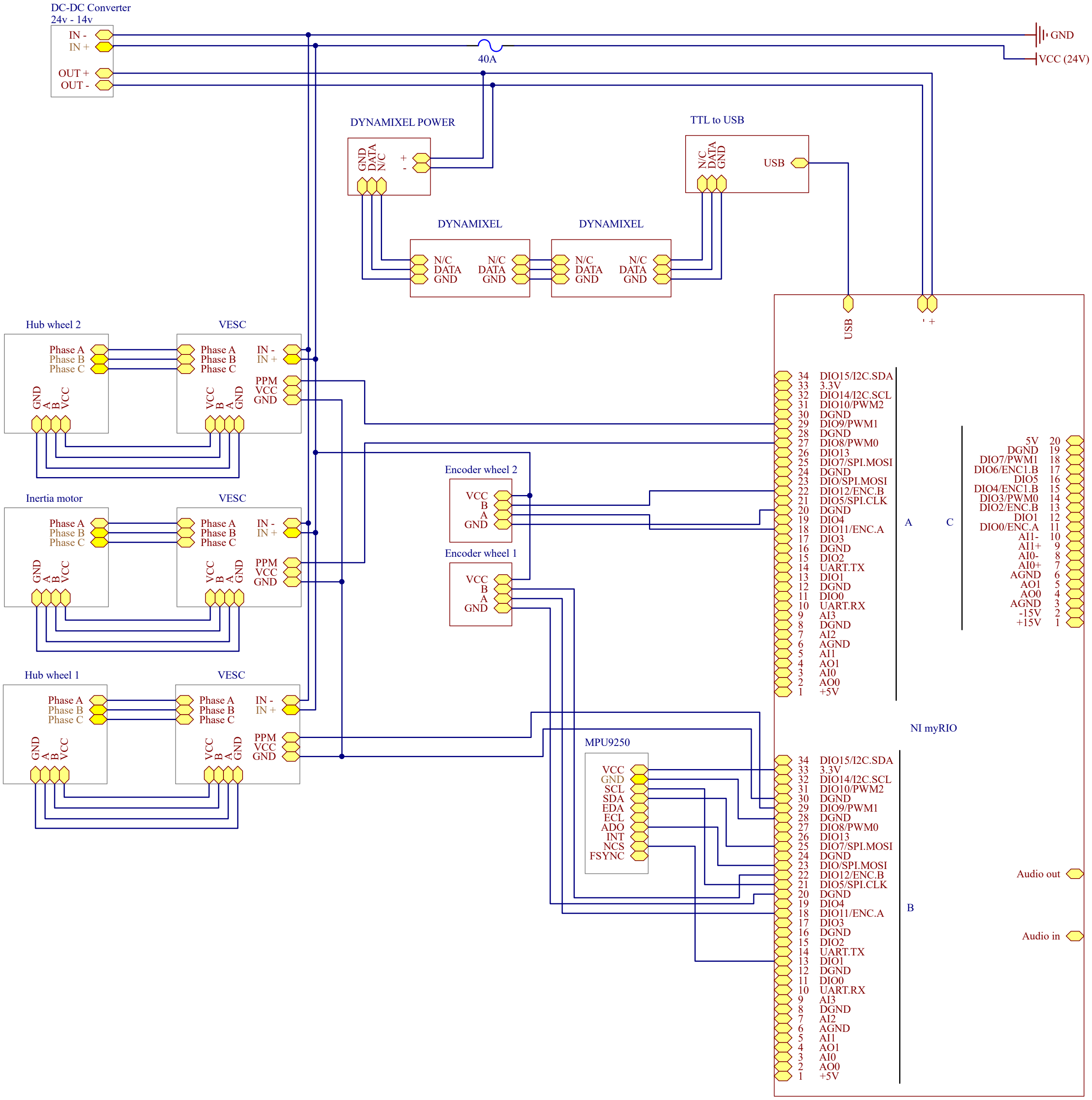


Figure 13: Wiring schematics

2.3 Mechanical Design

In order to obtain the robot's desired properties for a self-balancing service robot, a suitable mechanical design is made. It is desired that the weight of the robot is as low as possible, while maintaining a rigid system with a high center of mass. A simple and robust design where the different parts can easily be replaced is important considering that the mechanical system is to be built by the group participants who do not possess certificates of apprenticeship within mechanical disciplines. All parts are modeled, assembled and made drawings of in *SOLIDWORKS* and technical drawings can be found in Appendix A. Figure 14 and 15 illustrate the final design with the bicycle and differential drive configurations respectively.

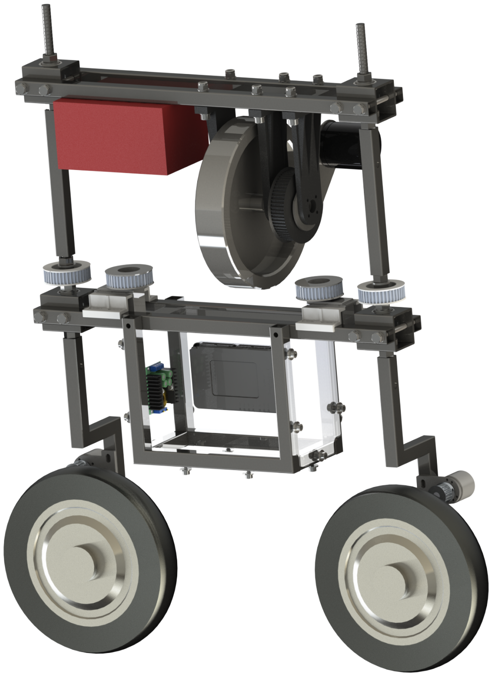


Figure 14: Bicycle configuration

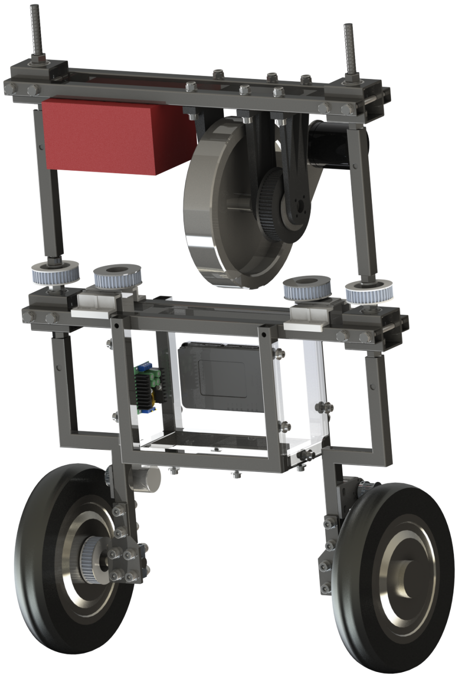


Figure 15: Differential drive configuration

The upcoming subchapters will elaborate on the sub-assemblies of the system in detail, and further describe their functionality. Calculations done in conjunction with the different components are presented later in the report.

2.3.1 Base Frame

The frame to which all actuators and sub-assemblies are attached is named the *base frame*. The base frame is designed such that the legs which the wheels are attached to can rotate independently of each other, and other components can easily be mounted onto the framework. All machined parts are produced in thin steel profiles and plates in order to provide a rigid body and to minimize the mass of the base frame. Figures 16 and 17 show the designed base frame in solid and transparent form respectively.

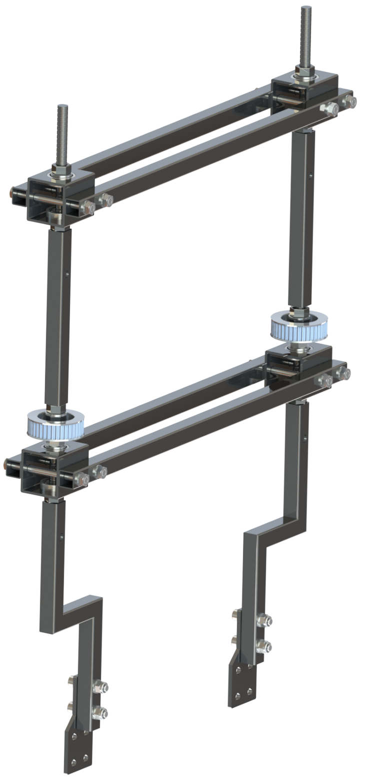


Figure 16: Base Frame

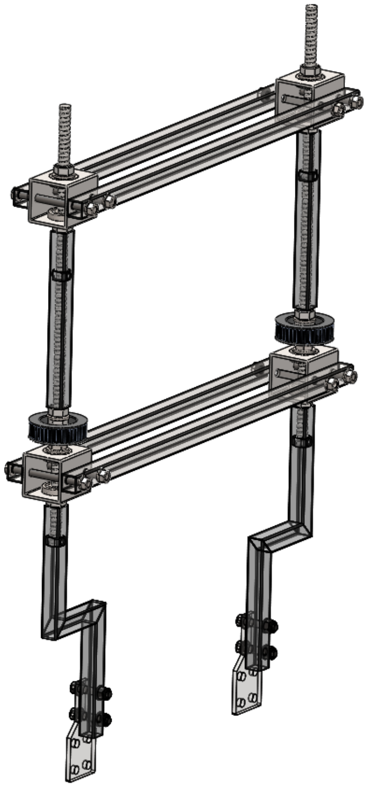


Figure 17: Transparent Base Frame

Figure 18 depicts a closer look on one of the revolute joints in the base frame. In the illustration, the rotating parts are highlighted in red, while the fixed parts are highlighted in green.

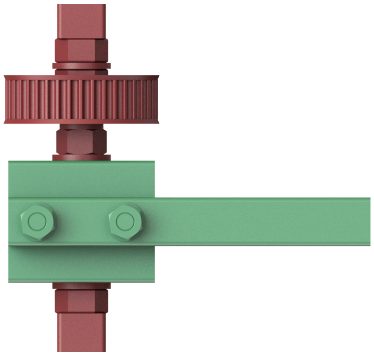


Figure 18: Rotating and fixed parts

As illustrated in Figures 16 and 17 the legs are angled because it is desired that the point of rotation of the hub wheels is directly below the rotational axis of the legs themselves. Due to their rigidity and relatively low weight, square hollow sections (SHS) are integral parts of the base frame design. The base frame is assembled using multiple bolts, washers, and nuts, in addition to two M12 threaded rods. The legs are mounted to the threaded rod by inserting a machined M12 nut into the SHS' and welding it tight through an eight-millimeter hole through the SHS. This design feature can be partially seen in Figure 17, and is further illustrated by Figures 19 through 22.

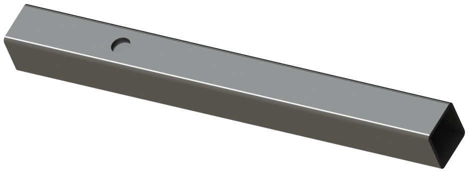


Figure 19: Upper leg

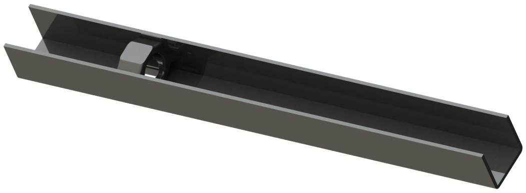


Figure 20: Section view of upper leg

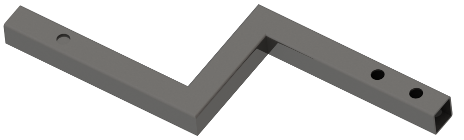


Figure 21: Lower leg

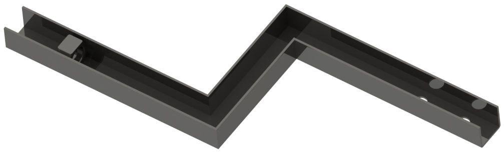


Figure 22: Section view of lower leg

The main reason for the inserted nut is to center the threaded rods according to the SHS'.

Figure 23 illustrates how the revolute joint is designed to allow rotation for the transition between bicycle and differential drive configurations. It consists of an SHS with two SKF 6201 ball bearings attached to it. Due to the low contact areas between the SHS and the bearings, the parts are fixed together using steel epoxy instead of interference fits.

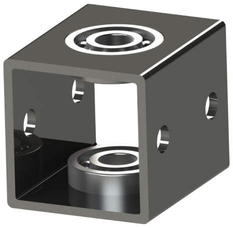


Figure 23: Revolute joint

Figures 24 to 27 illustrate the remaining parts of the base frame.

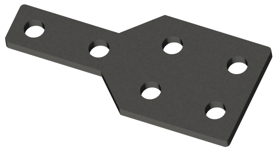


Figure 24: Wheel Bracket

Figure 25: Mounting beam

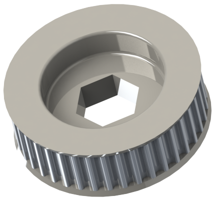


Figure 26: Belt pulley

Figure 27: Threaded rod

The wheel bracket from Figure 24 is designed based on the stock mounting block which is shipped together with the wheel itself. The mounting beam from Figure 25 is made to easily mount further electrical and mechanical components onto the base frame. And finally, the belt pulley illustrated in Figure 26 is designed to easily be mounted onto the threaded rod which is illustrated in Figure 27. This is done by including a hole that fits a regular DIN 934 nut, which opens up the possibility to screw the belt pulley onto the threaded rod and consequently centers the pulley according to the rod's center axis. The belt pulley is designed to be 3D printed.

2.3.2 Component Housing

The purpose of the component housing is to protect and store components such as the myRIO and communication modules.

Figure 28 shows the fully assembled component housing. Similarly to the base frame, it is mainly built from SHS'. In addition, custom laser cut plastic plates are added as walls to provide large areas for fastening of the different components. Since the plates do not carry heavy loads, threaded holes are utilized in the SHS' according to the holes in the plates in order to fasten the plates onto the SHS framework.

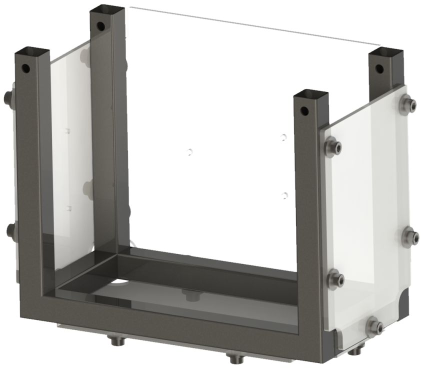


Figure 28: Component housing

2.3.3 Inertia Wheel System

The inertia wheel assembly is designed to safely accelerate the inertia wheel with minimal friction, vibrations and radial stresses on the motor. Figure 29 and 30 show the designed setup of the inertia wheel system in both assembled and exploded view.

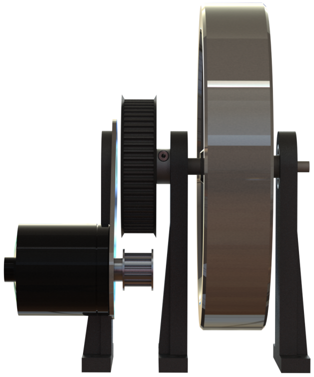


Figure 29: Assembled inertia wheel system

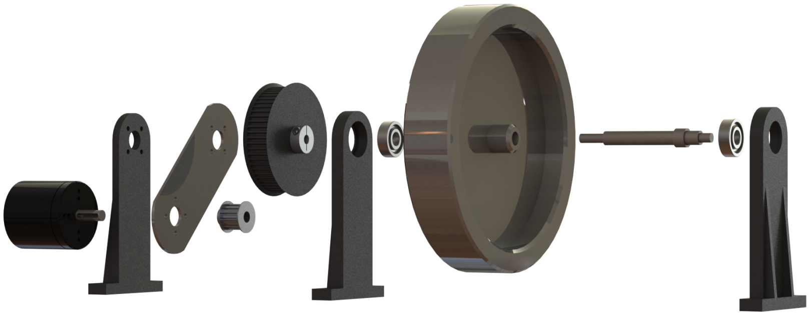


Figure 30: Exploded view of the inertia wheel system

Two brackets with ball bearings are placed on both sides of the inertia wheel to take up both static and dynamic radial forces. The brackets are made with a high precision 3D-printer and printed with a composite material of onyx plastic and kevlar which gives them a high strength to weight ratio [10] [11]. The kevlar is inserted into the brackets as shown below in Figure 31.

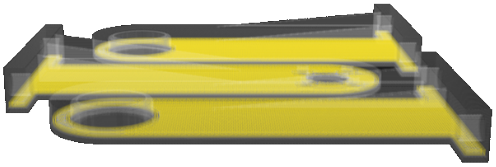


Figure 31: Kevlar layers shown as yellow inside the brackets

The inertia wheel is designed to yield a high moment of inertia to mass ratio in order to reduce weight yet providing relatively high torque. The inertia wheel is made in the CNC mill, and the shaft is made in the lathe machine with the help of lab engineers at UiA. A belt drive is used to obtain a gear ratio between the load and the motor. This results in a higher efficiency of the motor by avoiding stall torques yet allowing for low inertia wheel velocities as seen from a typical BLDC efficiency curve, as shown in Appendix E.1. A low inertia wheel velocity is desired for both easier control and safety.

Interference fits are used to assemble the bearings with the bearing brackets and the shaft. In order to fix the inertia wheel to the shaft, a tight clearance fit is used for precision before a Loctite glue is applied to fix them permanently together. This is advice received from the lab engineers partially due to more easy assembly with the available machinery. This is all done as precisely as possible to get all rotating parts relatively balanced and to avoid vibrations that could potentially yield unwanted noise in components like the IMU.

2.3.4 Other Parts

The remaining parts of the mechanical structure are designed to mount the different electrical components onto the framework, and to transfer rotational displacement. Figures 32 to 35 illustrate these different 3D printed parts.

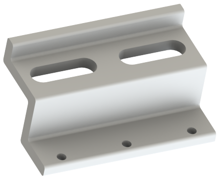


Figure 32: Servo Bracket

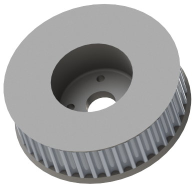


Figure 33: Servo belt pulley

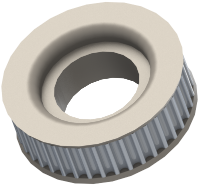


Figure 34: Wheel Pulley

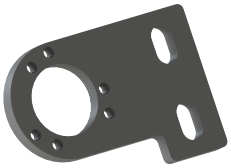


Figure 35: Encoder Bracket

The servo bracket from Figure 32 is used to mount the servo motors with a desired distance from the belt pulley of the base frame. On the servo motors themselves, the servo belt pulley in Figure 33 is designed to transfer the angular displacements of the servo motors to the base frame. The wheel pulley in Figure 34 is designed to fit directly onto the hub wheels, and has the task of transferring the motion of the hub wheels to the rotary encoder which is mounted onto the framework using the encoder bracket from Figure 35.

2.4 Robot Coordinate System

The general local coordinate system of the robot is illustrated in Figure 36. This coordinate system is fixed independently of which configuration is active. The roll, pitch, and yaw are defined according to the defined coordinate system as the rotation around the x-, y-, and z-coordinates respectively. Hence, when referring to the robot's pitch, it implies the angular displacement around the y-axis.

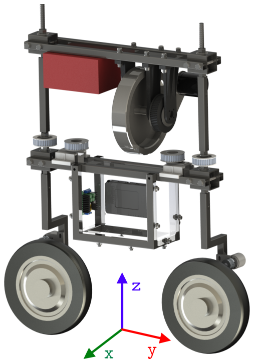


Figure 36: Robot Coordinate system

3 Theory

This chapter covers all the theory relevant to the thesis.

3.1 Power Supply

The selected power supply is as previously mentioned a battery that supplies the whole system with power to operate. Therefore it is important that the battery provides a correct voltage for the system, which is chosen to be 24 volts since the three chosen BLDC motors work well around that voltage. And to be conservative, a maximum continuous current of 40A is set as a requirement for the power supply. A third desired factor is that the battery must contain enough capacity to give a certain amount of continuous operating time.

In order to find the approximate capacity (C) needed for a certain driving distance (L) for the robot, its average consumption of energy is approximated. Initially, it is known that the electrical energy supplied from the battery, (E_{el}), must be equal to the mechanical energy of the system, (E_{mec}), divided by the efficiency of the system (μ) as seen in Equation 1.

$$E_{el} = \frac{E_{mec}}{\mu} \tag{1}$$

Where μ is the total efficiency of the system and contains all loss of both electrical and mechanical energy. This is mathematically described as in Equation 2, but is assumed to be approximately ≈ 0.8 for simplicity reasons and a lack of available data on the different components.

$$\mu = \mu_{drives} \cdot \mu_{motors} \cdot \mu_{gears} \cdot \mu_{misc} \tag{2}$$

The equation for electrical energy is shown in Equation 3 where the multiplication of 3.6 is to get the charge in [mAh], and the mechanical energy equation is shown in Equation 4. These two equations are then substituted into Equation 1 and solved for the capacity (C) as shown in Equations 5 and 6.

$$E_{el} = V \cdot C \cdot 3.6 \tag{3}$$

$$E_{mec} = P \cdot t = T \cdot \omega \cdot t = T \cdot \theta = T \cdot \frac{v}{r} \cdot t = T \cdot \frac{L}{r} \tag{4}$$

$$C = \frac{E_{mec,driving} + E_{mec,balancing} + E_{mec,steering}}{\mu \cdot V \cdot 3.6} \tag{5}$$

$$C = \frac{T_{dr} \cdot \frac{L}{r} + T_{bal} \cdot \theta_{bal} + T_{st} \cdot \theta_{st}}{\mu \cdot V \cdot 3.6} \tag{6}$$

However, only a rough estimation of the energy consumption is needed to minimize size, weight, and the number of charges per day. The energy consumed from steering and balancing should ideally be quite small compared to the driving, and is therefore neglected. This simplification is shown in Equation 7.

$$C = \frac{T_{dr} \cdot \frac{L}{r}}{\mu \cdot V \cdot 3.6} \tag{7}$$

It is desired to be able to drive with an average acceleration of $1.0m/s^2$, which yields an average driving torque of $T_{dr} \approx 5Nm$ based on the calculations done in MATLAB for the system as shown in Appendix D. The desired driving length is set to $L \approx 12500m$, and the radius of the wheels is $r \approx 0.125m$. This desired situation yields a required capacity as calculated in Equation 8 and represents roughly how much capacity the system consumes.

$$C = \frac{5 \cdot \frac{12500}{0.125}}{0.8 \cdot 24 \cdot 3.6} = 7234mAh \tag{8}$$

Because of the neglected energy consuming factors like steering and balancing, an additional 20% is added which yields a total capacity of $C = 8680mAh$.

3.2 Motor I/O

This chapter describes the theory behind the inputs and outputs of the motor systems.

3.2.1 Pulse-Width Modulation

Pulse-Width Modulation (PWM) is a common way to control the behavior of actuators by using a separate motor controller that converts the PWM signal into a certain voltage. Figure 37 illustrates the working principles of PWM.

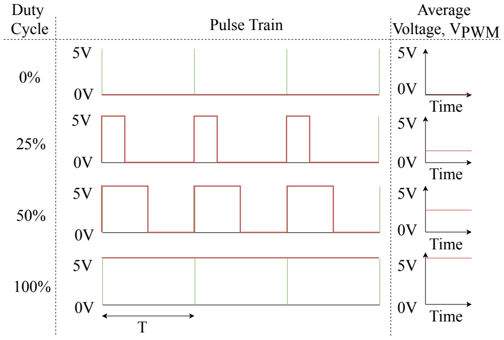


Figure 37: PWM

The signal consists of two different states; on-state with the input signal being for instance five volts, and the off-state with the input signal being zero volts. This is a digital control where the average on-time is used to generate an artificial analog signal. For example a 30% on-time of the period, T, yields a voltage, V_{PWM} of:

$$V_{PWM} = 0.3 \cdot 5V = 1.5V \tag{9}$$

3.2.2 Optical Quadrature Encoders

The outputs of the motors are encoder pulses generated by optical sensors that can be used to determine the rotational displacement of the motors and thereby also the velocity of the motors.

Figure 38 illustrates the two phases, A and B, of the quadrature encoders and how the direction of the motors can be determined. When a change in either phase A or B is perceived, a pulse signal is generated which implies a certain angular displacement corresponding to the resolution of a 90° phase shift, i.e. the angle between two pulses. Using this information, the angular displacement, θ, can be calculated with a given resolution of k_r degrees per pulse and N pulses.

$$\theta = k_r \cdot N \tag{10}$$

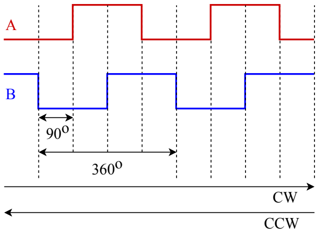


Figure 38: Encoders

3.3 Pitch Estimation

The pitch of the robot describes the angular offset of vertical positioning. A tri-axial inertial measurement unit (IMU) is used to determine the pitch angle by utilizing its data of acceleration and angular velocity from the integrated accelerometer and gyroscope, respectively.

3.3.1 Data Processing

The measurements made from the IMU in terms of acceleration and angular velocity data must be processed in order to provide the proper measurements of the pitch angle. The IMU is positioned as illustrated in Figure 39.

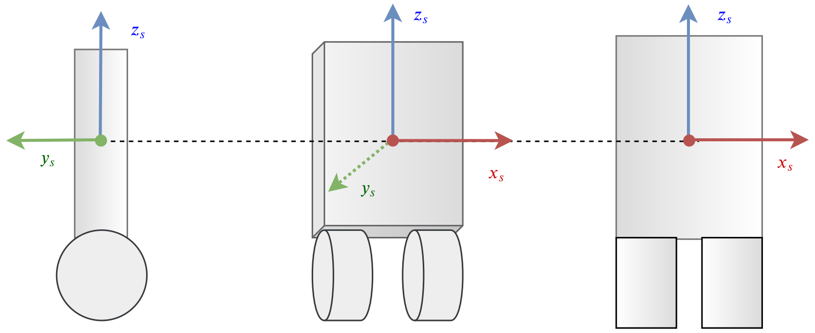


Figure 39: IMU coordinate system

The pitch angle, θ, of the robot corresponds to the roll angle, ϕ' of the IMU; around the x-axis. The robot pitch can be calculated from the accelerometer data by multiplying the rotational matrices of the three axes, and a vector describing the subjection of gravity in z-direction. [12]

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = R_x \, R_y \, R_z \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi' & \sin\phi' \\ 0 & -\sin\phi' & \cos\phi' \end{bmatrix} \begin{bmatrix} \cos\theta' & 0 & \sin\theta' \\ 0 & 1 & 0 \\ -\sin\theta' & 0 & \cos\theta' \end{bmatrix} \begin{bmatrix} \cos\psi' & \sin\psi' & 0 \\ -\sin\psi' & \cos\psi' & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\sin\theta' \\ \cos\theta' \cdot \sin\phi' \\ \cos\phi' \cdot \cos\theta' \end{bmatrix} \tag{12}$$

Where:

ϕ' = Rotation around the x-axis
θ' = Rotation around the y-axis
ψ' = Rotation around the z-axis

$$\frac{a_y}{a_z} = \frac{cos\theta' \, \sin\phi'}{\cos\phi' \, \cos\theta'} = \tan\phi' \quad \Rightarrow \quad \phi' = \tan^{-1}\left(\frac{a_y}{a_z}\right) \tag{13}$$

Equation 13 shows the relationship between the roll angle of the IMU and the accelerometer data. When it comes to the gyroscope, the output values of the sensor are angular velocities around the three axes. therefore, the angular velocity around the IMU's x-axis can be integrated to obtain an estimate of the robot pitch.

3.3.2 Complementary Filter

The complementary filter is a simple, but robust, filter that combines low-frequency data from the accelerometer with high-frequency data from the gyroscope. This is due to the high-frequency noise in the accelerometer data caused by vibrations, and low-frequency noise in the gyroscope that is generated through accumulation of gyroscope bias from integration. Figure 40 shows a block diagram of a complementary filter. [13]

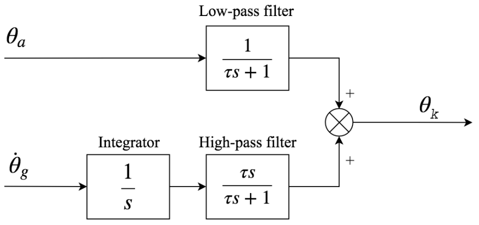


Figure 40: Continuous complementary filter

$$\theta_k = \frac{1}{1+\tau\, s}\,\theta_a + \frac{\tau\, s}{1+\tau\, s}\,\frac{1}{s}\,\dot{\theta}_g = \frac{\theta_a + \tau\,\dot{\theta}_g}{1+\tau\, s} \tag{14}$$

Where:

θ_k = Filtered pitch angle
θ_a = Estimated pitch angle from accelerometer
$\dot{\theta}_g$ = Angular pitch velocity
τ = Time constant

Discretizing through backward Euler approximation [14] of the system from Equation 14 yields:

$$1+\tau\, s = 1 + \frac{\tau - \tau\, z^{-1}}{\Delta t} \quad\Rightarrow\quad \theta_k\,(1+\frac{\tau-\tau\, z^{-1}}{\Delta t}) = \theta_a + \tau\,\dot{\theta}_g \tag{15}$$

$$\Downarrow$$

$$\theta_k\,(1+\frac{\tau}{\Delta t}) = \theta_a + \tau\,\dot{\theta}_g + \frac{\tau}{\Delta t}\theta_{k-1} \tag{16}$$

$$\Downarrow$$

$$\theta_k = \frac{\Delta t}{\Delta t+\tau}\,\theta_a + \frac{\tau\Delta t}{\Delta t+\tau}\,\dot{\theta}_g + \frac{\tau}{\Delta t+\tau}\theta_{k-1} \tag{17}$$

Where θ_k corresponds to the current pitch angle, while θ_{k-1} is the filtered angle of the previous iteration. A constant α is introduced as:

$$\alpha = \frac{\tau}{\Delta t+\tau} \quad\Rightarrow\quad 1-\alpha = \frac{\Delta t}{\Delta t+\tau} \tag{18}$$

α is substituted into Equation 17, and the resulting discretized complementary filter is:

$$\theta_k = (1-\alpha)\,\theta_a + \alpha\,(\dot{\theta}_g\,\Delta t + \theta_{k-1}) \tag{19}$$

Which is illustrated by block diagram in Figure 41.

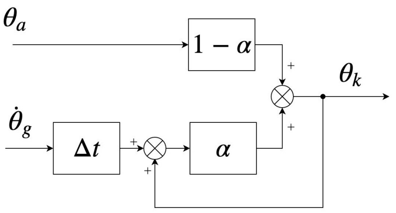


Figure 41: Discrete complementary filter

3.4 Kinematics and Odometry

The kinematics of the system describe the relationship between the global and local motion of the robot without considering forces. These relationships are derived by analyzing the two separate locomotion configurations individually. These two systems are illustrated in Figures 43 and 42, where the wheels are displayed in black, the gray arrow indicates the intended forward direction of the robot, and the red and green arrows indicate the robot's local x- and y-axes respectively.

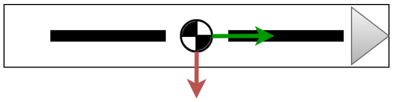


Figure 42: Bicycle Representation

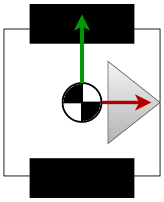


Figure 43: Differential Representation

For the two locomotion configurations illustrated in Figures 42 and 43 there is one kinematic model for the differential drive and two different possible kinematic models for the bicycle; dual wheel steering where both wheels are turned the same angle in opposite directions, and the conventional single wheel steering where one of the wheels (traditionally the front wheel) turns a given angle while the other one's angle is fixed. For all these kinematic models it is assumed that the center of mass is in the middle of both wheels.

A subject that is important in terms of localization and has direct relations to the kinematics is the robot's odometry. Odometry is a robot's ability to estimate its current global position and orientation based on sensor data. The most common way to determine the odometry of a robot is to utilize encoder data from the wheels together with its kinematics and dead reckoning. Dead reckoning estimates the current position and orientation based on data from the previous iteration and adds the measured difference as shown in Equation 20.

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \Delta\mathbf{p} \tag{20}$$

$\mathbf{p}_k$ = Current pose
$\mathbf{p}_{k-1}$ = Previous pose
$\Delta\mathbf{p}$ = Difference in pose

The following nomenclature generally applies to this chapter:

S = Distance traveled
r = Wheel radius
b = Distance between wheels
α = Steering angle

ψ = Angular displacement of robot
v = Robot translational velocity
θ = Angular displacement of wheel
ω = Robot rotational velocity

The encoder odometry is included in this thesis due to its close relationship with the kinematics, so that it can be used for localization purposes in further work.

3.4.1 Dual Wheel Bicycle Steering

This chapter will revolve around the derivation of the kinematics and odometry concerning the dual wheel bicycle steering.

Kinematics

In Figure 44 the dual wheel steering setting is illustrated with its related instantaneous center of rotation (ICR). This is the kinematic model that is implemented on the bicycle when balancing with the inertia wheel due to its high maneuverability. The parameters in Figure 44 are utilized to solve the kinematics of the system.

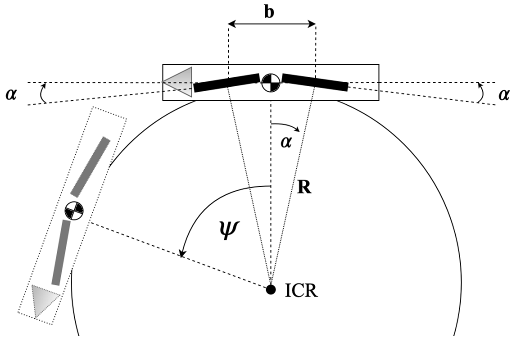


Figure 44: Dual steering bicycle kinematics

Since both wheels are turned equally according to the periphery of a circle with radius R and center in the robot's ICR, the displacement of the robot will correspond to the peripheral displacement;

$$S = \psi \cdot R \tag{21}$$

Since $\theta_f = \theta_r = \theta$ for dual wheel steering where f and r indicate the front and rear wheels respectively, the distance traveled can also be expressed as:

$$S = \theta \cdot r \tag{22}$$

Which by substitution yields:

$$\psi \cdot R = \theta \cdot r \quad \Rightarrow \quad \omega = \frac{\dot{\theta} \cdot r}{R} \tag{23}$$

Where the radius, R, can be calculated by trignometric analysis;

$$\sin\alpha = \frac{b/2}{R} \quad \Rightarrow \quad R = \frac{b}{2\sin\alpha} \tag{24}$$

$$\omega = \frac{2\,\dot{\theta}\,r\,\sin\alpha}{b} \tag{25}$$

The translational velocity of the robot in dual wheel steering is directly related to the angular displacement of the wheels by differentiating the expression from Equation 22;

$$v = \dot{\theta} \cdot r \tag{26}$$

The full forward kinematics of dual wheel bicycle steering can be expressed as:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = r \cdot \begin{bmatrix} 1 \\ \frac{2}{b} \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta} \\ \dot{\theta}\sin\alpha \end{bmatrix} \tag{27}$$

And the inverse kinematics as:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\theta}\sin\alpha \end{bmatrix} = \frac{1}{r} \cdot \begin{bmatrix} 1 \\ \frac{b}{2} \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{28}$$

Odometry
For dual wheel kinematics of a bicycle, the following was derived:

$$S = \theta \cdot r \quad , \quad \psi = \frac{2\,\theta\, r\, \sin\alpha}{b} \tag{29}$$

Which is rewritten according to Figure 45;

$$\Delta S = \Delta\theta \cdot r \quad , \quad \Delta\psi = \frac{2\,\Delta\theta\, r\, \sin\alpha}{b} \tag{30}$$

Where:

ΔS = Difference in robot displacement
$\Delta\theta$ = Difference in wheel displacement
$\Delta\psi$ = Difference in robot rotation

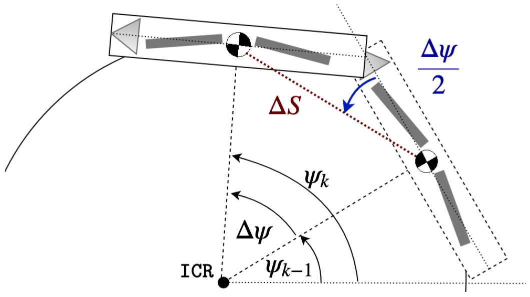


Figure 45: Odometry

As illustrated in Figure 45, the difference in robot displacement is approximated to be equal to the peripheral displacement of the robot, and the angles are defined from the robot's x-axis as illustrated in Figure 46. The x-axis is perpendicular to the forward direction of the bicycle configuration since there are multiple odometric models that are to be combined and the coordinate system of the bicycle configuration must correspond to the coordinate system of the differential configuration. The geometric differences of the robot's local and global coordinates, Δx and Δy, are illustrated in vector form in Figure 47. The black arrows in Figure 47 indicates the forward direction of the robot.

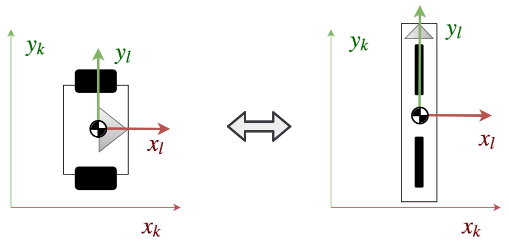


Figure 46: Coordinate systems

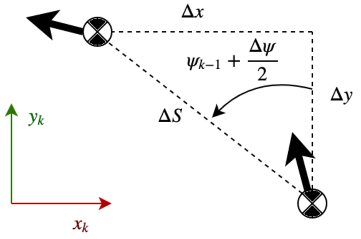


Figure 47: Geometry

Where l and k indicate the local and global coordinates, respectively. Therefore, the global displacement differences can be set up by trigonometric analysis according to Figure 47:

$$\Delta x = -\Delta S\, \sin(\psi_{k-1} + \frac{\Delta\psi}{2}) \tag{31}$$

$$\Delta y = \Delta S\, \cos(\psi_{k-1} + \frac{\Delta\psi}{2}) \tag{32}$$

Which by dead reckoning yields:

$$\begin{bmatrix} x_k \\ y_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \psi_{k-1} \end{bmatrix} + \begin{bmatrix} -\Delta S\, \sin(\psi_{k-1} + \frac{\Delta\psi}{2}) \\ \Delta S\, \cos(\psi_{k-1} + \frac{\Delta\psi}{2}) \\ \Delta\psi \end{bmatrix} \tag{33}$$

3.4.2 Front Wheel Bicycle Steering

Kinematics

This kinematic model is utilized for the front wheel balancing mode on the robot. A common and simple way to derive the kinematics of such a front wheel bicycle steering model is to assume that the origin of the robot is centered at the rear wheel, but since the origin of this system is directly between the wheels as discussed in 3.4.1, this approach can not be used. Figure 48 shows the kinematic motion of a system with the origin between the wheels.

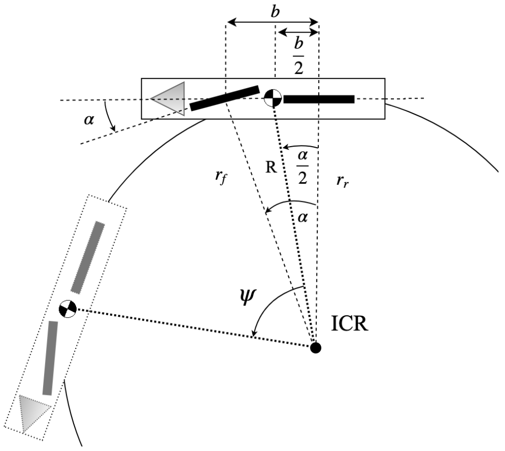


Figure 48: Single wheel steering kinematics

For the robot's origin, the distance traveled corresponds to the average distance traveled between the two wheels, and the velocity of the robot corresponds to the derivative of this distance;

$$S = \frac{S_f + S_r}{2} \tag{34}$$

$$v = \dot{S} = \frac{\dot{S}_f + \dot{S}_r}{2} \tag{35}$$

Where f and r indicate the front and rear wheels respectively. This can be rewritten by substituting with:

$$\dot{S}_f = \dot{\theta}_f \cdot r \tag{36}$$

$$\dot{S}_r = \dot{\theta}_r \cdot r \tag{37}$$

Which yields:

$$v = r\,\frac{\dot{\theta}_f + \dot{\theta}_r}{2} \tag{38}$$

The rotational velocity of the robot can be expressed with the velocity component of the front wheel perpendicular to the robot relative to the rear wheel;

$$\omega = \frac{v_f}{b}\sin\alpha = \frac{r\,\dot{\theta}_f}{b}\sin\alpha \tag{39}$$

Subsequently, the forward kinematics can be set up as:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = r \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{b}\sin\alpha & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_f \\ \dot{\theta}_r \end{bmatrix} \tag{40}$$

And the inverse kinematics can be set up as:

$$\begin{bmatrix} \dot{\theta}_f \\ \dot{\theta}_r \end{bmatrix} = \frac{1}{r} \cdot \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{b}\sin\alpha & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{1}{r} \cdot \begin{bmatrix} 0 & \frac{b}{\sin\alpha} \\ 2 & -\frac{b}{\sin\alpha} \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{41}$$

Odometry

The odometry of the front wheel steering is derived similarly to the odometry of the dual wheel bicycle steering from Chapter 3.4.1.

From the kinematics, it was derived that:

$$v = r\,\frac{\dot{\theta}_f + \dot{\theta}_r}{2} \quad , \quad \omega = r\,\frac{\dot{\theta}_f}{b}\,\sin\alpha \tag{42}$$

Which by integration yields:

$$S = r\,\frac{\theta_f + \theta_r}{2} \quad , \quad \psi = \frac{r\,\theta_f}{b}\,\sin\alpha \tag{43}$$

Again, the difference in robot displacement is approximated to be equal to the peripheral displacement of the robot, as shown in Figure 49:

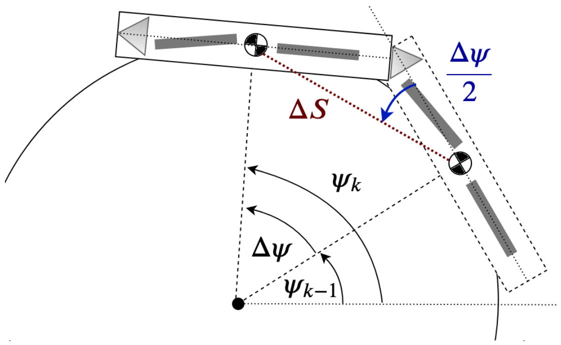


Figure 49: Single wheel steering odometry

$$\Delta S = r\,\frac{\Delta\theta_f + \Delta\theta_r}{2} \quad , \quad \Delta\psi = \frac{r\,\Delta\theta_f}{b}\,\sin\alpha \tag{44}$$

ΔS = Difference in robot displacement
$\Delta\theta$ = Difference in wheel displacement
$\Delta\psi$ = Difference in robot rotation

Since the front wheel steering and dual wheel steering share the same coordinate system, the odometry is solved the same way as illustrated in Figure 50.

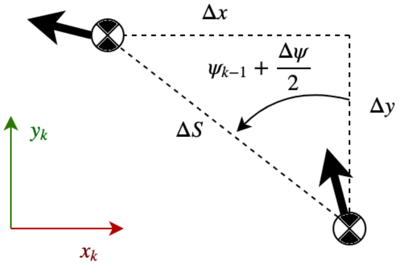


Figure 50: Geometric relationships

$$\Delta x = -\Delta S\,\sin(\psi_{k-1} + \frac{\Delta\psi}{2}) \tag{45}$$

$$\Delta y = \Delta S\,\cos(\psi_{k-1} + \frac{\Delta\psi}{2}) \tag{46}$$

And the resulting odometry is by dead reckoning set up as:

$$\begin{bmatrix} x_k \\ y_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \psi_{k-1} \end{bmatrix} + \begin{bmatrix} -\Delta S\,\sin(\psi_{k-1} + \frac{\Delta\psi}{2}) \\ \Delta S\,\cos(\psi_{k-1} + \frac{\Delta\psi}{2}) \\ \Delta\psi \end{bmatrix} \tag{47}$$

3.4.3 Differential Drive

Kinematics

Figure 51 illustrates the typical movement of a differential drive robot. The travel distance for each wheel can be described as:

$$S_R = \psi \cdot (R + \frac{b}{2}) \tag{48}$$

$$S_L = \psi \cdot (R - \frac{b}{2}) \tag{49}$$

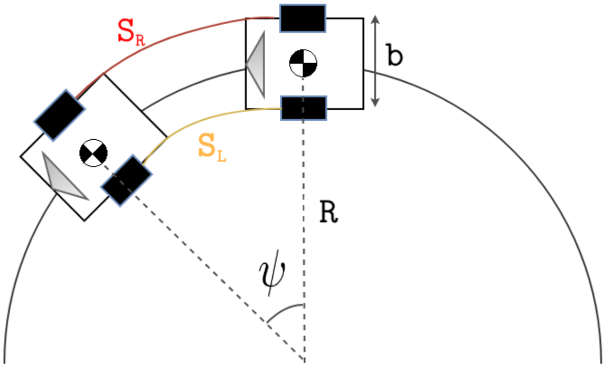


Figure 51: Differential drive

Subtracting the two expressions yields:

$$S_R - S_L = \psi \cdot b \Rightarrow \psi = \frac{S_R - S_L}{b} \tag{50}$$

The following applies:

$$\dot{S}_R = \dot{\theta}_R \cdot r \tag{51}$$

$$\dot{S}_L = \dot{\theta}_L \cdot r \tag{53}$$

$$\omega = \dot{\psi} \tag{52}$$

$$v = \dot{S} \tag{54}$$

And by differentiation and substitution into Equation 50, the following expression for the robot's turn rate ω is derived:

$$\omega = r \cdot \frac{\dot{\theta}_R - \dot{\theta}_L}{b} \tag{55}$$

The average distance travelled of the two wheels describes the total translational distance covered by the robot;

$$S = \frac{S_R + S_L}{2} \tag{56}$$

Which by differentiation and substitution yields an expression for the robot's velocity, v:

$$v = r \cdot \frac{\dot{\theta}_R + \dot{\theta}_L}{2} \tag{57}$$

Knowing expressions for both translational and rotational velocity of the robot, the forward kinematics can be set up;

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = r \cdot \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_R \\ \dot{\theta}_L \end{bmatrix} \tag{58}$$

And subsequently, the inverse kinematics are set up as:

$$\begin{bmatrix} \dot{\theta}_R \\ \dot{\theta}_L \end{bmatrix} = \frac{1}{r} \cdot \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix}^{-1} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{1}{r} \cdot \begin{bmatrix} 1 & \frac{b}{2} \\ 1 & -\frac{b}{2} \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{59}$$

Odometry

The odometry of the differential configuration can be derived based on the measurements illustrated in Figure 52 and the kinematics previously described;

$$S = \frac{S_R + S_L}{2} \, , \, \psi = \frac{S_R - S_L}{b} \tag{60}$$

Which yields:

$$\Delta S = \frac{\Delta S_R + \Delta S_L}{2} \quad , \quad \Delta\psi = \frac{\Delta S_R - \Delta S_L}{b} \tag{61}$$

Where:

ΔS = Difference in robot displacement
ΔS_R = Difference in right wheel displacement
ΔS_L = Difference in left wheel displacement
$\Delta\psi$ = Difference in robot rotation

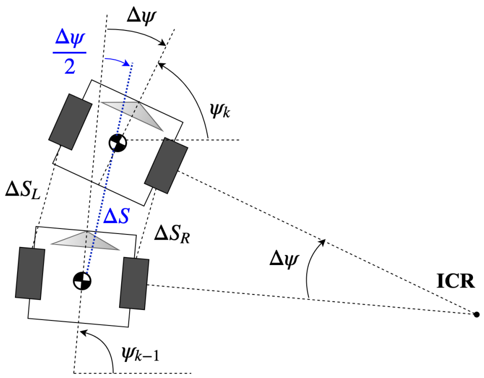


Figure 52: Robot Odometry

Similarly to the odometry for the bicycle configuration described in Chapters 3.4.1 and 3.4.2, the difference in robot displacement is approximated to be equal to the peripheral displacement of the robot. Figure 53 shows the geometry necessary to calculate the coordinates by dead reckoning.

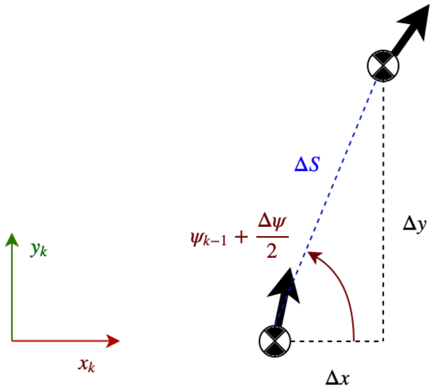


Figure 53: Geometry

Hence, the global differences in robot displacement can be set up as follows:

$$\Delta x = \Delta S \, \cos(\psi_{k-1} + \frac{\Delta\psi}{2}) \tag{62}$$

$$\Delta y = \Delta S \, \sin(\psi_{k-1} + \frac{\Delta\psi}{2}) \tag{63}$$

Subsequently, the full odometry is calculated through dead reckoning as shown in Equation 64.

$$\begin{bmatrix} x_k \\ y_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \psi_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta S \, \cos(\psi_{k-1} + \frac{\Delta\psi}{2}) \\ \Delta S \, \sin(\psi_{k-1} + \frac{\Delta\psi}{2}) \\ \Delta\psi \end{bmatrix} \tag{64}$$

3.5 Dynamic Systems

This chapter will revolve around the derivation of the dynamic systems of the three different balancing modes on the robot. Note that the definition of the coordinate systems may vary between the control modes. This is done for derivation purposes only, and the main coordinate system of the robot remains as described in Chapter 2.4.

3.5.1 Inertia Wheel, Bicycle Dynamics

The dynamic behavior of the service robot in bicycle mode will act like an inverted pendulum with a fixed pivot point and will be further described and presented with a mathematical model, transfer function and state space model in this chapter. These dynamics apply when the kinematics in Chapter 3.4.1 are used.

Mathematical Model

Figures 54, 55 and 56 show a sketch, the free body diagram and the kinetic diagram of the inertia wheel bicycle mode.

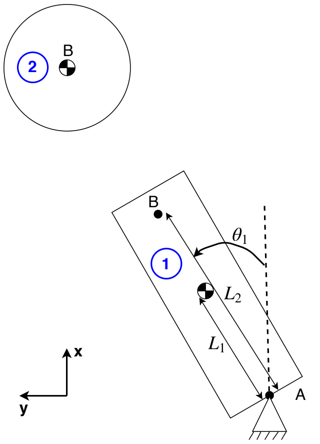


Figure 54: Sketch: rear view

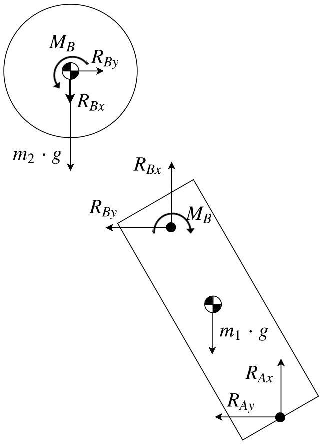


Figure 55: Free body diagrams

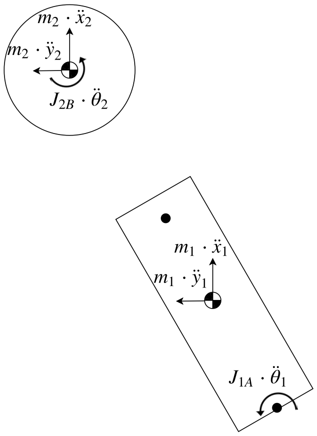


Figure 56: Kinetic diagrams

The following dynamic equations 65, 67, 68 and 69 are derived from Figure 55 and 56, where the two main bodies are separated. These two bodies; body 1 and 2, is shown with blue circles. Body 2 is the inertia wheel and its connected rotating parts, and body 1 is the equivalent body of the bicycle, which includes everything except body 2.

m_1	= Mass of body 1
m_2	= Mass of body 2
g	= Gravitational acceleration
R_{Ax} , R_{Ay}	= Reaction forces in point A
R_{Bx} , R_{By}	= Reaction forces in point B
M_B	= Torque from the inertia wheel
L_1	= Distance between A and COM
L_2	= Distance between A and B
J_{1A}	= Moment of inertia of body 1 around point A
J_{2B}	= Moment of inertia of body 2 around point B
θ_1	= Pitch angle of body 1
$\ddot{\theta}_1$	= Angular acceleration of body 1
$\ddot{\theta}_2$	= Angular acceleration of body 2
$\ddot{x}_1$, $\ddot{y}_1$	= Linear acceleration of body 1 at COM
$\ddot{x}_2$, $\ddot{y}_2$	= Linear acceleration of body 2 at COM

Body 1:

$$\Sigma \overset{\overset{\curvearrowleft}{+}}{M}_{1A} = J_{1A} \cdot \ddot{\theta}_1 = m_1 \cdot g \cdot \sin\theta_1 \cdot L_1 + L_2 \cdot \begin{bmatrix} \cos\theta_1 \\ \sin\theta_1 \end{bmatrix} \times \begin{bmatrix} R_{Bx} \\ R_{By} \end{bmatrix} - M_B \tag{65}$$

$$J_{1A} \cdot \ddot{\theta}_1 = m_1 \cdot g \cdot \sin\theta_1 \cdot L_1 + L_2 \cdot \cos\theta_1 \cdot R_{By} - L_2 \cdot \sin\theta_1 \cdot R_{Bx} - M_B \tag{66}$$

$$\Sigma F_1 = m_1 \cdot \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \end{bmatrix} = \begin{bmatrix} R_{Ax} \\ R_{Ay} \end{bmatrix} + \begin{bmatrix} -m_1 \cdot g \\ 0 \end{bmatrix} + \begin{bmatrix} R_{Bx} \\ R_{By} \end{bmatrix} \tag{67}$$

Body 2:

$$\Sigma \overset{\overset{\curvearrowleft}{+}}{M}_{2B} = J_{2B} \cdot \ddot{\theta}_2 = M_B \tag{68}$$

$$\Sigma F_2 = m_2 \cdot \begin{bmatrix} \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} = \begin{bmatrix} -R_{Bx} \\ -R_{By} \end{bmatrix} + \begin{bmatrix} -m_2 \cdot g \\ 0 \end{bmatrix} \tag{69}$$

$$\begin{bmatrix} R_{Bx} \\ R_{By} \end{bmatrix} = -m_2 \cdot \begin{bmatrix} \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} + \begin{bmatrix} -m_2 \cdot g \\ 0 \end{bmatrix} \tag{70}$$

$$\begin{bmatrix} R_{Bx} \\ R_{By} \end{bmatrix} = \begin{bmatrix} -m_2 \cdot \ddot{x}_2 - m_2 \cdot g \\ -m_2 \cdot \ddot{y}_2 \end{bmatrix} \tag{71}$$

The following kinematic equations 72 and 75 is also determined through analysis of the free body diagram and kinetic diagram in Figure 55 and 56.

Body 1:

$$\mathbf{r_{G1}} = L_1 \cdot \begin{bmatrix} \cos\theta_1 \\ \sin\theta_1 \end{bmatrix} \tag{72}$$

$$\dot{\mathbf{r}}_{\mathbf{G1}} = L_1 \cdot \dot{\theta}_1 \cdot \begin{bmatrix} -\sin\theta_1 \\ \cos\theta_1 \end{bmatrix} \tag{73}$$

$$\ddot{\mathbf{r}}_{\mathbf{G1}} = L_1 \cdot \ddot{\theta}_1 \cdot \begin{bmatrix} -\sin\theta_1 \\ \cos\theta_1 \end{bmatrix} + L_1 \cdot \dot{\theta}_1^2 \cdot \begin{bmatrix} -\cos\theta_1 \\ -\sin\theta_1 \end{bmatrix} \tag{74}$$

Body2:

$$\mathbf{r_{G2}} = L_2 \cdot \begin{bmatrix} \cos\theta_1 \\ \sin\theta_1 \end{bmatrix} \tag{75}$$

$$\dot{\mathbf{r}}_{\mathbf{G2}} = L_2 \cdot \dot{\theta}_1 \cdot \begin{bmatrix} -\sin\theta_1 \\ \cos\theta_1 \end{bmatrix} \tag{76}$$

$$\ddot{\mathbf{r}}_{\mathbf{G2}} = L_2 \cdot \ddot{\theta}_1 \cdot \begin{bmatrix} -\sin\theta_1 \\ \cos\theta_1 \end{bmatrix} + L_2 \cdot \dot{\theta}_1^2 \cdot \begin{bmatrix} -\cos\theta_1 \\ -\sin\theta_1 \end{bmatrix} \tag{77}$$

$$\ddot{\mathbf{r}}_{\mathbf{G2}} = \begin{bmatrix} \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} = \begin{bmatrix} -L_2 \cdot \ddot{\theta}_1 \cdot \sin\theta_1 - L_2 \cdot \dot{\theta}_1^2 \cdot \cos\theta_1 \\ L_2 \cdot \ddot{\theta}_1 \cdot \cos\theta_1 - L_2 \cdot \dot{\theta}_1^2 \cdot \sin\theta_1 \end{bmatrix} \tag{78}$$

Solving for the desired output θ_1, starting with Equation 66, represented as Equation 79 below:

$$
\begin{aligned}
J_{1,A} \cdot \ddot{\theta}_1 &= m_1 \cdot g \cdot \sin\theta_1 \cdot L_1 + L_2 \cdot \cos\theta_1 \cdot R_{By} - L_2 \cdot \sin\theta_1 \cdot R_{Bx} - M_B && (79)\\
&= m_1 \cdot g \cdot \sin\theta_1 \cdot L_1 + L_2 \cdot \cos\theta_1 \cdot (-m_2 \cdot \ddot{y}_2) \\
&\quad - L_2 \cdot \sin\theta_1 \cdot (-m_2 \cdot \ddot{x}_2 - m_2 \cdot g) - M_B && (80)\\
&= m_1 \cdot g \cdot \sin\theta_1 \cdot L_1 + L_2 \cdot \cos\theta_1 \cdot (-m_2 \cdot [L_2 \cdot \ddot{\theta}_1 \cdot \cos\theta_1 - L_2 \cdot \dot{\theta}_1^2 \cdot \sin\theta_1]) \\
&\quad - L_2 \cdot \sin\theta_1 \cdot (-m_2 \cdot [-L_2 \cdot \ddot{\theta}_1 \cdot \sin\theta_1 - L_2 \cdot \dot{\theta}_1^2 \cdot \cos\theta_1] - m_2 \cdot g) - M_B && (81)\\
&= m_1 \cdot g \cdot \sin\theta_1 \cdot L_1 + (-L_2 \cdot \cos\theta_1 \cdot m_2 \cdot [L_2 \cdot \ddot{\theta}_1 \cdot \cos\theta_1 - L_2 \cdot \dot{\theta}_1^2 \cdot \sin\theta_1]) \\
&\quad + (L_2 \cdot \sin\theta_1 \cdot m_2 \cdot [-L_2 \cdot \ddot{\theta}_1 \cdot \sin\theta_1 - L_2 \cdot \dot{\theta}_1^2 \cdot \cos\theta_1] + L_2 \cdot \sin\theta_1 \cdot m_2 \cdot g) - M_B && (82)\\
&= m_1 \cdot g \cdot \sin\theta_1 \cdot L_1 - L_2^2 \cdot \cos^2\theta_1 \cdot m_2 \cdot \ddot{\theta}_1 + \cancel{L_2^2 \cdot m_2 \cdot \dot{\theta}_1^2 \cdot \cos\theta_1 \cdot \sin\theta_1} \\
&\quad - L_2^2 \cdot \sin^2\theta_1 \cdot m_2 \cdot \ddot{\theta}_1 - \cancel{L_2^2 \cdot m_2 \cdot \dot{\theta}_1^2 \cdot \cos\theta_1} \cdot \sin\theta_1 + L_2 \cdot \sin\theta_1 \cdot m_2 \cdot g - M_B && (83)\\
&= m_1 \cdot g \cdot \sin\theta_1 \cdot L_1 - L_2^2 \cdot m_2 \cdot \ddot{\theta}_1 \cdot (\cos^2\theta_1 + \sin^2\theta_1) + L_2 \cdot \sin\theta_1 \cdot m_2 \cdot g - M_B && (84)\\
&= m_1 \cdot g \cdot \sin\theta_1 \cdot L_1 - L_2^2 \cdot m_2 \cdot \ddot{\theta}_1 + L_2 \cdot \sin\theta_1 \cdot m_2 \cdot g - M_B && (85)
\end{aligned}
$$

$$\ddot{\theta}_1 = \frac{m_1 \cdot g \cdot \sin\theta_1 \cdot L_1 + L_2 \cdot \sin\theta_1 \cdot m_2 \cdot g - M_B}{J_{1A} + L_2^2 \cdot m_2} \qquad (86)$$

$$\ddot{\theta}_1 = \frac{m_1 \cdot g \cdot \sin\theta_1 \cdot L_1 + L_2 \cdot \sin\theta_1 \cdot m_2 \cdot g - J_{2B} \cdot \ddot{\theta}_2}{J_{1A} + L_2^2 \cdot m_2} \qquad (87)$$

To linearize the differential equation 87, small pitch angles are assumed. Therefore $\sin\theta_1 \approx \theta_1$, $\cos\theta_1 \approx 1$ and also $\dot{\theta}_1^2 \approx 0$ is applied on Equation 87 as seen in Equation 88. And so the following differential equations describe the dynamics of the system.

$$\ddot{\theta}_1 = \frac{m_1 \cdot g \cdot \theta_1 \cdot L_1 + L_2 \cdot \theta_1 \cdot m_2 \cdot g - J_{2B} \cdot \ddot{\theta}_2}{J_{1A} + L_2^2 \cdot m_2} \qquad (88)$$

$$\ddot{\theta}_1 = (\frac{m_1 \cdot g \cdot L_1 + m_2 \cdot g \cdot L_2}{J_{1A} + L_2^2 \cdot m_2}) \cdot \theta_1 + (\frac{-J_{2B}}{J_{1A} + L_2^2 \cdot m_2}) \cdot \ddot{\theta}_2 \qquad (89)$$

Transfer Function
In order to control the pitch angle of the service robot, a transfer function with the balancing wheel angular acceleration $\ddot{\theta}_2$ as input and the pitch angle θ_1 as output is desired. From sub-chapter 3.5.1 the dynamic differential equation 88 is used to find the desired transfer function.

$$\ddot{\theta}_1 = \frac{m_1 \cdot g \cdot \theta_1 \cdot L_1 + L_2 \cdot \theta_1 \cdot m_2 \cdot g - J_{2,B} \cdot \ddot{\theta}_2}{J_{1,A} + L_2^2 \cdot m_2}$$

$$(J_{1,A} + L_2^2 \cdot m_2) \cdot \ddot{\theta}_1 = (m_1 \cdot g \cdot L_1 + m_2 \cdot g \cdot L_2) \cdot \theta_1 - J_{2,B} \cdot \ddot{\theta}_2 \tag{90}$$

$$(J_{1,A} + L_2^2 \cdot m_2) \cdot \ddot{\theta}_1 - (m_1 \cdot g \cdot L_1 + m_2 \cdot g \cdot L_2) \cdot \theta_1 = -J_{2,B} \cdot \ddot{\theta}_2 \tag{91}$$

Applying a Laplace transformation on Equation 91:

$$(J_{1,A} + L_2^2 \cdot m_2) \cdot s^2 \cdot \theta_1 - (m_1 \cdot g \cdot L_1 + m_2 \cdot g \cdot L_2) \cdot \theta_1 = -J_{2,B} \cdot s^2 \cdot \theta_2 \tag{92}$$

$$((J_{1,A} + L_2^2 \cdot m_2) \cdot s^2 - (m_1 \cdot g \cdot L_1 + m_2 \cdot g \cdot L_2))\theta_1 = -J_{2,B} \cdot s^2 \cdot \theta_2 \tag{93}$$

$$\frac{\theta_1}{\theta_2} = \frac{-J_2 \cdot s^2}{(J_{1,A} + L_2^2 \cdot m_2) \cdot s^2 - (m_1 \cdot g \cdot L_1 + m_2 \cdot g \cdot L_2)} \tag{94}$$

The input θ_2 is changed from angular position to angular acceleration by dividing both sides with s^2.

$$G = \frac{\theta_1}{s^2 \cdot \theta_2} = \frac{-J_2 \cdot s^2}{((J_{1,A} + L_2^2 \cdot m_2) \cdot s^2 - (m_1 \cdot g \cdot L_1 + m_2 \cdot g \cdot L_2)) \cdot s^2} \tag{95}$$

$$G = \frac{\theta_1}{\ddot{\theta}_2} = \frac{-J_2}{(J_{1,A} + L_2^2 \cdot m_2) \cdot s^2 - (m_1 \cdot g \cdot L_1 + m_2 \cdot g \cdot L_2)} \tag{96}$$

$$G = \frac{\theta_1}{\ddot{\theta}_2} = \frac{(-J_2)}{(J_{1,A} + L_2^2 \cdot m_2) \cdot s^2 + (-m_1 \cdot g \cdot L_1 - m_2 \cdot g \cdot L_2)} \tag{97}$$

State Space Model
The differential equation 89 found in Chapter 3.5.1 is transformed into a state space canonical form as shown below.

$$\ddot{\theta}_1 = (\frac{m_1 \cdot g \cdot L_1 + m_2 \cdot g \cdot L_2}{J_{1A} + L_2^2 \cdot m_2}) \cdot \theta_1 + (\frac{-J_{2B}}{J_{1A} + L_2^2 \cdot m_2}) \cdot \ddot{\theta}_2$$

The dynamic states x, the input u and the output y of the state-space model are determined below in Equation 98, 99 and 100 respectively.

$$x = [x_1 \; x_2]^T = [\theta_1 \; \dot{\theta}_1]^T \tag{98}$$

$$u = \ddot{\theta}_2 \tag{99}$$

$$y = \theta_1 \tag{100}$$

The dynamic states are then differentiated as shown below:

$$\dot{x}_1 = \dot{\theta}_1 \Rightarrow \dot{x}_1 = x_2 \tag{101}$$

$$\dot{x}_2 = \ddot{\theta}_1 \Rightarrow \dot{x}_2 = (\frac{m_1 \cdot g \cdot L_1 + m_2 \cdot g \cdot L_2}{J_{1A} + L_2^2 \cdot m_2}) \cdot x_1 + (\frac{-J_{2B}}{J_{1A} + L_2^2 \cdot m_2}) \cdot u \tag{102}$$

The canonical form of the state-space model is shown in Equations 103 and 104.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ (\frac{m_1 \cdot g \cdot L_1 + m_2 \cdot g \cdot L_2}{J_{1A} + L_2^2 \cdot m_2}) & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ (\frac{-J_{2B}}{J_{1A} + L_2^2 \cdot m_2}) \end{bmatrix} \cdot u \tag{103}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0] \cdot u \tag{104}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ (\frac{m_1 \cdot g \cdot L_1 + m_2 \cdot g \cdot L_2}{J_{1A} + L_2^2 \cdot m_2}) & 0 \end{bmatrix} \tag{105}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ (\frac{-J_{2B}}{J_{1A} + L_2^2 \cdot m_2}) \end{bmatrix} \tag{106}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{107}$$

$$\mathbf{D} = [0] \tag{108}$$

3.5.2 Front Wheel Steering, Bicycle Dynamics

This Chapter 3.5.2 will cover the dynamics of the bicycle when it is balanced by only altering the steering angle of the front wheel [1].

Mathematical Model

Two sketches covering some general geometric parameters of the bicycle are shown in Figure 57 and 58, where the center of mass is assumed to be exactly in the middle of the hub wheels.

From Figure 58 the radius around the ICR is calculated and linearized assuming the steering angle is small, as done in Equation 109 where $\sin\frac{\alpha}{2} \approx \frac{\alpha}{2}$.

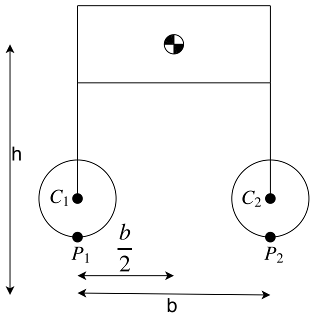


Figure 57: Sketch, side view of the bicycle

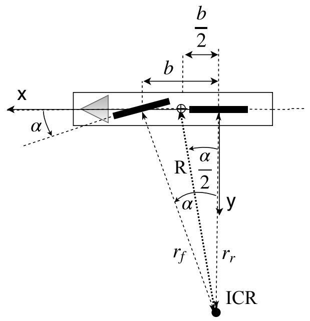


Figure 58: Sketch, Top view of the bicycle

$$\frac{b}{2} = R \sin\frac{\alpha}{2} \Rightarrow R = \frac{b}{2 \sin\frac{\alpha}{2}} = \frac{b}{\alpha} \tag{109}$$

The bicycle's respective sketch and free body diagram based on a rear view of the bicycle are shown in Figure 59 and 60.

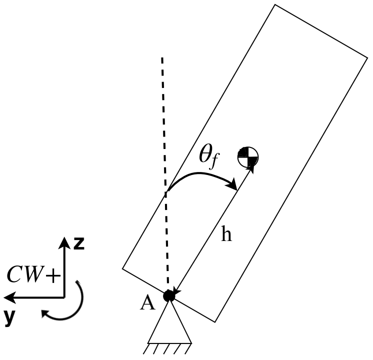


Figure 59: Sketch, rear view of the bicycle

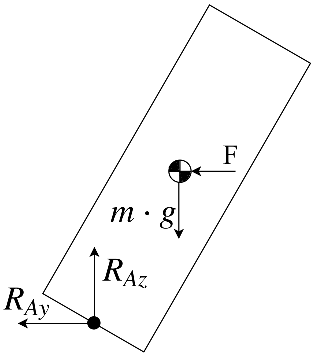


Figure 60: Free body diagram

The force F is the combination of the centrifugal force (F_c) and the force occurring from changing the steering angle (F_s). This can then be expressed with equation 110. Point A is the contact point between the wheels and the ground, and the parameter m is the total mass of the whole bicycle. The bicycles translational speed is assumed constant and referred to as v.

$$F = F_c + F_s \tag{110}$$

The contribution from the centrifugal force can be written as in Equation 111

$$F_c = \frac{m \cdot v^2}{R} = \frac{m \cdot v^2 \cdot \alpha}{b} \tag{111}$$

Figure 61 shows the distances and accelerations of the points P2, which is the front wheel, and P_m, which is the center of mass, based on the bicycles rotation around the rear wheel, P_1.

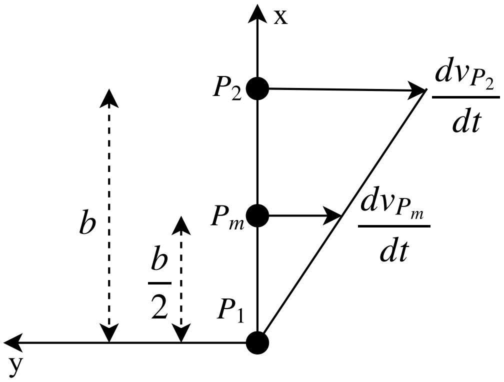


Figure 61: Distances and accelerations relative to point P_1 [1]

So, the force occuring from the steering angle can be found based on Equation 112 and the relationship described in Equation 113 which is based on Figure 61.

$$\frac{dv_{P2}}{dt} = v \cdot \frac{d\alpha}{dt} \tag{112}$$

$$\frac{dv_{Pm}}{dt} = \frac{\frac{b}{2}}{b} \cdot \frac{dv_{P2}}{dt} = \frac{1}{2} \cdot v \cdot \frac{d\alpha}{dt} \tag{113}$$

This yields the relationship for F_s as in Equation 114

$$F_s = m \cdot \frac{1}{2} \cdot v \cdot \frac{d\alpha}{dt} = \frac{m \cdot v}{2} \cdot \dot{\alpha} \tag{114}$$

Newtons 2nd law applied on Figure 60 around point A, gives Equation 115

$$\Sigma M = J_A \cdot \ddot{\theta}_f = m \cdot g \cdot h \cdot \sin\theta_f - F \cdot h \cdot \cos\theta_f \tag{115}$$

The pitch angle θ_f is assumed to be small, close to zero, and is therefore linearized by applying the following approximations in Equation 116: $\sin\theta_f \approx \theta_f$ and $\cos\theta_f \approx 1$.

$$J_A \cdot \ddot{\theta}_f = m \cdot g \cdot h \cdot \theta_f - F \cdot h \tag{116}$$

Substituting Equation 111 and 114 into Equation 116 yields the dynamics of the system as shown in differential Equation 117.

$$J_A \cdot \ddot{\theta}_f - m \cdot g \cdot h \cdot \theta_f = -(\frac{m \cdot v^2 \cdot \alpha}{b} + \frac{m \cdot v}{2} \cdot \dot{\alpha}) \cdot h \tag{117}$$

Transfer Function

By applying the Laplace transform on Equation 117, Equation 118 is found, which then leads to the desired transfer function shown in Equation 119.

$$(J_A \cdot s^2 - m \cdot g \cdot h) \cdot \theta_f = -(\frac{m \cdot v^2 \cdot h}{b} + \frac{m \cdot v \cdot h}{2} \cdot s) \cdot \alpha \tag{118}$$

$$G = \frac{\theta_f(s)}{\alpha(s)} = (\frac{(-\frac{m \cdot v \cdot h}{2 \cdot J_A}) \cdot s + (-\frac{m \cdot v^2 \cdot h}{J_A \cdot b})}{s^2 + (-\frac{m \cdot g \cdot h}{J_A})}) = \frac{b_0 \cdot s^2 + b_1 \cdot s + b_2}{s^2 + a_1 \cdot s + a_2} \tag{119}$$

State Space Model

The transfer function in Equation 119 is transformed into a canoncial state space form as shown below [15].

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -a_2 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_1 - a_1 \cdot b_0 \\ b_2 - a_2 \cdot b_0 \end{bmatrix} \tag{120}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad \mathbf{D} = [b_0] \tag{121}$$

The input is the steering angle of the front wheel and the output is the pitch angle of the bicycle, as shown in Equation 123 and 124. The first state of the system x_1 is the pitch angle of the bicycle as seen from the **C** matrix which is [1 0]. The second state is defined as in Equation 122 according to source [1].

$$x = [x_1 \ x_2]^T = [\theta_f \ (\dot{\theta}_f - k \cdot \alpha)]^T \quad \text{where:} \quad k = -\frac{m \cdot v \cdot h}{2 \cdot J_A \cdot b} \tag{122}$$

$$u = \alpha \tag{123}$$

$$y = \theta_f \tag{124}$$

The dynamic system is then represented in state space form as shown in Equation 125 and 126.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ (\frac{m \cdot g \cdot h}{J_A}) & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -\frac{m \cdot v \cdot h}{2 \cdot J_A} \\ \frac{m \cdot v^2 \cdot h}{J_A \cdot b} \end{bmatrix} \cdot u \tag{125}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0] \cdot u \tag{126}$$

3.5.3 Differential Drive Dynamics

Mathematical Model

The differential drive mode can be controlled based on the theory behind an inverted pendulum with a variable pivot point. A sketch, FBD and KD of such an inverted pendulum are illustrated in Figures 62, 63 and 64 respectively.

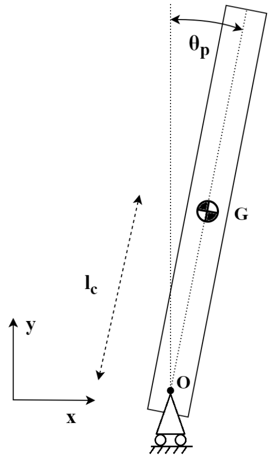


Figure 62: Sketch

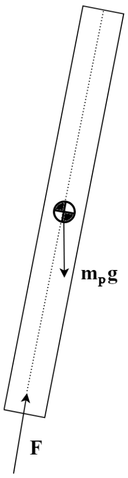


Figure 63: Free body diagram

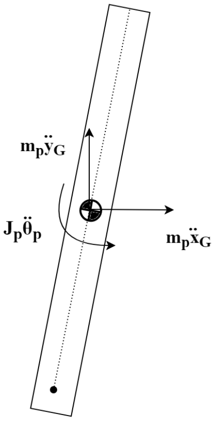


Figure 64: Kinetic diagram

Where:

m_p = Mass of the pendulum
J_p = Mass moment of inertia of the pendulum
g = Gravitational acceleration
F = Input force of the system
θ_p = Pitch of the inverted pendulum
l_c = Distance between reaction point and COM of the pendulum

Note that all parameters are redefined from the bicycle's mathematical models in order to differentiate the dynamic models. θ_p for instance, corresponds to θ_1 from Chapter 3.5.1 and θ_f from Chapter 3.5.2. Force analysis of the pendulum's body yields:

$$\sum F_p = \begin{bmatrix} m_p\ddot{x_G} \\ m_p\ddot{y_G} \end{bmatrix} = \begin{bmatrix} -F\sin\theta_p \\ -m_p g + F\,\cos\theta_p \end{bmatrix} \tag{127}$$

Where air resistance is neglected. The equation of motion of the center of mass can be expressed as:

$$\mathbf{r}_G = \mathbf{r}_O + \mathbf{r}_{GO} = \mathbf{r}_O + l_c \begin{bmatrix} -\sin\theta_p \\ \cos\theta_p \end{bmatrix} \tag{128}$$

$$\dot{\mathbf{r}}_G = \dot{\mathbf{r}}_0 + \dot{\mathbf{r}}_{GO} = \dot{\mathbf{r}}_O + \dot{\theta}_p\, l_c \begin{bmatrix} -\cos\theta_p \\ -\sin\theta_p \end{bmatrix} \tag{129}$$

$$\ddot{\mathbf{r}}_G = \ddot{\mathbf{r}}_O + \ddot{\mathbf{r}}_{GO} = \ddot{\mathbf{r}}_O + \ddot{\theta}_p\, l_c \begin{bmatrix} -\cos\theta_p \\ -\sin\theta_p \end{bmatrix} + \dot{\theta}_p^2\, l_c \begin{bmatrix} \sin\theta_p \\ -\cos\theta_p \end{bmatrix} \tag{130}$$

Where:

$\mathbf{r}_G\,,\dot{\mathbf{r}}_G\,,\ddot{\mathbf{r}}_G$ = Displacement, velocity and acceleration of the pendulum's center of mass, G
$\mathbf{r}_O\,,\dot{\mathbf{r}}_O\,,\ddot{\mathbf{r}}_O$ = Displacement, velocity and acceleration of the reaction point, O
$\mathbf{r}_{GO}\,,\dot{\mathbf{r}}_{GO}\,,\ddot{\mathbf{r}}_{GO}$ = Displacement, velocity and acceleration of G relative to O

Which yields:

$$\begin{bmatrix}\ddot{x}_G\\ \ddot{y}_G\end{bmatrix} = \begin{bmatrix}\ddot{x}_O\\ 0\end{bmatrix} + \ddot{\theta}_p\,l_c\begin{bmatrix}-\cos\theta_p\\ -\sin\theta_p\end{bmatrix} + \dot{\theta}_p^2\,l_c\begin{bmatrix}\sin\theta_p\\ -\cos\theta_p\end{bmatrix} \tag{131}$$

And by substitution into Equation 127 gives:

$$m_p\,(\begin{bmatrix}\ddot{x}_O\\ 0\end{bmatrix} + \ddot{\theta}_p\,l_c\begin{bmatrix}-\cos\theta_p\\ -\sin\theta_p\end{bmatrix} + \dot{\theta}_p^2\,l_c\begin{bmatrix}\sin\theta_p\\ -\cos\theta_p\end{bmatrix}) = \begin{bmatrix}-F\sin\theta_p\\ -m_pg + F\,\cos\theta_p\end{bmatrix} \tag{132}$$

Solving the set of equations by elimination of F in Equation 132 yields:

$$-\frac{m_p}{\sin\theta_p}(\ddot{x}_O - \ddot{\theta}_p\,l_c\,\cos\theta_p + \dot{\theta}_p^2\,l_c\,\sin\theta_p) = \frac{m_p}{\cos\theta_p}(g - \ddot{\theta}_p\,l_c\,\sin\theta_p - \dot{\theta}_p^2\,l_c\,\cos\theta_p) \tag{133}$$

$$\Downarrow$$

$$\ddot{x}_O\cos\theta_p - \ddot{\theta}_p\,l_c\,\cos^2\theta_p + \dot{\theta}_p^2\,l_c\,\sin\theta_p\cos\theta_p = -g\sin\theta_p + \ddot{\theta}_p\,l_c\,\sin^2\theta_p + \dot{\theta}_p^2\,l_c\,\sin\theta_p\cos\theta_p \tag{134}$$

$$\Downarrow$$

$$\ddot{x}_O\cos\theta_p = -g\sin\theta_p + \ddot{\theta}_p\,l_c \tag{135}$$

Which can be linearized by assuming that the system only operates within small values of θ_p;

$$\cos\theta_p \approx 1 \quad , \quad \sin\theta_p \approx \theta \quad , \quad \dot{\theta}_p^2 \approx 0 \tag{136}$$

$$\Downarrow$$

$$\ddot{x}_O = -g\theta_p + \ddot{\theta}_p\,l_c \tag{137}$$

The overall input of the inverted pendulum is the wheel acceleration. Hence, $\ddot{x}_O$ is redefined as:

$$\ddot{x}_O = -r\,\ddot{\theta}_w \tag{138}$$

Where $\ddot{\theta}_w$ is the wheels' angular acceleration. The complete mathematical model of the inverted pendulum is:

$$-\,r\,\ddot{\theta}_w = -g\theta_p + \ddot{\theta}_p\,l_c \tag{139}$$

Transfer Function
Laplace transform of the mathematical model is performed to obtain the transfer function of the differential drive mode;

$$-\,r\,\theta_w\,s^2 = -g\,\theta_p + l_c\,\theta_p\,s^2 \tag{140}$$

$$\Downarrow$$

$$\frac{\theta_p}{\theta_w\,s^2} = \frac{r}{-l_c\,s^2 +\, g} \tag{141}$$

$$\Downarrow$$

$$\frac{\theta_p}{\theta_w\,s^2} = \frac{-r/l_c}{s^2 - g/l_c} \tag{142}$$

Note that the selected input of the system is the angular acceleration of the wheels.

State Space Model
The mathematical model is transformed into state space canonical form in the same way as described in Chapter 3.5.1. The states of the system are set up as:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta_p \\ \dot{\theta}_p \end{bmatrix} \tag{143}$$

$$\dot{x}_1 = \dot{\theta}_p \Rightarrow \dot{x}_1 = x_2 \tag{144}$$

$$\dot{x}_2 = \ddot{\theta}_p \Rightarrow \dot{x}_2 = \frac{g}{l}\cdot x_1 - \frac{r}{l_c}\cdot u \tag{145}$$

Where the input, u, and the output, y, of the system are:

$$u = \ddot{\theta}_w \quad , \quad y = \theta_p \tag{146}$$

Which yields:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ g/l_c & 0 \end{bmatrix} \tag{147}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ -r/l_c \end{bmatrix} \tag{148}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{149}$$

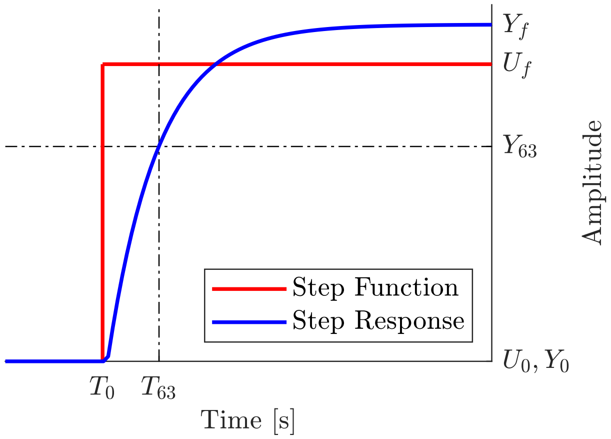
$$\mathbf{D} = \begin{bmatrix} 0 \end{bmatrix} \tag{150}$$

3.5.4 Experimental Approaches

The BLDC motors used in this project have limited information on the internal electrical system in terms of torque and velocity constants. Therefore, experimental approaches are necessary to retrieve the dynamics of the motors.

Step Response Estimation

Step response estimation is used to estimate a first-order transfer function as a simplified DC motor system when the electrical time constant, τ_e, is much smaller than the mechanical time constant, τ_m [16]. A step response is performed by applying a certain step input to a system and comparing it to the corresponding output signal [17]. Figure 65 shows the necessary values extracted from a step response in order to obtain a transfer function of the system.



Where:

T_0 = Time of step input
T_{63} = Time at 63% of output amplitude
U_0 = Initial input
Y_0 = Initial output
Y_{63} = Output amplitude at 63%
U_f = Final input amplitude
Y_f = Final output amplitude

Figure 65: Step Response

The first order transfer function is set up as follows:

$$G(s) = \frac{K}{\tau\, s + 1} \tag{151}$$

Where the system parameters are calculated by utilizing the values from the step response;

$$K = \frac{Y_f - Y_0}{U_f - U_0} \tag{152}$$

$$\tau = T_{63} - T_0 \tag{153}$$

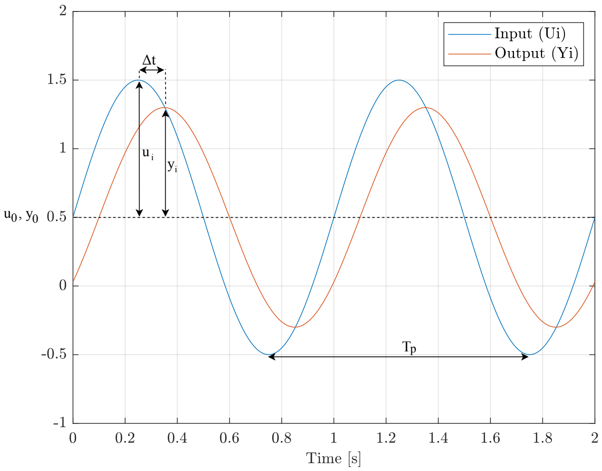
Frequency Response Estimation

Several frequency response experiments can be applied on a system to estimate a transfer function that describes its dynamic behavior. Such an experiment consists of applying a sinusoidal wave with a certain frequency as input to a system and then measuring the output sinus. The input and output sinusoidal waves are described as in Equation 154 and 155.

$$U_i = u_0 + u_i \cdot \sin\left(\omega_i \cdot t\right) \tag{154}$$

$$Y_i = y_0 + y_i \cdot \sin\left(\omega_i \cdot t + \phi\right) \tag{155}$$

Figure 66 shows the necessary measurements extracted from a frequency response experiment at a single frequency.



Where:

Δt = Time delay between the waves
u_0 = Offset of the input wave
y_0 = Offset of the output wave
u_i = Amplitude of input wave
y_i = Amplitude of output wave
T_p = Period of the waves.

Figure 66: Input and output wave

The measurements are then used to find the phase shift in degrees and magnitude in decibel. These Equations are shown in 156 and 157. This experiment is then repeated for multiple frequencies relevant for the system to work at so that the system dynamics can be represented as a bode plot. [18]

$$\phi_i(\omega_i) = -\frac{\Delta t}{T_p} \cdot 360 = -\frac{t_{out} - t_{in}}{T_p} \cdot 360 \;\; [deg] \tag{156}$$

$$G_i(\omega_i) = 20 \cdot log\left(\frac{y_i}{u_i}\right) = 20 \cdot log\left(\frac{\frac{y_{max}-y_{min}}{2}}{\frac{u_{max}-u_{min}}{2}}\right) \;\; [dB] \tag{157}$$

3.6 Control Theory

This chapter covers the theory of system control. There are multiple possible methods for designing feedback control systems, like frequency domain, root locus, and state space [19]. However, in this case, the chosen approach is to design a balancing controller by using state-space. This is because a full state-feedback controller allows for placing all the closed loop poles uniquely with powerful analytical techniques [20]. A standard state space form is represented in Figure 67 and Equations 158 and 159.

$$\mathbf{\dot{x}(t)} = \mathbf{A} \cdot \mathbf{x(t)} + \mathbf{B} \cdot \mathbf{u(t)} \tag{158}$$

$$\mathbf{y(t)} = \mathbf{C} \cdot \mathbf{x(t)} + \mathbf{D} \cdot \mathbf{u(t)} \tag{159}$$

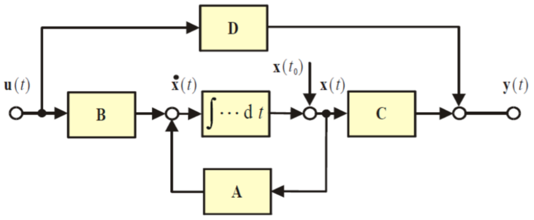


Figure 67: State space plant [2]

$\mathbf{u(t)}$ = Input vector r x 1
$\mathbf{y(t)}$ = Output vector m x 1
$\mathbf{x(t)}$ = State vector n x 1
$\mathbf{A}$ = System matrix n x n
$\mathbf{B}$ = Input matrix n x r
$\mathbf{C}$ = Output matrix m x n
$\mathbf{D}$ = Feedforward matrix m x r

3.6.1 Controllability And Observability

A system is fully controllable if an input can achieve a desired state for each of the states of the system. In the same way, a system is fully observable if the output and input can be measured to find the initial states $x(t_0)$. To find out if the system has full controllability and observability, the Kalman Criteria can be applied. This criteria is shown mathematically for both controllability and observability in Equation 160 and 161 [20].

$$rank[\mathbf{C_r}] = rank\begin{bmatrix}\mathbf{B} & \mathbf{AB} & \dots & \mathbf{A^{n-1}B}\end{bmatrix} = n \tag{160}$$

$$rank[\mathbf{O_b}] = rank\begin{bmatrix}\mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA^{n-1}}\end{bmatrix} = n \tag{161}$$

3.6.2 State Observer

To be able to get an estimate of states that are not physically possible or difficult to measure, an observer can be applied if the system is fully observable. Since the system is assumed to be mainly deterministic, a Luenberger state observer is used. Such an observer is built from system dynamics and the observer gain L, as shown in Figure 68.

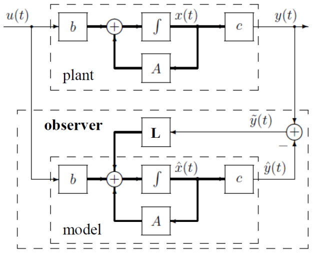


Figure 68: State space system with Luenberger state observer [2]

The observer error is shown in Equation 162. and the state equations of for the observer are shown in Equation 163 and 164. The observer's corresponding error dynamics are based on Equations 162,163,164 and the previous state equations where the matrix **D** is assumed to be zero [20]. Equation 165 shows the general error dynamics.

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} \tag{162}$$

$$\dot{\hat{\mathbf{x}}} = \mathbf{A} \cdot \hat{\mathbf{x}} + \mathbf{B} \cdot \mathbf{u} + \mathbf{L} \cdot (y - \hat{y}) \tag{163}$$

$$\hat{\mathbf{y}} = \mathbf{C} \cdot \hat{\mathbf{x}} \tag{164}$$

$$\dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}} \Rightarrow \dot{\mathbf{e}} = (\mathbf{A} - \mathbf{L} \cdot \mathbf{C}) \cdot \mathbf{e} \tag{165}$$

If $(\mathbf{A} - \mathbf{L} \cdot \mathbf{C})$ is stable, meaning it has all its eigenvalues on LHP, the error (e) will approach zero exponentially [21].

3.6.3 Pole Placement

There exist multiple methods to place the desired closed-loop poles of the systems. However, a requirement for all of them is that the system is controllable as discussed in Chapter 3.6.1. In this chapter, the Bass-Gura approach and the linear quadratic regulator (LQR) will be described.

The effects of different pole placements on the real and imaginary axis in the pole-zero map are shown in Figure 69, where the most dominant poles are the ones closest to the imaginary axis. It is seen that the poles are desired to be on the left-hand side of the imaginary axis to be stable, and also to lay as close to the real axis as possible to reduce oscillations.

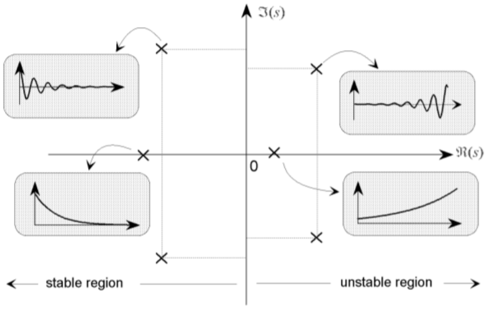


Figure 69: Homogeneous response from different pole locations[3]

Bass-Gura:
First, the characteristic equation of A is formed as seen in Equation 166 to identify all values for $a_{1...n}$. Then the Toeplitz matrix (T) is constructed as seen in Equation 167.

$$|sI - A| = s^n + a_1 \cdot s^{n-1} + a_2 \cdot s^{n-2} + ... + a_{n-1} \cdot s + a_n \tag{166}$$

$$T = \begin{bmatrix} 1 & a_{n-1} & a_{n-2} & \dots & a_2 & a_1 \\ 0 & 1 & a_{n-1} & \dots & a_3 & a_2 \\ 0 & 0 & 1 & \ddots & a_4 & a_3 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 & a_{n-1} \\ 0 & \dots & \dots & \dots & 0 & 1 \end{bmatrix} \tag{167}$$

Then the desired closed loop poles $\mu_1, ..., \mu_n$ are determined to identify all $\alpha_{1...n}$ from the desired characteristic equation as seen in Equation 168.

$$(s - \mu_1)...(s - \mu_n) = s^n + \alpha_1 \cdot s^{n-1} + \alpha_2 \cdot s^{n-2} + ... + \alpha_n \tag{168}$$

The state feedback gain matrix is then calculated in Equation 169. And C_r is the controllability matrix from Equation 160 [2] [22].

$$\mathbf{k} = \begin{bmatrix} (\alpha_n - a_n) & (\alpha_{n-1} - a_{n-1}) & \dots & (\alpha_1 - a_1) \end{bmatrix} \cdot T^{-1} \cdot C_r^{-1} \tag{169}$$

LQR:
It can be difficult to find the desired pole position of the eigenvalues depending on how many states the model consists of. Therefore, a linear quadratic regulator can be used to find the best pole placement solution based on a cost function that penalizes the states and the input of the system. [23]

The cost function can be written as in Equation 170.

$$J = \int_0^{\inf} (x^T \cdot Q \cdot x + u^T \cdot R \cdot u) dt \tag{170}$$

The state cost matrix, Q, is shown in Equation 171, and it penalizes the states based on how fast the states should converge to where they are supposed to be.

$$\mathbf{Q} = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & q_n \end{bmatrix} \tag{171}$$

The input cost matrix is shown in Equation 172, and it penalizes the input based on how aggressive it is allowed to be.

$$\mathbf{R} = \begin{bmatrix} r_1 & 0 & 0 & 0 \\ 0 & r_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & r_r \end{bmatrix} \tag{172}$$

3.6.4 Steady State Error and Robust Control

To achieve the desired steady-state accuracy, a prefilter is included since state-feedback alone is not sufficient enough if the reference is different from zero. A steady-state error equal to zero is desired, meaning $y(t) \approx w(t)$ as $t \to \infty$. See Figure 70 for illustration.

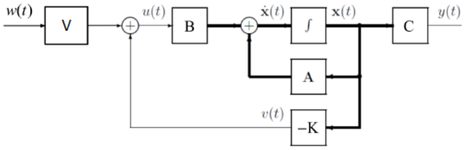


Figure 70: Prefilter; V included with the state-feedback gain; K. [2]

Figure 70 is used to derive the following equations [2]:

$$\dot{\mathbf{x}}(t) = (A - B \cdot K) \cdot \mathbf{x}(t) + B \cdot V \cdot w(t) \tag{173}$$

$$y(t) = C \cdot \mathbf{x}(t) \tag{174}$$

The steady-state requirements, $\dot{\mathbf{x}}(t) = 0$ and $y(t) = w(t)$ is then applied:

$$(A - B \cdot K) \cdot \mathbf{x}(t) + B \cdot V \cdot w(t) = 0 \tag{175}$$

$$y(t) = w(t) = C \cdot \mathbf{x}(t) \tag{176}$$

Equation 175 is solved for $x(t)$ and is replaced in Equation 176. This substitution yields the following:

$$y(t) = w(t) = C \cdot (B \cdot K - A)^{-1} \cdot B \cdot V \cdot w(t) \tag{177}$$

Because of the requirement of $y(t) = w(t)$ the following must be true, where I is the identity matrix:

$$I = C \cdot (B \cdot K - A)^{-1} \cdot B \cdot V \tag{178}$$

Finally, the prefilter V is shown in Equation 179.

$$V = (C \cdot (B \cdot K - A)^{-1} \cdot B)^{-1} \tag{179}$$

To increase the robustness of the control structure, an integral state can be added in addition to the state-feedback and prefilter. The integral state uses the error between input and output which can counteract external disturbances and account for uncertainties in the plant model, thus giving the system a more robust control even if the states are estimated through an observer. See Figure 71 for how this integral state is integrated into the control system.

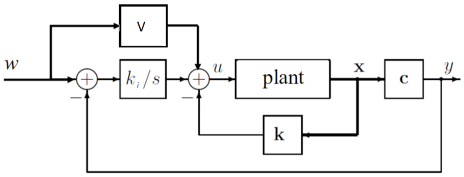
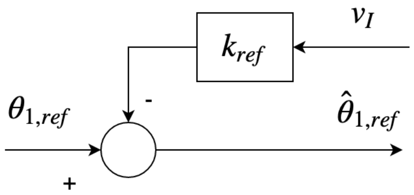


Figure 71: Integral control with prefilter. [2]

3.6.5 Variable Setpoint Control

In order to obtain systems that do not saturate or drift undesirably, variable pitch setpoint control can be added in cascade to the overall control structure. This is necessary due to that the state space systems concerning the inertia wheel balancing and the differential drive balancing may drift and saturate if subjected to noise over time.

Inertia Wheel Velocity Compensation
When subjected to noise or external forces such as centrifugal force, the inertia wheel balancing will accelerate in a single direction until it is stabilized. For both safety and convenience purposes, it is desired that the inertia wheel spins with a velocity close to zero. The variable setpoint control can be implemented by utilizing a gained signal of the velocity, offsetting the setpoint to reduce the velocity of the inertia wheel, as illustrated in Figure 72.

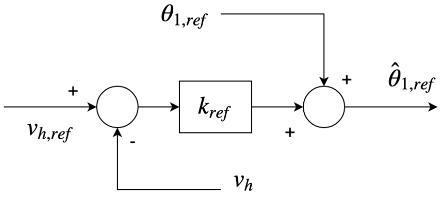


Where:

$\theta_{1,ref}$ = Robot pitch setpoint
$\hat{\theta}_{1,ref}$ = Altered robot pitch setpoint
v_I = Inertia wheel velocity
k_{ref} = Inertia wheel velocity gain

Figure 72: Offsetting the setpoint

Hub Wheel Velocity Control
Also while balancing with the differential mode, variable setpoint can be implemented. The purpose of this is for the robot to stand still in the same location over time without drifting out of place when the input velocity is zero. Also when it is desired that the robot should move in a certain direction, a velocity reference can be set to offset the input angle in a direction that will make the robot move with a desired velocity. This control is illustrated in Figure 73.



Where:

$\theta_{1,ref}$ = Robot pitch setpoint
$\hat{\theta}_{1,ref}$ = Altered robot pitch setpoint
$v_{h,ref}$ = Velocity setpoint
v_h = Translational velocity
k_{ref} = Error gain

Figure 73: Hub Wheels Velocity Control

4 Method

This chapter describes how the theory is implemented on the physical service robot. Figure 74 gives an overview of the approach used for controlling the system. It is again specified that due to time limitations that the front wheel balancing control is not implemented on the robot. Hence, when referring to the "bicycle system", it implies using the inertia wheel system for balancing unless otherwise specified.

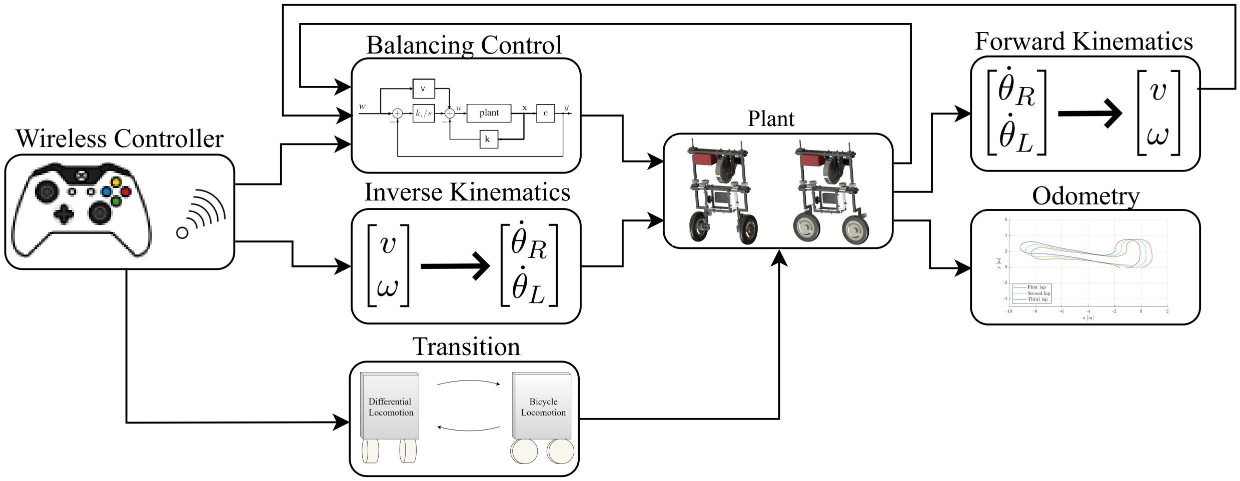

Balancing Control
w
v
k_i/s
u
plant
x
c
y
k
Wireless Controller
Inverse Kinematics
$\begin{bmatrix} v \\ \omega \end{bmatrix} \longrightarrow \begin{bmatrix} \dot{\theta}_R \\ \dot{\theta}_L \end{bmatrix}$
Transition
Differential Locomotion
Bicycle Locomotion
Plant
Forward Kinematics
$\begin{bmatrix} \dot{\theta}_R \\ \dot{\theta}_L \end{bmatrix} \longrightarrow \begin{bmatrix} v \\ \omega \end{bmatrix}$
Odometry
First lap
Second lap
Third lap
x [m]
y [m]

Figure 74: Overview of the system control

As seen in Figure 74, the wireless controller is used to control both locomotion configurations and to transition between the configurations with a button. The balancing control structure is based on the dynamics of the system to stabilize the robot. Kinematics are used to control the movements of the robot and to control the variable setpoint depending on the configuration. Odometry is applied to lay the groundwork for estimation of the robots pose. All of these elements are implemented appropriately to work in accordance with the plant.

4.1 Interference Fits

As discussed earlier in Chapter 2.3.3 interference fits are used between the bearings and the shaft and between the bearings and brackets. An interference fit is assembled with either pressing the parts together or shrink fit through temperature differences between the components. However, because of a lack of available machines for shrink fits, press fits were used. An illustration with some relevant forces and geometric parameters is shown in Figure 75 and is used together with the part drawings in Appendix A.4 for calculations in the next subchapters.

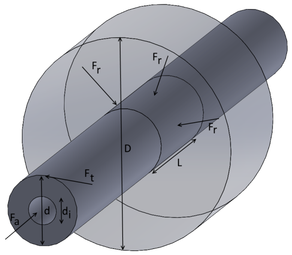


Figure 75: Forces and geometric parameters relevant for an interference fit

F_a	Axial force
F_r	Radial force between the shaft and hub
F_t	Tangential force
D	Outer diameter of the hub
d	Common diameter of the shaft and hub
d_i	Inner diameter of the shaft

4.1.1 Axial force requirement

The axial force requirement to loosen a interference fit can be set up as in Equation 180 based on Figure 75.

$$F_a = \mu_s \cdot F_r = \mu_s \cdot P_{min} \cdot (\pi \cdot d \cdot L) \tag{180}$$

The minimum pressure required for a certain press force not to loosen the interference fit is shown in Equation 181. Where $\mu_s \approx 0.1$ since it is a press fitted assembly in a static situation. [24] And $F_a \approx 100N$ so that each of the the two ball bearings can absorb all axial forces from the inertia wheel systems weight, which should be well below 10kg. This yields a conservative axial force requirement. The minimum bore pressure necessary to hold an axial load of 100N is then calculated in Equation 182 and 183.

$$P_{min} = \frac{F_a}{\mu_s \cdot \pi \cdot d \cdot L} \tag{181}$$

Minimum pressure between bearing and shaft:

$$P_{min} = \frac{F_a}{\mu_s \cdot \pi \cdot d \cdot L} = \frac{100}{0,1 \cdot 12 \cdot 10^{-3} \cdot 8.8 \cdot 10^{-3}} = 9.5Mpa \tag{182}$$

Minimum pressure between bearing and bracket:

$$P_{min} = \frac{F_a}{\mu_s \cdot \pi \cdot d \cdot L} = \frac{100}{0,1 \cdot 32 \cdot 10^{-3} \cdot 8.8 \cdot 10^{-3}} = 3.6Mpa \tag{183}$$

4.1.2 Engineering fits

Bearing and shaft:
The geometric properties used for the shaft in Equation 184 and the bearing in Equation 185 is specified through the part drawings and in Figure 75. The bearing and shaft steel is approximated to both having a Young's modulus of 208GPa and a Poission's ratio of 0.3. [25]

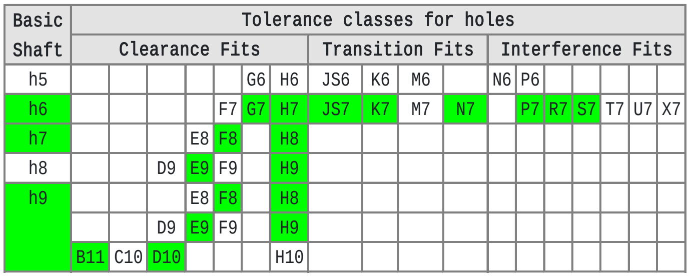
$$\alpha_s = \frac{d}{E_s} \cdot \left(\frac{d^2 + d_i^2}{d^2 - d_i^2} - v \right) = \frac{12 \cdot 10^{-3}}{208 \cdot 10^9} \cdot \left(\frac{(12 \cdot 10^{-3})^2 + (0)^2}{(12 \cdot 10^{-3})^2 - (0)^2} - 0,3 \right) = 4,038 \cdot 10^{-14} \tag{184}$$

$$\alpha_b = \frac{d}{E_b} \cdot \left(\frac{D^2 + (d)^2}{D^2 - (d)^2} + v \right) = \frac{12 \cdot 10^{-3}}{208 \cdot 10^9} \cdot \left(\frac{(18 \cdot 10^{-3})^2 + (12 \cdot 10^{-3})^2}{(18 \cdot 10^{-3})^2 - (12 \cdot 10^{-3})^2} + 0,3 \right) = 1,673 \cdot 10^{-13} \tag{185}$$

The minimum interference necessary to achieve a minimum pressure of P_{min} is then as shown in Equation 186. [24]

$$\delta_{min} = P_{min} \cdot (\alpha_s + \alpha_b) = 9,5 \cdot 10^6 \cdot (4,038 \cdot 10^{-14} + 1,673 \cdot 10^{-13}) = 1,97 \cdot 10^{-6} m \approx 0,002 mm \tag{186}$$

Figure 76 is used to choose an interference fit. P7/h6 is chosen since it is important to have an accurate and rigid assembly, but not a critical bore pressure requirement. [26] [4]



Basic	Tolerance classes for holes																	
Shaft	Clearance Fits							Transition Fits				Interference Fits						
h5						G6	H6	JS6	K6	M6		N6	P6					
h6					F7	G7	H7	JS7	K7	M7	N7		P7	R7	S7	T7	U7	X7
h7				E8	F8		H8											
h8			D9	E9	F9		H9											
h9				E8	F8		H8											
			D9	E9	F9		H9											
	B11	C10	D10				H10											

Figure 76: Preferable fits of the shaft-basis system (ISO 286-1:2010) [4]

Table 3: Deviation and max/min interference for a P7/h6 fit and a common diameter of 12mm

Fit	Deviation	$\delta_{\mathbf{max}}/\delta_{\mathbf{min}}$
P7	-0,011 ... -0,029mm	0,029mm/0mm
h6	0,000 ... -0,011mm	

As seen from Table 3 the smallest interference is 0mm which obviously does not satisfy the roughly estimated 0,002mm requirement, however it is highly unlikely that the fit will end up that low in the interference range. Also, since the axial force requirements of 100N per bracket is quite conservative, this is not seen as a probable issue.

$$P_{max} = \frac{\delta_{max}}{\alpha_s + \alpha_b} = \frac{0,029 \cdot 10^{-3}}{(4,038 \cdot 10^{-14} + 1,673 \cdot 10^{-13})} = 140 MPa \tag{187}$$

$$F_a = \mu_d \cdot P_{max} \cdot (\pi \cdot d \cdot L) = 0.09 \cdot 140 \cdot 10^6 \cdot \pi \cdot 12 \cdot 10^{-3} \cdot 8,8 \cdot 10^{-3} = 4180 N \tag{188}$$

The maximum interference gives a pressure of 140Mpa as shown in Equation 187, which is well below the typical steel yield strength of 350MPa which means that it is unlikely to get occurrences of plastic deformations. [27] To assemble the scenario with maximum interference an axial press force of 4180N is required, as seen from Equation 188.

Bearing and brackets:
The same approach is used for the bearing and bracket fit as for the bearing and shaft fit shown previously. However, first the Young's modulus for the composite material is found using the rule of mixtures as shown in Equation 189. [28] The volume and Young's modulus of each material are given in Table 4. [10] [11]

Table 4: Young's modulus and average volume of onyx and kevlar

	Onyx	**Kevlar**
E-modulus [GPa] (compressive)	1,4	28
Volume [%]	75,7	24,3

$$E_{comp} = E_{onyx} \cdot V_{onyx} + E_{kevlar} \cdot V_{kevlar} \tag{189}$$

$$= 1,4 \cdot 10^9 \cdot 0,757 + 28 \cdot 10^9 \cdot 0,243$$
$$= 7,9GPa$$

The bearing calculations are shown in Equation 190 and the bracket calculations are shown in Equation 191. [24]

$$\alpha_b = \frac{d}{E_b} \cdot \left(\frac{d^2 + d_i^2}{d^2 - d_i^2} - v \right) = \frac{32 \cdot 10^{-3}}{208 \cdot 10^9} \cdot \left(\frac{(32 \cdot 10^{-3})^2 + (26 \cdot 10^{-3})^2}{(32 \cdot 10^{-3})^2 - (26 \cdot 10^{-3})^2} - 0,3 \right) = 7,05 \cdot 10^{-13} \tag{190}$$

$$\alpha_{br} = \frac{d}{E_{br}} \cdot \left(\frac{D^2 + (d)^2}{D^2 - (d)^2} + v \right) = \frac{32 \cdot 10^{-3}}{7,9 \cdot 10^9} \cdot \left(\frac{(50 \cdot 10^{-3})^2 + (32 \cdot 10^{-3})^2}{(50 \cdot 10^{-3})^2 - (32 \cdot 10^{-3})^2} + 0,3 \right) = 1,09 \cdot 10^{-11} \tag{191}$$

The minimum interference necessary in order to achieve a minimum pressure of P_{min} is then as shown in Equation 192:

$$\delta_{min} = P_{min} \cdot (\alpha_s + \alpha_b) = 3,6 \cdot 10^6 \cdot (7,05 \cdot 10^{-13} + 1,09 \cdot 10^{-11}) = 4,2 \cdot 10^{-5} m \approx 0,04mm \tag{192}$$

It is assumed that the same interference fit P7/h6 will work in this case as well. The Deviation and max/min interference for a P7/h6 fit and a 32mm common diameter is shown in Table 5.

Table 5: Deviation and max/min interference for a P7/h6 fit and a common diameter of 32mm

Fit	**Deviation**	$\delta_{\mathbf{max}}/\delta_{\mathbf{min}}$
P7	-0,017 ... -0,042mm	0,042mm/0,001mm
h6	0,000 ... -0,016mm	

Now the interference requirement is in the other end of the interference range. However, since P7/h6 is chosen mainly to have an accurate and rigid assembly, and a conservative axial force is used, it is considered acceptable.

$$P_{max} = \frac{\delta_{max}}{\alpha_s + \alpha_b} = \frac{0,042 \cdot 10^{-3}}{(7,05 \cdot 10^{-13} + 1,09 \cdot 10^{-11})} = 3,62MPa \tag{193}$$

$$F_a = \mu_d \cdot P_{max} \cdot (\pi \cdot d \cdot L) = 0.09 \cdot 3,62 \cdot 10^6 \cdot \pi \cdot 32 \cdot 10^{-3} \cdot 8,8 \cdot 10^{-3} = 288N \tag{194}$$

The maximum interference gives a pressure of 3,62Mpa as shown in Equation 193, however since the yield strength data of Onyx plastic is unavailable, ABS plastic is used for comparison since it is weaker and less stiff. [29] The lowest yield strength value of molded ABS plastic is larger than the bore pressure of 3,62MPa, so it is considered to be acceptable. [30] To assemble the scenario with maximum interference, an axial press force of 288N is required, as seen from Equation 194.

4.2 Power Supply

The desired battery cell type is Lithium-ion due to its high energy density, which gives the battery pack a relatively low weight and small volume compared to for example a lead battery. It also has a low self-discharge ratio and does not have to be fully discharged before recharging, which is convenient on a service robot [31]. The battery cell model is selected based on the charge, voltage, and current requirements, as shown in Table 6. These requirements were discussed previously in Chapter 3.1. HG2 18650 is the chosen battery cell, where some of its most important characteristics are shown in Table 7. For additional information, see the datasheet [32].

Table 6: Battery pack requirements

Voltage	**Current**	**Capacity**
$\approx 24V$	$\approx 40A$	$\approx 8680mAh$

Table 7: Battery cell characteristics

Type	**Model**	**Nominal voltage**	**Nominal capacity**	**Continuous discharge current**
Lithium-ion	HG2 18650	3.6 V	3000 mAh	20 A

In order to acquire a voltage of approximately 24V, six battery cells are placed in series. And to make sure the battery can deliver enough current continuously and packs enough capacity, three battery cells are placed in parallel to reach a discharge current of 60A and 9000mAh. The battery cells are assembled with battery spacers and nickel-strips as shown in Figure 77 and the assembly characteristics are shown in Table 8.

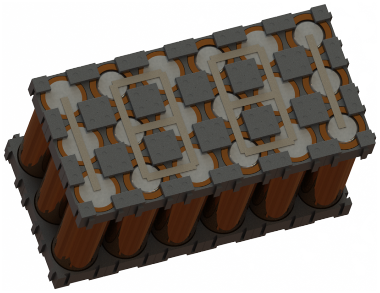


Figure 77: Battery cells assembled

Table 8: Characteristics of the battery cell assembly

Type	**#Cells in series**	**#Cells in parallel**	**Nominal voltage**	**Nominal capacity**	**Continuous discharge current**
Li-ion assembly	6	3	21.6 V	9000 mAh	60 A

In order to make sure the battery cells do not overheat during assembly, a spot welder machine is used. Also, for safety reasons, a 40A fuse is attached to the battery to make sure the robot is not draining more current than allowed. To make sure each battery cell has an even distribution of load during both charging and discharging, a battery management system (BMS) is connected in between the battery cell assembly and charge/discharge connector. The protective casing which the battery is embedded in is shown in Appendix A.5.

4.3 Actuators and Sensors

This chapter will go through the methods of how different components of the robot are set up and how the adjacent theory is implemented using Virtual Instruments (VIs) in LabVIEW. Figure 78 illustrates how the LabVIEW codes are generally presented:

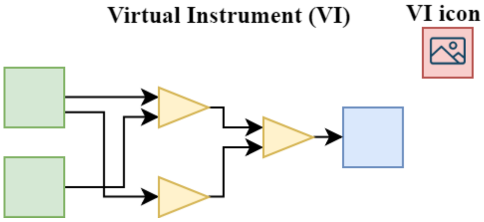


Figure 78: VI Template

Where the VIs are presented with their respective icons in the top right corner for simple navigation through the final LabVIEW code attached in Appendix B.

4.3.1 VESC®

Vedder Electronic Speed Controllers (VESCs) are universal motorø controllers specifically made for brushless DC motors that can be communicated with using UART, CAN bus, or digital control. Through an open source software, *VESC Tool* [33], the motors can be configured according to their internal dynamics, and the inputs can be defined based on the user's preferences. Figure 79 shows the Maytech VESC, which is used in this project.

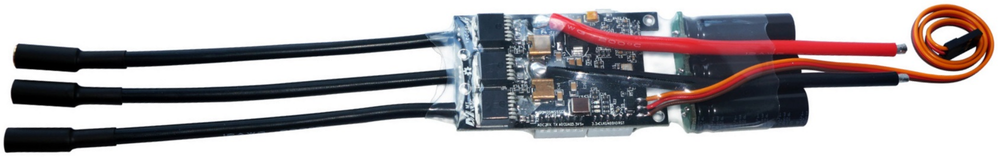


Figure 79: Maytech VESC [5]

Motor Settings

As previously described there is a total of three BLDC motors in the system; two hub wheels in contact with the ground, and one motor for the inertia wheel. The motors are set up with field oriented control (FOC) in the VESC Tool. In short, FOC forces the resultant force vector of a three-phase motor to be tangential to the rotor and thereby optimizing the torque and efficiency of the motor. The FOC is automatically tuned and optimized through motor identification in VESC Tool, and will therefore not be elaborated on. Table 9 lists the identified motor parameters from VESC Tool.

Table 9: Motor Parameters

	Hub wheel 1	Hub wheel 2	Inertia motor
Armature Resistance $[m\Omega]$	172.73	168.8	46.89
Armature Inductance $[\mu H]$	194.0	190.9	44.33
Flux Linkage $[mWb]$	19.54	19.33	8.839

App Settings

The app settings define the input signals to the motors. For this purpose, the digital pulse-position modulation (PPM) setting is selected based on the simplicity of the input signal and that the myRIO does not directly support CAN bus. PPM has direct relations to PWM which can be controlled through the PWM I/O of the myRIO, and by experimental trials, the PWM signal is mapped through VESC Tool according to Table 10. [34]

Table 10: PWM to PPM Mapping

PWM frequency	PWM LB	PWM UB	PPM LB	PPM UB
400 Hz	0.1	0.9	0.1670 ms	2.3620 ms

Where LB and UB indicate lower and upper boundaries, respectively. Figure 80 illustrates the PPM mapping with a linear approximation. For convenience, the PWM signal is further mapped from a signal between -1 to 1 in the LabVIEW program, where -1 is full speed in negative direction, and 1 is full speed in positive direction. This input signal will be referred to as u throughout the report and is mapped as illustrated in Figure 81.

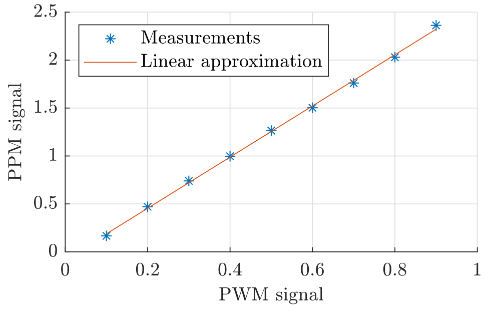


Figure 80: PPM mapping

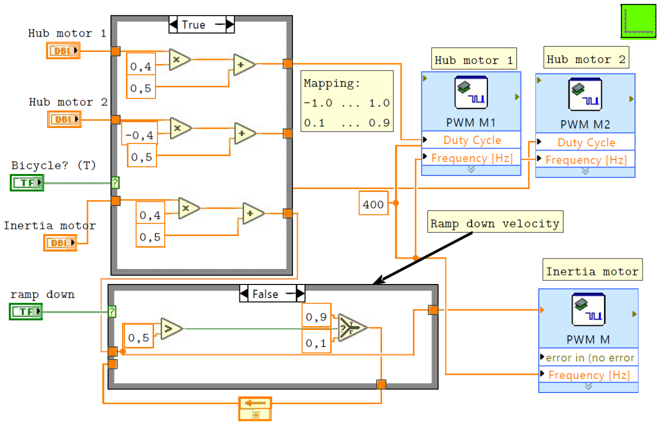


Figure 81: PWM scaling.vi

Essentially, the input signal, u, between -1 and 1 is proportional to the voltage that is supplied to the motors. Notice also the case structure in the bottom of Figure 81 which safely ramps the velocity of the inertia motor down if the user chooses to turn off inertia wheel balancing.

The remaining settings applied to the VESCs' input settings are listed in Table 11. Note that the values marked with "minimum" in the table are set to the lowest value allowed by the VESC Tool to provide an opportunity to accelerate the motors desirably with minimal restrictions.

Table 11: Other input settings

Configuration	Duty Cycle
Deadband	5 % (minimum)
Median Filter	Off
Safe Start	Off
Positive Ramping Time	0.01 s (minimum)
Negative Ramping Time	0.01 s (minimum)

4.3.2 Dynamixel MX-64 Servo Actuators

Dynamixel actuators are high-performance servo actuators with integrated DC motors, controllers and drivers. They are specifically designed to be used in robots and mechanical structures that require high accuracy with low space consumption [35]. The Dynamixel MX-64 used on the robot is illustrated in Figure 82.

Figure 82: Dynamixel MX-64 [36]

Setup
The setup of the Dynamixel MX-64 Robot Servo Actuators is done through a Dynamixel Wizard in the software RoboPlus [37]. To correctly set up the actuators, each actuator must have a unique ID, the baud rate must be within the specifications, and the angle limits must be correct. The specifications of the Dynamixel servos are shown in Appendix F.1. If needed, the actuators can also be calibrated in the Dynamixel Wizard application. These Dynamixel servos utilize TTL serial communication, which is controlled through the myRIO using a USB adapter as illustrated in Figure 83.

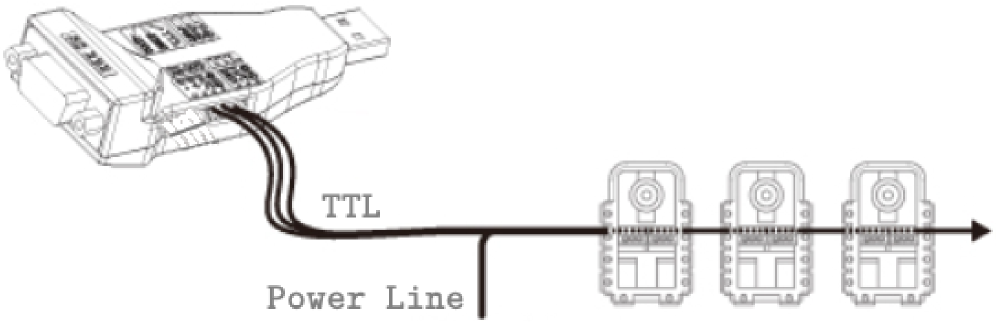


Figure 83: Multiple Dynamixels in series with USB adapter

The relevant settings for the two Dynamixel servos in the system are listed in Table 12.

Table 12: Dynamixel settings

Dynamixel ID	1	2
Baud Rate	9615	9615
Max Torque	100%	100%

Interfacing

In order to control the Dynamixel servo actuators, an example from the National Instruments website is downloaded as a template [38]. However, this template is made for earlier versions of the Dynamixel servo motors, and minor changes are therefore made to it.

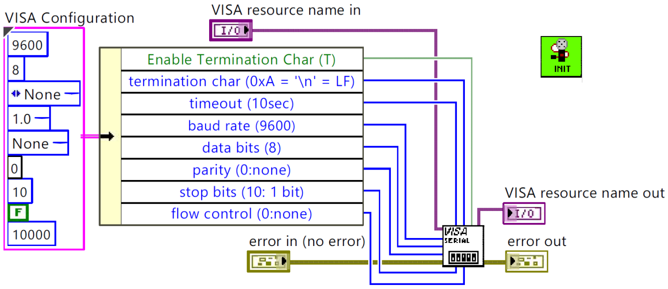


Figure 84: Initialize Dynamixel.vi

The dynamixel program is initialized as illustrated in Figure 84 by configuring the myRio's USB serial port. The baud rate is set to 9600 which is supported by the myRio, and corresponds to be sufficiently within the required 3% tolerance of the baud rate defined in the Dynamixel Wizard application [39].

The main program, illustrated in Figure 85, allows for simple control of the Dynamixel servos. The desired angles of the Dynamixels are specified through an input cluster that writes to the pre-defined Dynamixel IDs. Figure 86 shows an example of input parameters of the program.

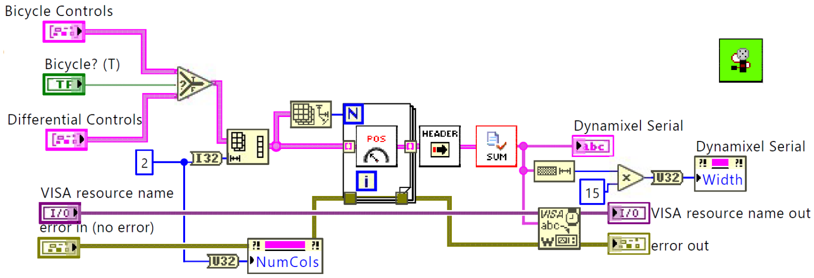


Figure 85: Main Dynamixel.vi

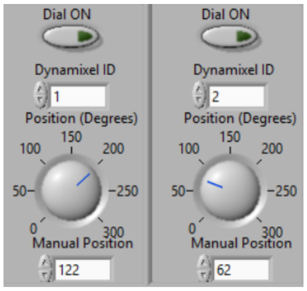


Figure 86: Dynamixel input parameters

4.3.3 Quadrature Encoders

Encoder values are extracted from the myRIO using dedicated encoder inputs that read encoder pulses independently from the program loop time. The encoder data is calculated into kinematic parameters as illustrated in Figure 87.

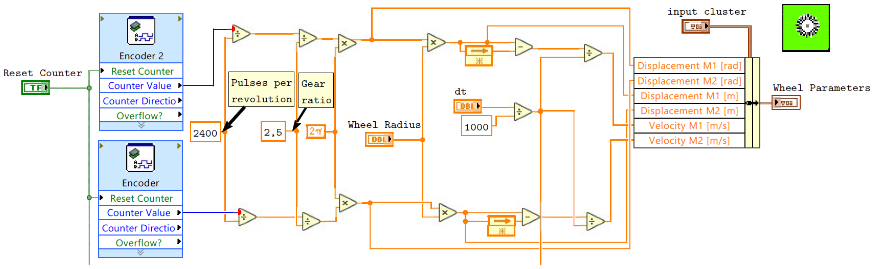
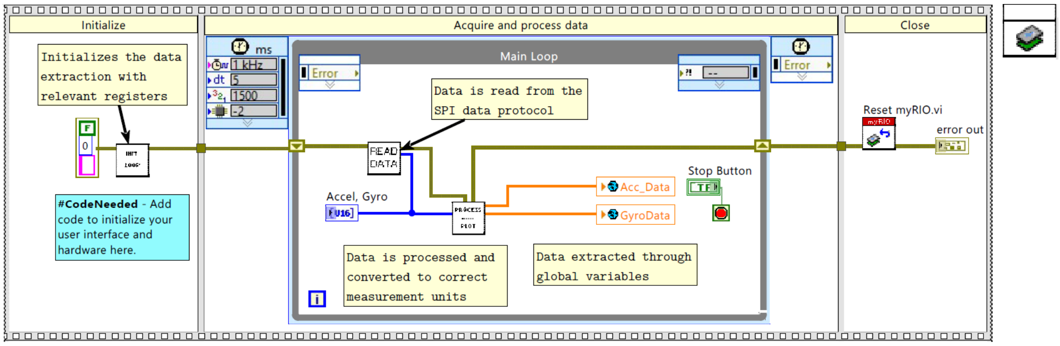


Figure 87: Read Encoders.vi

4.3.4 Inertial Measurement Unit

Accelerometer and gyroscope data are extracted from the IMU using Serial Peripheral Interface (SPI). In short, SPI bus is a synchronous data protocol that utilizes a serial clock signal (SCK) and two data transmission lines; Master In Slave Out (MISO) and Master Out Master In (MOSI) [40]. In LabVIEW, a program from the National Instruments forums made specifically for MPU9250 is used [41]. Figure 88 shows the program used to extract data from the IMU, where the different sub-VIs in the program are explained.



Initializes the data extraction with relevant registers

#CodeNeeded - Add code to initialize your user interface and hardware here.

Data is read from the SPI data protocol

Data is processed and converted to correct measurement units

Data extracted through global variables

Figure 88: SPI_main.vi

The data extracted from this VI is a number between -1 and 1 for the accelerometer data to indicate its magnitude relative to gravity and in degrees per second for the gyroscope data. These measurements are subsequently inserted in the complementary filter illustrated in Figure 89.

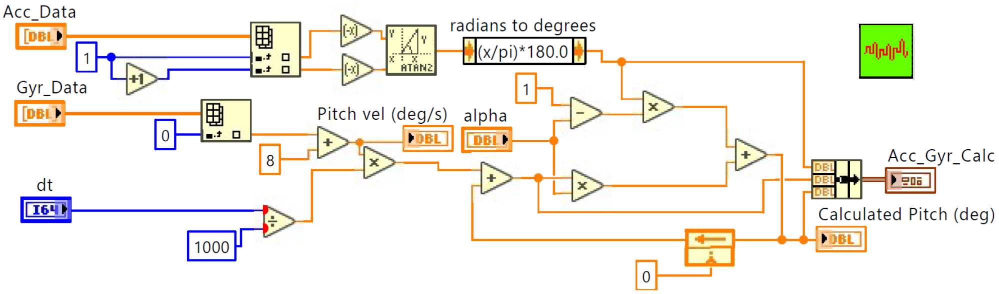


Figure 89: ComplementaryFilter.vi

Notice that a *atan2* function is utilized instead of the conventional *atan* in order to include all four quadrants of the unit circle and avoid ambiguous solutions. This code is built based on the theory from Chapter 3.3.2, where α is tuned to optimize the calculated pitch angle of the robot in terms of elimination of noise and phase delay, and by experimental trials is determined to be $\alpha = 0.999$.

Since the center of mass may have a small pitch offset relative to the wheels' contact point, the IMU will have an initial offset angle. Therefore, the pitch angle is re-defined by subtracting experimentally determined offset angles, as illustrated in Figure 90. This offset is different between the bicycle and differential drive configuration because the center of mass is shifted as it transforms.

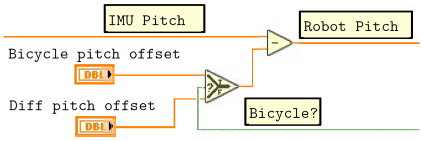


Figure 90: Re-defining pitch

4.4 System Identification

The dynamic systems of the robot are identified by utilizing the theory described in Chapter 3.5. During the system identification, it was discovered that the direct dynamic systems between the input signal and the robot's pitch angle, i.e. the output of the systems, would contain a zero in the origin of the complex plane. The consequence of this is that the theory from Chapter 3.6 does not apply [42]. This is solved by instead of directly controlling the input signal, u, the change in input signal, $\dot{u}$, is controlled as illustrated from Figure 91 to Figure 92.

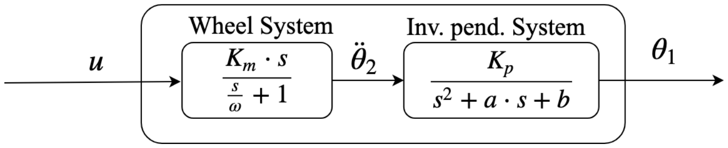

u
Wheel System
$\frac{K_m \cdot s}{\frac{s}{\omega}+1}$
$\ddot{\theta}_2$
Inv. pend. System
$\frac{K_p}{s^2+a\cdot s+b}$
θ_1

Figure 91: Dynamic plant with input of u

$\Downarrow$

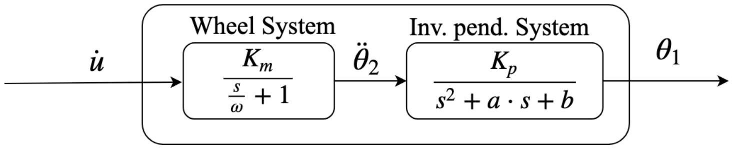


Figure 92: Dynamic system with input of $\dot{u}$

Subsequently, an integrator will become a part of the controller as shown in Figure 93, which is implemented in LabVIEW using Euler integration as illustrated in Figure 94.

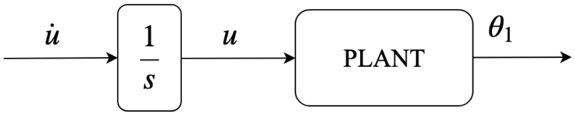


Figure 93: Plant with integrator

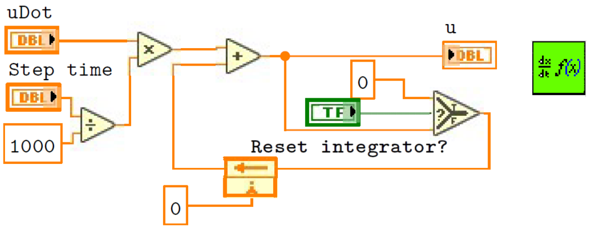


Figure 94: InputIntegration.vi

Therefore, the dynamic systems derived in this chapter will have inputs of $\dot{u}$, which corresponds to a change in voltage, and the pitch angle as outputs.

4.4.1 Bicycle System

The full bicycle system is the combination of the dynamic systems of the inertia wheel system and the bicycle's mechanical system.

Mechanical System

The mechanical system is defined by identifying the parameters of the state space system derived in Chapter 3.5.1. This is done using the "Mass Properties" tab in *SOLIDWORKS* on the 3D model to provide precise values that correspond with the physical system. Table 13 lists the identified parameters relevant for the bicycle dynamics.

Table 13: Identified bicycle parameters

Parameter	**Value**	**Parameter**	**Value**
m_1 [kg]	15.425	J_{1A} [kgm^2]	3.4949
m_2 [kg]	5.567	J_{2B} [kgm^2]	0.05485
L_1 [m]	0.398	g [$\frac{\mathrm{m}}{\mathrm{s}^2}$]	9.81
L_2 [m]	0.64	-	-

The resulting state space system representing G_m for the bicycle system is:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 35.5371 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ -0.0205 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad \mathbf{D} = [0] \tag{195}$$

The bode diagram of this system is illustrated in Figure 95.

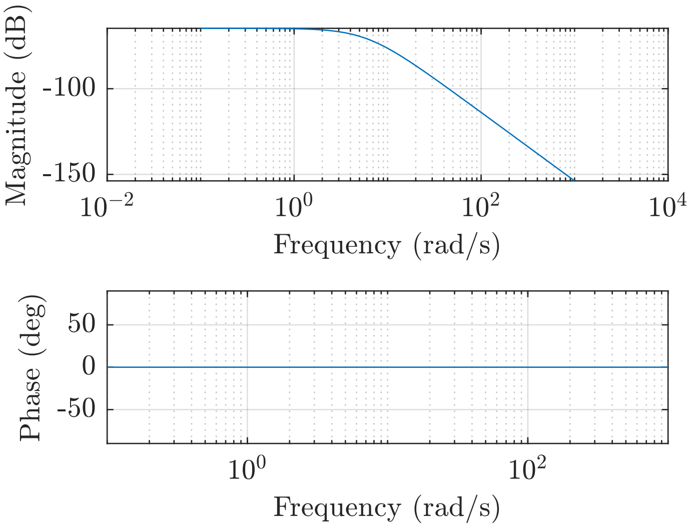


Figure 95: Bode diagram of mechanical bicycle system

Where the input and output of the system are the acceleration of the wheels and the pitch angle of the robot, respectively:

$$u = \ddot{\theta}_2 \quad , \quad y = \theta_1 \tag{196}$$

Inertia Wheel System

The inertia wheel system describes the dynamics between the input signal, u, that goes from -1 to 1, as discussed in Chapter 4.3.1 and the acceleration of the inertia wheel. This system is identified by performing a frequency response experiment on the inertia wheel system and reading out the resulting angular position from encoder data through LabVIEW according to the theory from Chapter 3.5.4. The angular position is less noisy than the angular accelerometer data and is therefore preferred to measure when making the model. The resulting data is shown in Table 14 with its corresponding bode plot in red in Figure 96.

Table 14: Identified frequency response parameters

Freq [rad/s]	Mag [dB]	Phase [deg]
0,1	57,7	-89,8
0,25	49,8	-90,9
0,5	43,8	-92,8
1	37,9	-94,8
2,5	29,9	-101,2
5	23,6	-109,7
7,5	19,7	-118,4
10	16,5	-126,8

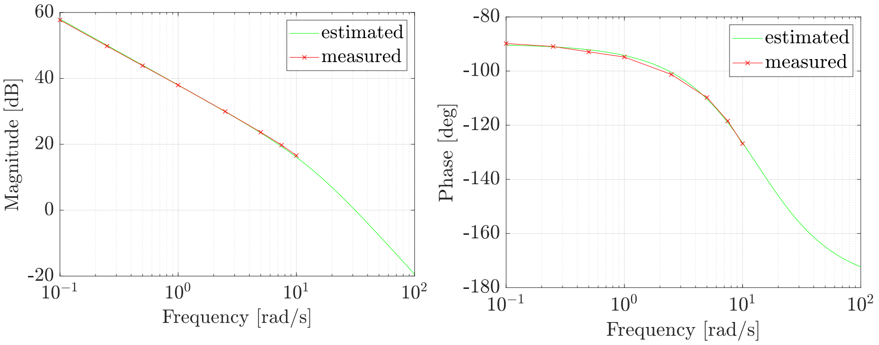


Figure 96: Matching transfer function with a bode plot

The bode plot measured from the frequency response in red is then matched with a transfer function, as shown in Equation 197. The matched transfer function is shown in green in Figure 96.

$$G = \frac{\theta_2}{u} = \frac{K}{s \cdot (\frac{s}{\omega_c} + 1)} = \frac{79,43}{s \cdot (\frac{s}{13,5} + 1)} \tag{197}$$

Where ω_c is the corner frequency [rad/s], and the magnitude can be expressed as: $K = 10^{(\frac{dB}{20})}$.

A frequency response estimation is the selected approach due to its accuracy and that it is easily achieved with an encoder and a simple platform, as seen in Figure 98. Figure 97 shows the results from a open loop step response on the Inertia wheel system, where the measurements from the physical system are in green, and the simulation of the transfer function in Simulink is shown in blue.

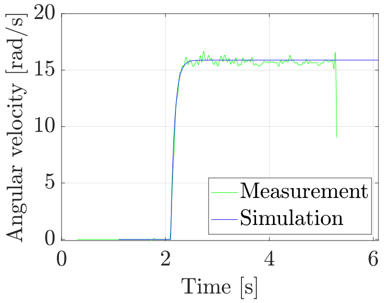


Figure 97: Simulation and measurements of a 20% voltage step input.

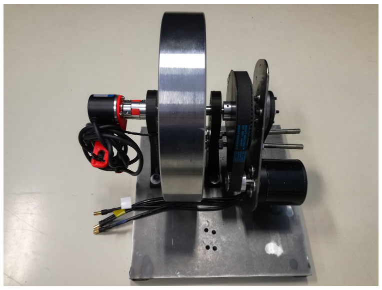


Figure 98: Frequency response setup

The angular position is used to acquire the estimated model to avoid noisy data. Therefore the transfer function must be differentiated twice, as shown in Equation 198, to get the desired transfer function output. To get rid of the free zero in the numerator, an integrator is introduced as described previously. This makes the input of the system a change in input signal which also corresponds to what is desired to control; a torque.

$$G = \frac{\ddot{\theta}_2}{u} = \frac{K}{s \cdot (\frac{s}{\omega_c} + 1)} \cdot s^2 = \frac{1072 \cdot s}{s + 13,5} \tag{198}$$

$$G_{Iw} = \frac{\ddot{\theta}_2}{\dot{u}} = \frac{1}{s} \cdot \frac{1072 \cdot s}{s + 13,5} = \frac{1072}{s + 13,5} \tag{199}$$

This transfer function is then rewritten to state space form as shown in Equation 200. This is done by using the MATLAB function *tf2ss*, which converts transfer functions to state space representation.

$$\mathbf{A} = -13,5 \quad \mathbf{B} = 1 \quad \mathbf{C} = 1072,3 \quad \mathbf{D} = 0 \tag{200}$$

Combined Model

The inertia wheel system and the mechanical system are combined using the MATLAB function *series* according to Figure 99. This function creates a resulting system where the individual states of the subsystems are maintained.

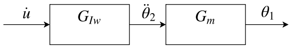


Figure 99: Full plant

The resulting system is:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 35.54 & 0 & -21,96 \\ 0 & 0 & -13,5 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad \mathbf{D} = [0] \tag{201}$$

With the state vector:

$$\mathbf{x} = \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ x_2 \end{bmatrix} \tag{202}$$

Where:

θ_1 = Robot angular pitch
$\dot{\theta}_1$ = Robot angular pitch velocity
x_2 = Inertia wheel angular acceleration state

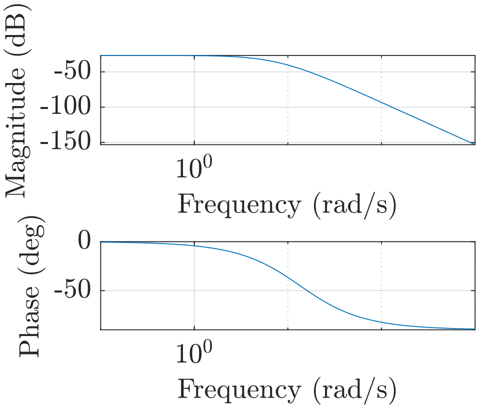


Figure 100: Bode diagram of combined bicycle system

Note that the last state, x_2, corresponds to a value that is proportional to the acceleration of the inertia wheel by a factor equal to the inertia wheels' system C-parameter. The Bode diagram of the full system is illustrated in Figure 100.

4.4.2 Differential Drive System

The full differential system is the combination of the dynamic systems of the hub wheels and the mechanical system described in Chapter 3.5.3.

Mechanical System

Similarly to the bicycle system, the parameters of the differential drive's mechanical system are found using the "Mass Properties" tab in *SOLIDWORKS* and by manual measurement. These parameters are listed in Table 15.

Table 15: Identified differential parameters

Parameter	**Value**
r [m]	0.125
l_c [m]	0.443
$g\,[\frac{\text{m}}{s^2}]$	9.81

Which yields the following state space representation of the mechanical system:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 22.14 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ -0.2822 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad \mathbf{D} = [0] \tag{203}$$

The bode diagram of this system is illustrated in Figure 101.

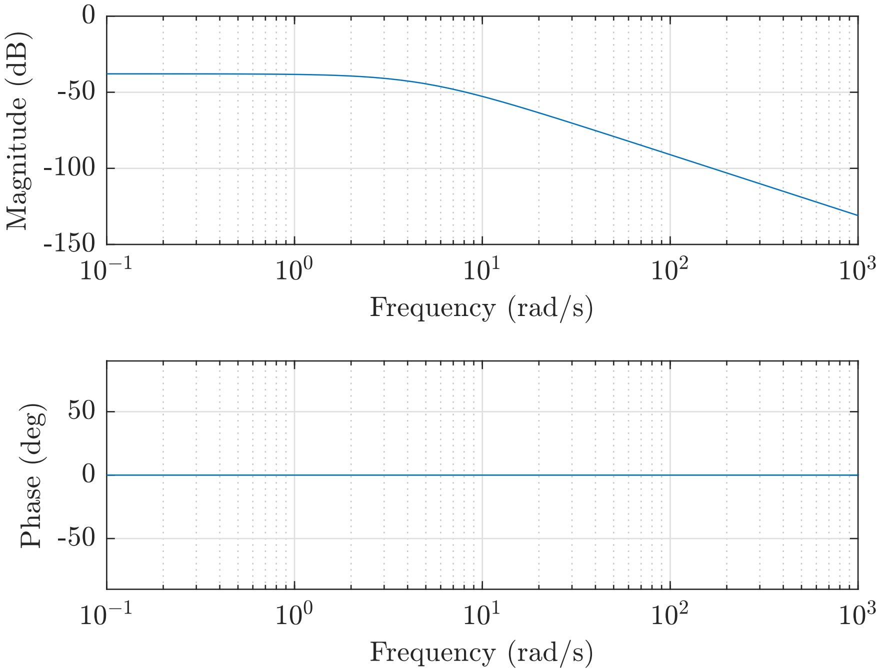


Figure 101: Bode diagram of mechanical differential system

Where the input and output of the system are the acceleration of the wheels and the pitch angle of the robot, respectively:

$$u = \ddot{\theta}_w \quad , \quad y = \theta_p \tag{204}$$

Hub Wheels System

The hub wheel system describes the dynamics between the input signal, u, and the acceleration of the hub wheels. This system is identified by performing a step response estimation of the wheels and reading out the resulting wheel velocity from encoder data through LabVIEW according to the theory from Chapter 3.5.4. A step response estimation is the selected approach due to the complexity of performing a frequency response of the robot with all its components while manually balancing it on two wheels. Figure 102 shows the results from the step response.

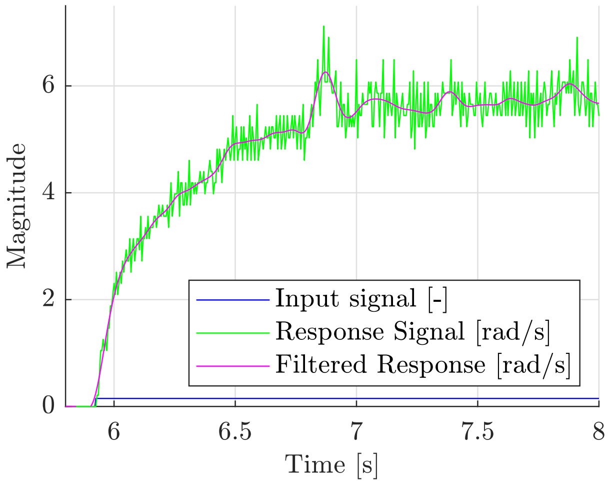


Figure 102: Step Response

Where the system parameters are identified to be:

Table 16: Step parameters

T_0	[s]	5.925
T_{63}	[s]	6.07
U_0	[-]	0
Y_0	[rad/s]	0
Y_{63}	[rad/s]	3.591
U_f	[-]	0.15
Y_f	[rad/s]	5.7

Which yields:

$$\frac{\dot{\theta}_w}{u} = \frac{\ddot{\theta}_w}{\dot{u}} = G_w(s) = \frac{38}{0.145\,s+1} \tag{205}$$

Where:

u = Input signal between -1 and 1
$\dot{u}$ = Change in input signal
$\dot{\theta}_w$ = Hub wheels velocity
$\ddot{\theta}_w$ = Hub wheels acceleration

Where u and $\dot{u}$ are the input signal and the change in input signal, and $\dot{\theta}_w$ and $\ddot{\theta}_w$ are the velocity and acceleration of the hub wheels, respectively. Applying the MATLAB function *tf2ss* to the transfer function converts it to state space representation;

$$\mathbf{A} = -6.897 \quad \mathbf{B} = 1 \quad \mathbf{C} = 262.1 \quad \mathbf{D} = 0 \tag{206}$$

The Bode diagram of the hub wheels' system is illustrated in Figure 103, and the open loop step response of $G_w(s)$ with the same input as done in the step response experiment from Figure 102 is illustrated with different values of T_{63} in Figure 104.

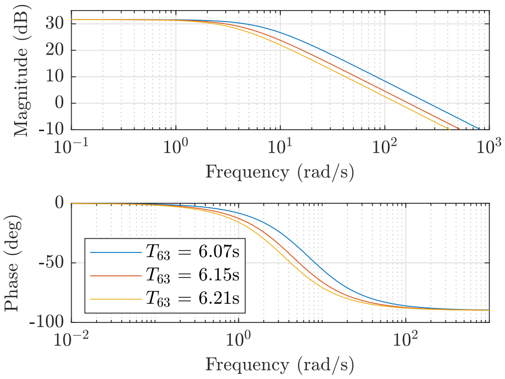


Figure 103: Hub Wheel system Bode diagram

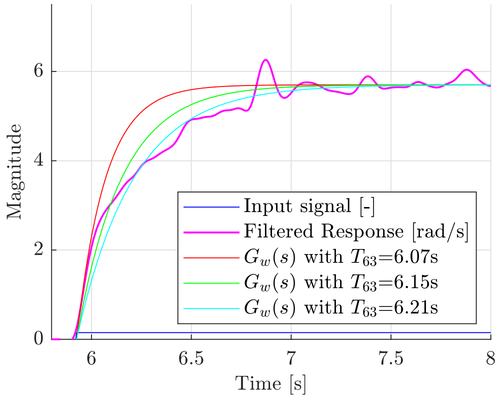


Figure 104: Open loop step response

It can also be seen from Figure 104 that a value of $T_{63} = 6.21$ seconds would fit better with the overall response signal, but due to the initial response leap and experimental verification it is determined that the best fit is with $T_{63} = 6.07$ seconds.

Combined Model

Similarly to the bicycle system, the hub wheel system and the mechanical system are combined using the MATLAB function *series* according to Figure 105, where $G_w(s)$ is the hub wheel system and $G_m(s)$ is the mechanical system.

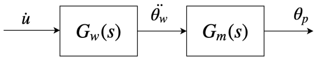


Figure 105: Full plant

The resulting system is:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 22.14 & 0 & -73.95 \\ 0 & 0 & -6.897 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 0 \end{bmatrix} \tag{207}$$

With the state vector:

$$\mathbf{x} = \begin{bmatrix} \theta_p \\ \dot{\theta}_p \\ x_3 \end{bmatrix} \tag{208}$$

Where:

θ_p = Robot pitch
$\dot{\theta}_p$ = Robot pitch velocity
x_3 = Hub wheel acceleration state

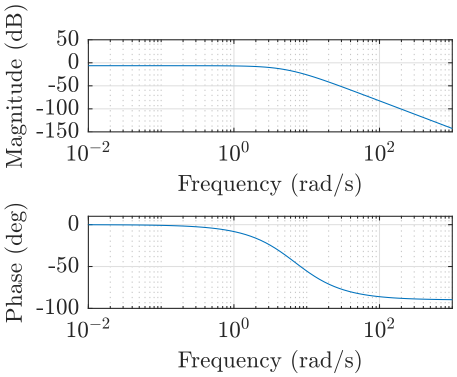


Figure 106: Full differential system

Similarly to the bicycle system, the last state, x_3, corresponds to a value that is proportional to the acceleration of the hub wheels by a factor equal to the hub wheels' system C-parameter. This value is measurable, but due to accumulated noise by double differentiation of the encoder data, observer data is preferred. The Bode diagram of the full system is illustrated in Figure 106.

4.4.3 Discretization

To make sure that the models found for continuous systems are correct when applied on a discrete real-time controller, a discretization of the relevant models is carried out. An exact discretization for the continuous state-space models [A,B,C,D] is done as seen in the Equations 209 with a sample time of Δt. [43]

$$A_d = e^{A\cdot\Delta t} \quad , \quad B_d = A^{-1}\cdot(A_d - I)\cdot B \quad , \quad C_d = C \quad , \quad D_d = D \tag{209}$$

The discrete state-space models are compared to the continuous systems to find the largest acceptable sampling time for the real-time controller where the discrete systems are approximately equal to the continuous state-space models. This comparison is seen in Figures 107 and 108 with three different sample times of 100ms, 20ms and 5ms for the bicycle model and differential drive model respectively.

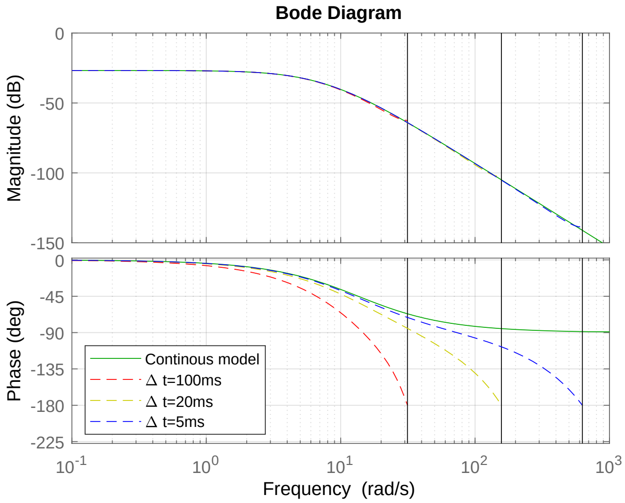


Figure 107: Comparing the bicycle mode's continuous model with its exact discrete model

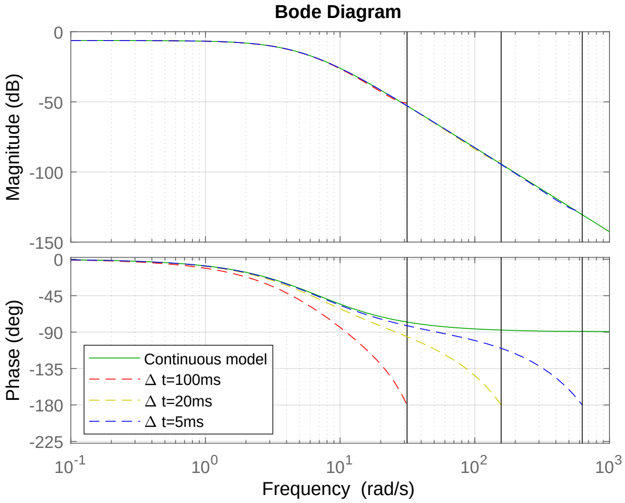


Figure 108: Comparing the differential mode's continuous model with its exact discrete model

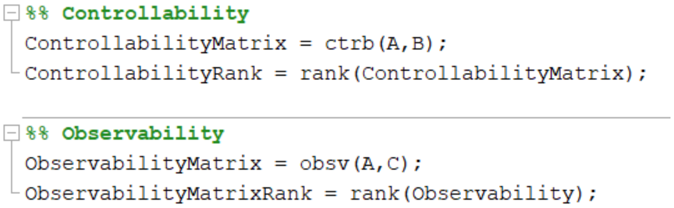
To be able to use the continuous transfer function up to the fastest working frequencies of the systems, which is assumed to be approximately 10rad/s, the sampling time must be less than 20 milliseconds as seen from the bode diagrams. Therefore, a sampling time of five milliseconds is chosen to achieve some safety margin.

4.5 Control

This chapter revolves around how the control theory from Chapter 3.6 is implemented on the robot in accordance with the dynamic systems presented in Chapter 4.4. The majority of control tuning is done in MATLAB, with the master code, *Main.vi*, attached in Appendix C.

4.5.1 Controllability and Observability

The controllability and observability of the combined systems are tested by utilizing simple functions in MATLAB according to the dynamic systems, as illustrated in Figure 109.



```
%% Controllability
ControllabilityMatrix = ctrb(A,B);
ControllabilityRank = rank(ControllabilityMatrix);

%% Observability
ObservabilityMatrix = obsv(A,C);
ObservabilityMatrixRank = rank(Observability);
```

Where:

A = A-matrix of state space system
B = B-matrix of state space system
C = C-matrix of state space system

Figure 109: MATLAB functions

The resulting ranks of the controllability matrices and the observability matrices of the two systems are listed in Table 17.

Table 17: Resulting ranks of the systems

	Differential System	Bicycle System
Rank of observability matrix	3	3
Rank of controllability matrix	3	3

This means that all states of both systems are fully controllable and observable considering that the ranks correspond to the order of the systems. Also, the controllability and observability of the combined systems are tested when an integral state is added according to Chapter 3.6.4. For this, the state space matrices are re-defined according to Equation 210 [2].

$$\mathbf{A}_i = \begin{bmatrix} 0 & \mathbf{C} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \quad \mathbf{B}_i = \begin{bmatrix} 0 \\ \mathbf{B} \end{bmatrix} \quad \mathbf{C}_i = \mathbf{C} \quad \mathbf{D}_i = \mathbf{D} \tag{210}$$

And further tested as in Figure 109 which yields the ranks presented in Table 18.

Table 18: Resulting ranks of the systems with integral state

	Differential System	Bicycle System
Rank of observability matrix	4	4
Rank of controllability matrix	4	4

Where the ranks again correspond to the increased order of the systems; four, including the integral state.

Hence, the systems are fully controllable and observable, both with and without integral control.

4.5.2 State Observer

The Luenberger state observer is implemented in LabVIEW by utilizing a ”Control & Simulation Loop”. Such a loop uses numeric ordinary differential equation (ODE) solvers to simulate the given systems at a given step size that corresponds to the iteration time of the main program. Based on Chapter 4.4.3, this step time is set to five milliseconds in order to assume continuous systems. Figure 110 illustrates how the observers are implemented in LabVIEW.

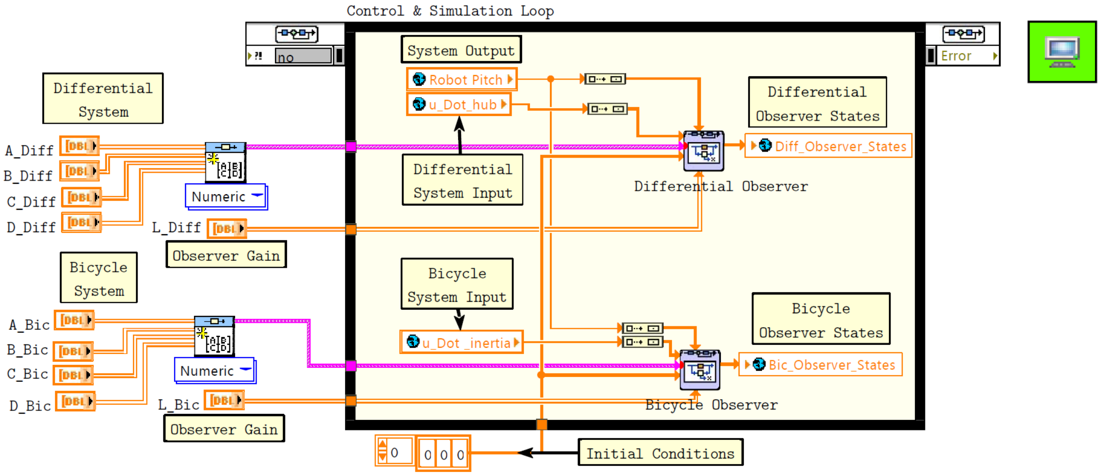
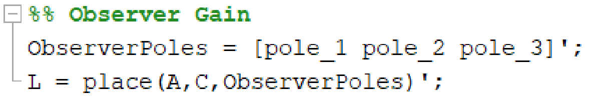


Figure 110: Control & Simulation Loop

Where observer gains are calculated through pole placement of the observer poles in MATLAB according to the code illustrated in Figure 111 [2].

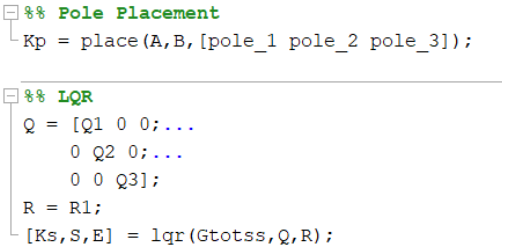


```
%% Observer Gain
ObserverPoles = [pole_1 pole_2 pole_3]';
L = place(A,C,ObserverPoles)';
```

Figure 111: Placing observer poles

4.5.3 State Feedback

The state feedback gain is determined through Bass-Gura pole placement and LQR in MATLAB, as illustrated in Figure 112.



```
%% Pole Placement
Kp = place(A,B,[pole_1 pole_2 pole_3]);

%% LQR
Q = [Q1 0 0;...
     0 Q2 0;...
     0 0 Q3];
R = R1;
[Ks,S,E] = lqr(Gtotss,Q,R);
```

Figure 112: Placing system poles

Where:

Kp = Feedback gain from Bass-Gura pole placement
Ks = Feedback gain from LQR
Q = State cost matrix for LQR
R = Input cost value for LQR

The state feedback can then be calculated in LabVIEW, as seen in Figure 113.

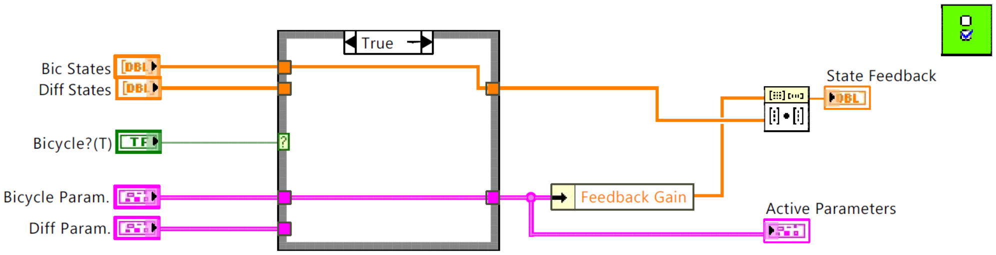
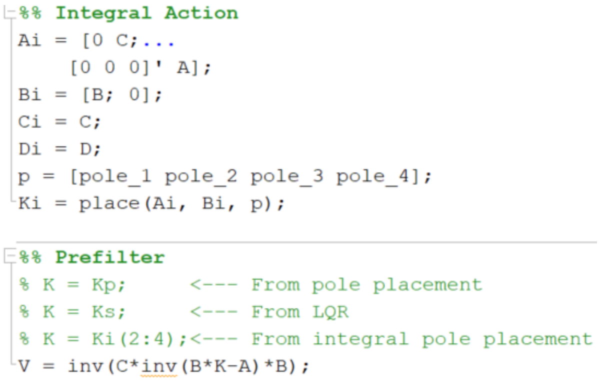


Figure 113: State feedback

Where a case structure determines which control system is active.

4.5.4 Prefilter and Integral Action

The prefilter and integral action controllers are analytically calculated through MATLAB as illustrated in Figure 114.



```
%% Integral Action
Ai = [0 C;...
    [0 0 0]' A];
Bi = [B; 0];
Ci = C;
Di = D;
p = [pole_1 pole_2 pole_3 pole_4];
Ki = place(Ai, Bi, p);

%% Prefilter
% K = Kp;       <--- From pole placement
% K = Ks;       <--- From LQR
% K = Ki(2:4);<--- From integral pole placement
V = inv(C*inv(B*K-A)*B);
```

Figure 114: Calculation of prefilter and integral

Where:

Ai = Redefined A-matrix
Bi = Redefined B-matrix
Ci = Redefined C-matrix
Di = Redefined D-matrix
Kp = Bass-Gura gain vector
Ks = LQR gain vector
Ki = Integral gain vector
V = Prefilter Gain

Which corresponds to the redefinition of the state space matrices with the inclusion of the integral state as presented in Chapter 4.5.1

Notice that the prefilter is dependent on which tuning approach is selected. The first value in the integral gain vector is the integral gain itself, while the three other values are the respective state feedback gains. Figure 115 illustrates how the prefilter and integral action are implemented in LabVIEW.

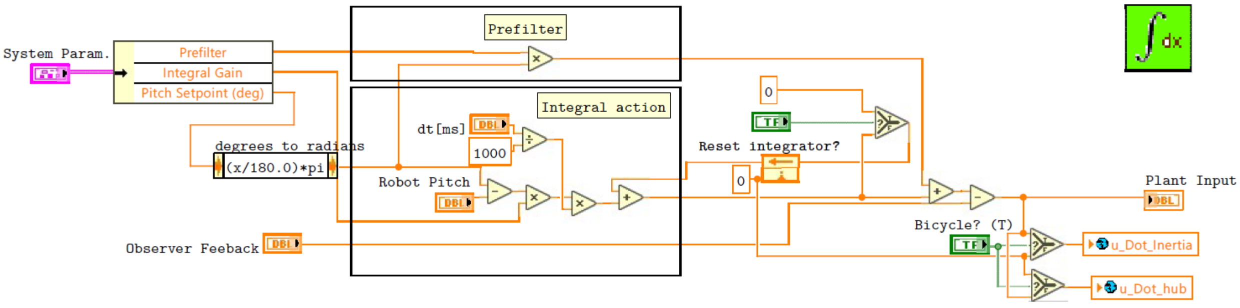


Figure 115: Prefilter and integral action VI

4.5.5 Variable Setpoint Control

The pitch setpoint of the bicycle system is offset based on the velocity of the inertia wheel. In order to avoid noisy data and unnecessarily many sensors which leads to an unnecessarily complex system, the transfer function obtained in Chapter 4.4.1 is simulated in the control and simulation loop, as shown in Figure 116, to estimate the inertia wheel velocity. It is seen from Figure 117 that the simulated velocity is relatively accurate especially at lower speeds.

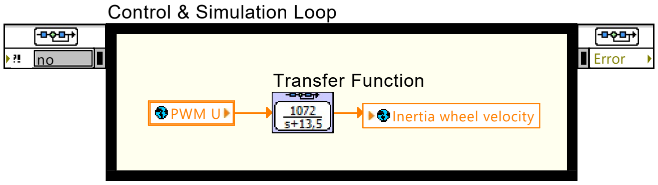


Figure 116: Transfer function implemented in the Control & Simulation Loop in LabVIEW

Where the input value, u, is used as input in the transfer function to estimate the inertia wheel velocity.

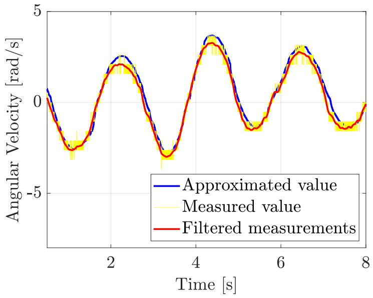


Figure 117: Measured and estimated angular velocity of the inertia wheel

For the differential drive, however, the variable setpoint control is implemented directly using the translational velocity of the robot as feedback. The LabVIEW codes in Figures 118 and 119 show how they are implemented.

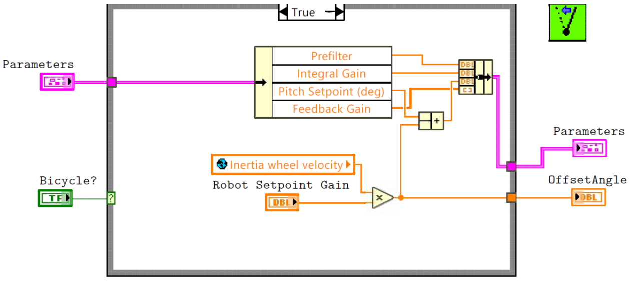


Figure 118: variable_pitch_offset.vi in bicycle mode

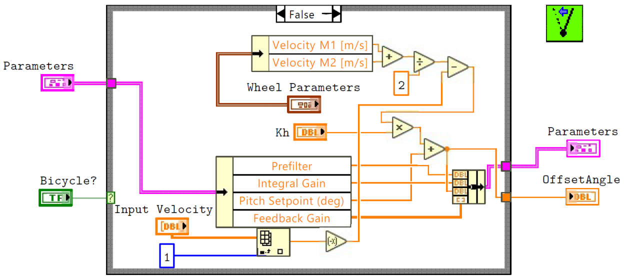


Figure 119: variable_pitch_offset.vi in differential drive mode

4.5.6 Deadband Compensation

The deadbands of the systems are identified by carefully ramping up the input signal from zero until the motors start to rotate. This value contains both the defined deadband of five percent inherited from the VESC tool and the static friction of the motors. In order to compensate for the deadband, the value is added and subtracted to the input signal as illustrated in Figure 120.

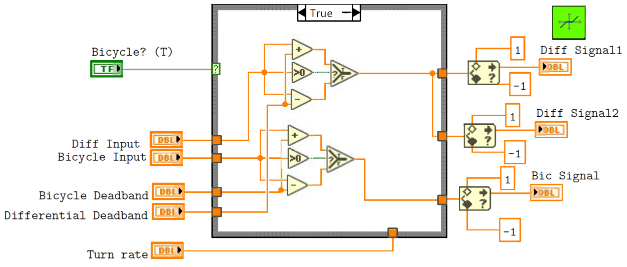


Figure 120: Deadband'.vi

4.6 Simulation

In order to confirm that the designed mathematical dynamic models are behaving in a realistic and correct manner, a simulation code in Simulink is designed. The simulation model is also used to test and confirm that the control designs are controlling the plants sufficiently. The simulation codes in Simulink are designed as seen in Figures 121 and 122 with the proper time step of five milliseconds. Such simulations are especially advantageous for testing unstable dynamics before applying it on the physical system which could get damaged if control parameters or dynamic models are incorrect.

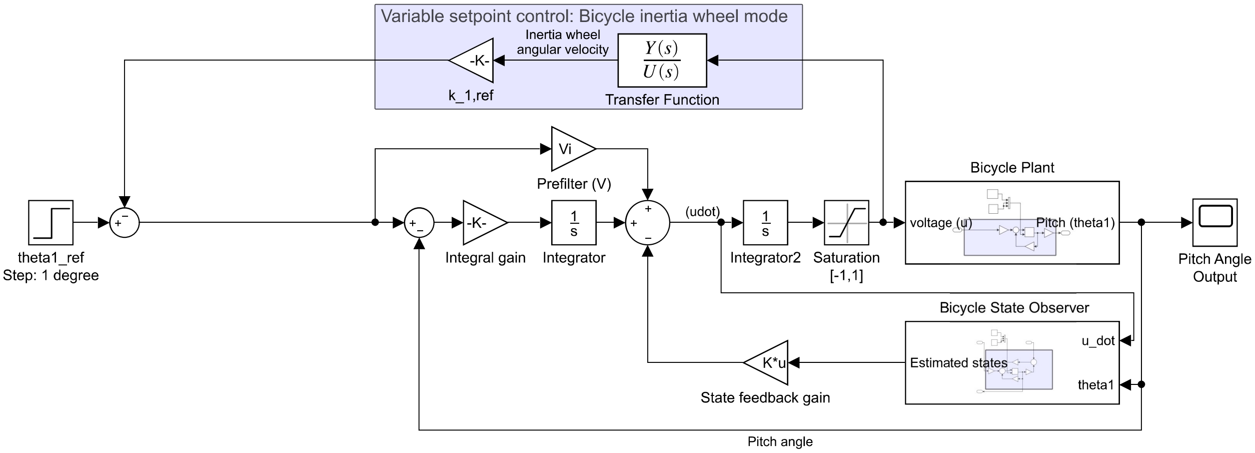


Figure 121: Simulink simulation for controlling the bicycle configuration with the inertia wheel

As seen in Figure 122, the hub wheel velocity is not measurable in the simulation and is therefore estimated through its equivalent transfer function as done with the inertia wheel.

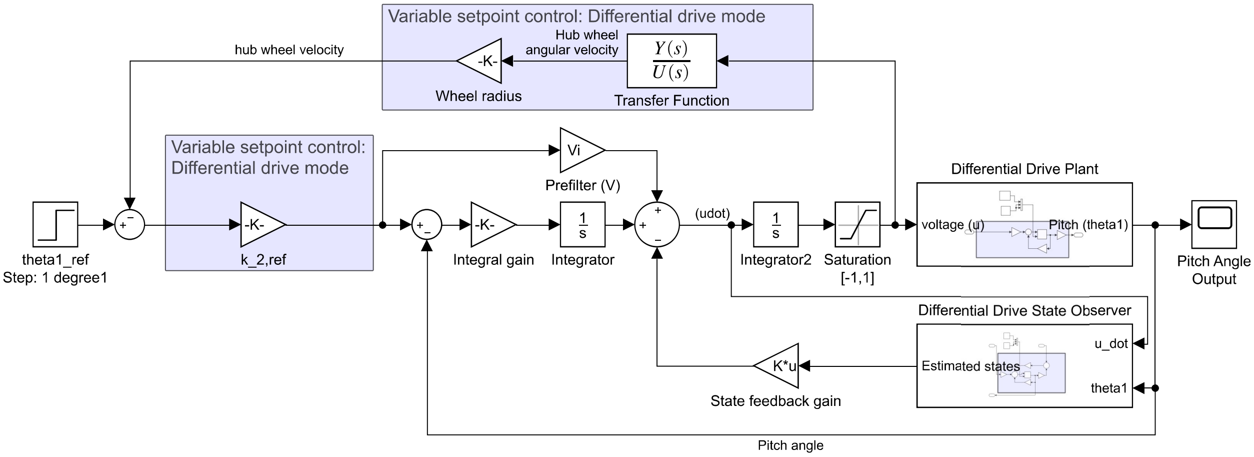


Figure 122: Simulink simulation for differential drive configuration

4.7 Transition Algorithm

The transition algorithm ensures the robot to maintain its balance through the transformation between the bicycle and differential drive configurations.

The flowchart in Figure 123 illustrates how the transformation algorithm is built where:

v = Translational velocity of the robot
x = Velocity threshold
T = Timer value
t_0 = Transformation time

And the green text indicating that a function is initiated, and red text indicating that a function is turned off.

The program starts in bicycle configuration with the bicycle balancing control active. If a transition is requested, it is checked if the average velocity is below a set threshold of 0.005 meters per second. If the robot meets this requirement, a timer of 750 milliseconds is set while the servo motors transition the wheels from bicycle to differential drive configuration. The purpose of this is for the robot to keep its balance with the inertia wheel still active through the transition process. The timer is subsequently then reset, bicycle balancing control is turned off, and differential drive balancing is initiated. Then, if a transition is once again requested, the velocity will be checked to be below the threshold, and the robot will turn off the differential drive balancing control and re-initiate the bicycle balancing control before the servos transition the wheels to bicycle configuration. This cycle is repeated until the program is shut down.

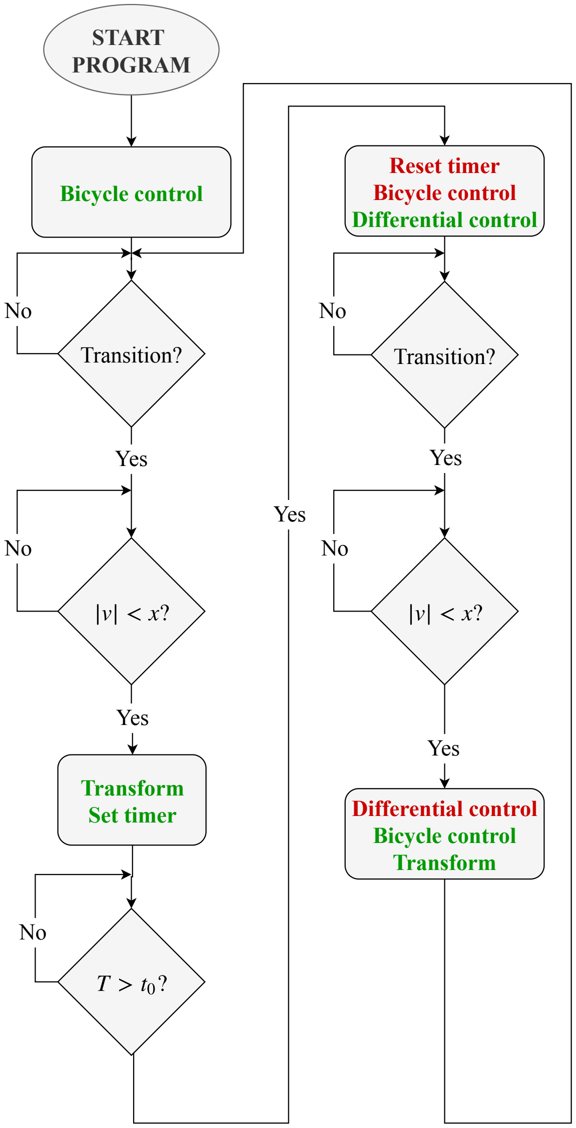


Figure 123: Transition algorithm

Note that while the bicycle control is active, also the dual wheel kinematics and odometry are active. Similarly, the differential kinematics and odometry are active while the differential drive control is active.

4.8 Locomotion

When the robot balances satisfactorily in both bicycle and differential drive configurations, the robot can be controlled and localized by utilizing the kinematics and odometry from Chapter 3.4.

4.8.1 Kinematic Control

The kinematics are controlled using an Xbox 360 controller as illustrated in Figure 124.

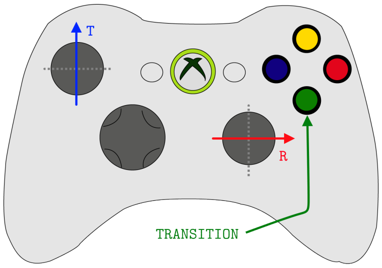


Figure 124: Xbox Controller [6]

Where the vertical direction on the left stick, T, controls the translational velocity of the robot in both configurations, and the horizontal direction on the right stick, R, controls the angular velocity of the robot in differential drive mode and the steering angle in bicycle mode. The green button switches the robot between the two different modes.

Since the myRIO does not support joystick inputs, the program that reads the Xbox controller data is run separately on a PC and sent to the myRIO over WiFi using shared variables as illustrated in Figure 125.

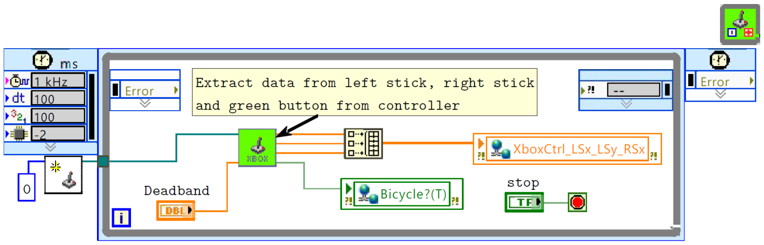


Figure 125: Joystick.vi

Where the analog input signals from the sticks are scaled between -1 and 1, and the transition button is toggled until it is pressed a second time. A deadband is also added to the analog signals to avoid undesired signals due to locking of the sticks.

For the bicycle configuration, the translational signal is scaled and added to the hub wheels to obtain a certain velocity, and the steering angle is also scaled and added to the servo motor setpoints to turn them desirably. On the other hand, for the differential drive mode, the variable setpoint control of the pitch angle enables the user to input a desired velocity, and the robot will offset its pitch to compensate for this velocity. The rotational signal of the differential drive mode is scaled and added to the right wheel and subtracted to the left wheel according to the inverse kinematics described in Chapter 51 to control the robot's rotational velocity.

4.8.2 Forward Kinematics

The robot's forward kinematics are calculated from the peripheral velocity of the two individual wheels in both configurations. The LabVIEW implementation of the forward kinematics is illustrated in Figure 126.

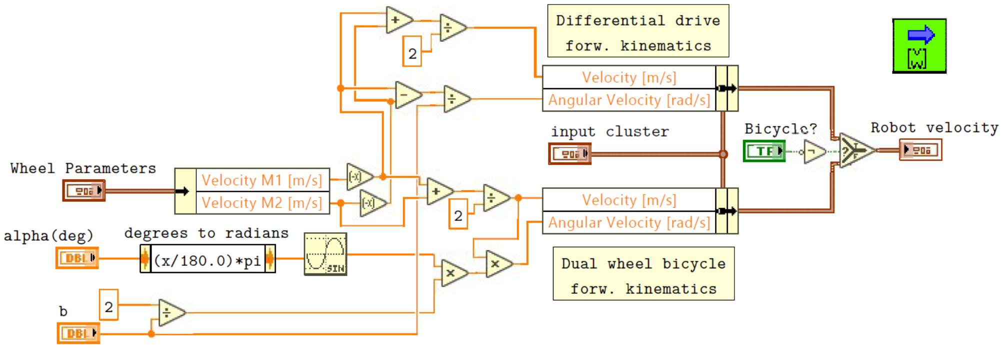


Figure 126: ForwardKinematics.vi

Where the forward kinematics are calculated individually for both differential drive and bicycle configurations, and consequently a boolean operator switches between the modes depending on which control structure is active. The forward kinematics are further used in the transition algorithm and the differential drive mode's variable setpoint control.

4.8.3 Odometry

Figure 127 shows a simplified illustration of how the odometry of the system is implemented on the robot.

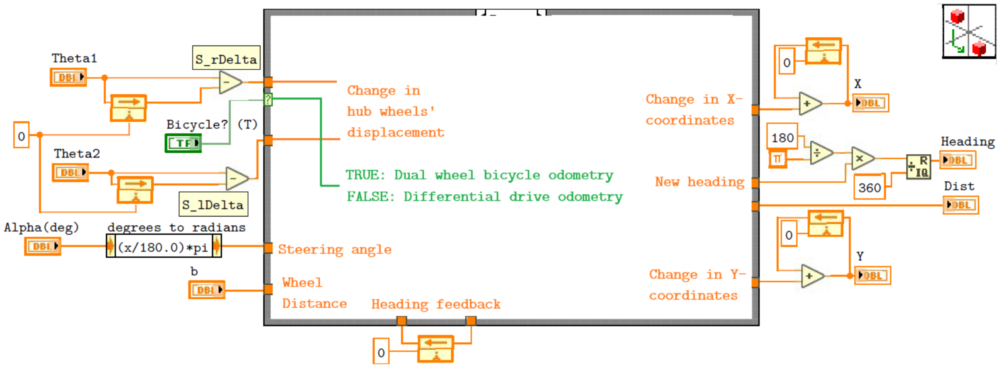


Figure 127: Odom.vi

Where the two different odometric models of dual wheel bicycle steering and differential drive steering from Chapters 3.4.1 and 3.4.3 respectively are directly translated into LabVIEW.

5 Results

This chapter will go through the results obtained from the constructed robot.

5.1 Mechanical Structure

The resulting mechanical structure of the robot in both bicycle and differential drive configurations are illustrated in Figures 128 and 129 respectively.

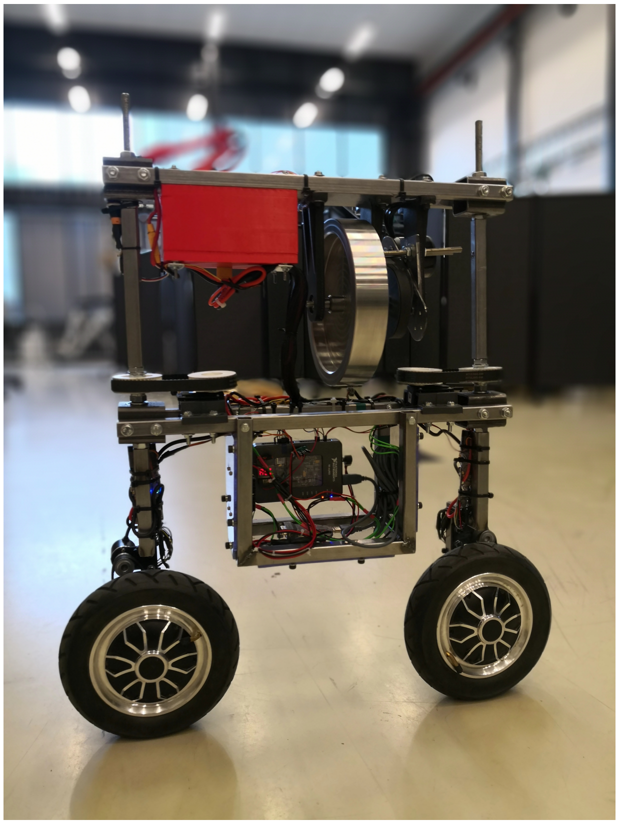


Figure 128: Bicycle

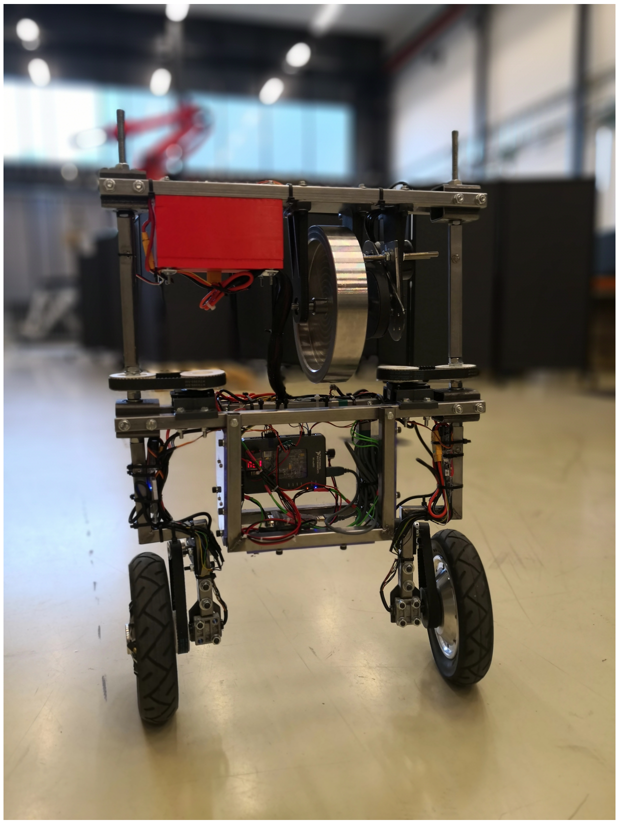


Figure 129: Differential drive

Minor changes were done from the mechanical CAD design in order to improve minor issues that arose during the assembly process. For example, the 3D printed inertia wheel brackets illustrated in Figure 130 are improved by using threaded rods between the brackets as well as a reinforcement plate between the bearing brackets and the base frame to increase rigidity.

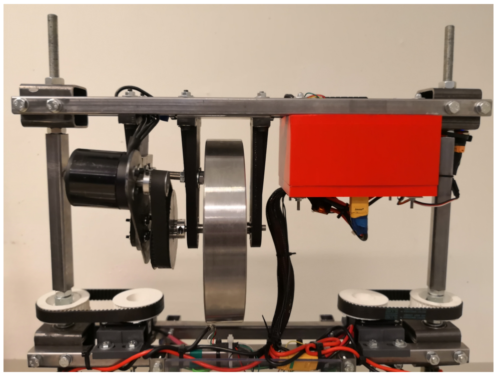


Figure 130: Top part of robot

5.2 Final Control Structure

The final control structure and its relevant control parameters are presented in this chapter. The control structure and gains are further used to obtain the results for self-balancing.

5.2.1 Block Diagram

The final control structure used to stabilize the service robot is illustrated as a block diagram in Figure 131

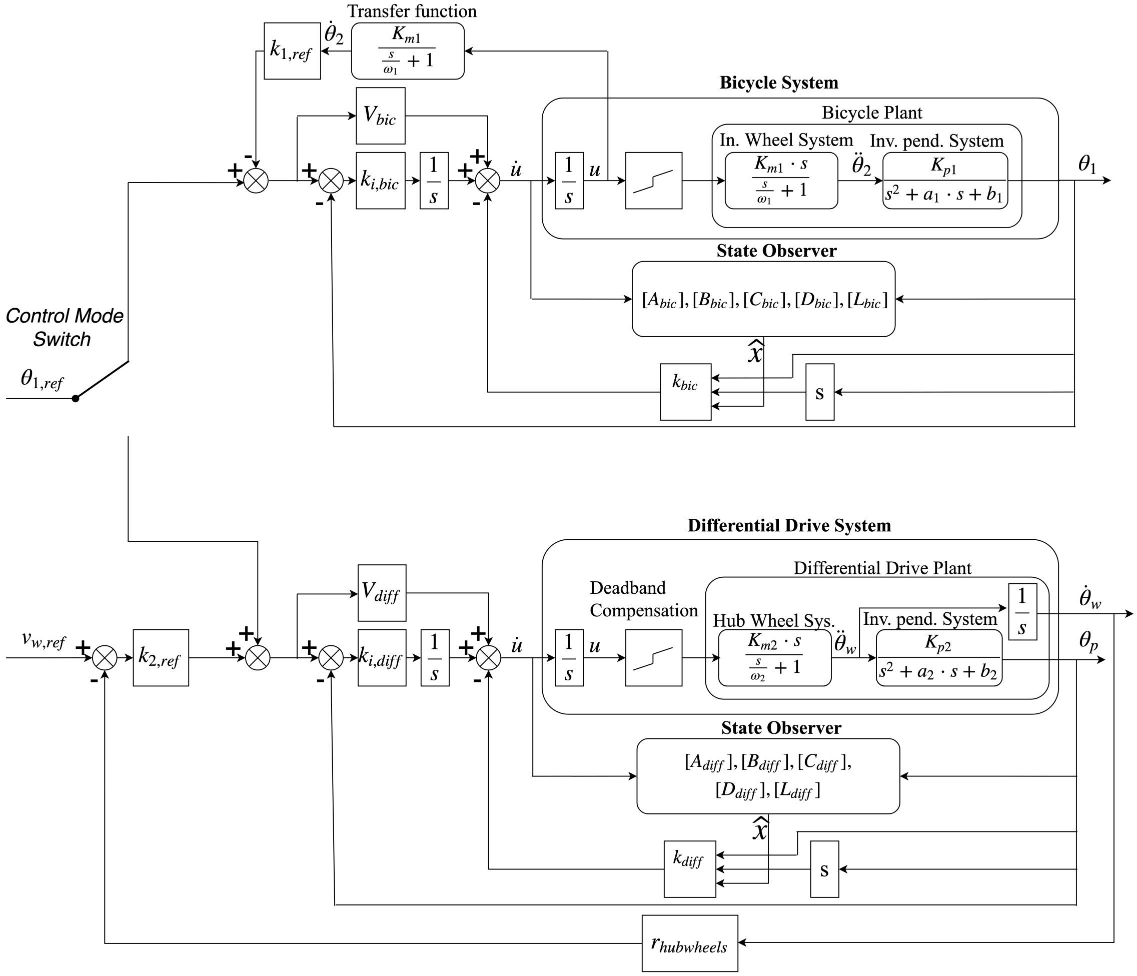


Figure 131: Overview of both the final bicycle and differential drive control system design

5.2.2 Control System Parameters

The tuning parameters that gave the best results in terms of stable balancing are listed in Table 19. All values in this table are obtained analytically.

Table 19: System poles and gains

		Bicycle System	Differential System
Luenberger Observer	Poles	$\begin{bmatrix}-45 & -40 & -20\end{bmatrix}$	$\begin{bmatrix}-25 & -22 & -20.5\end{bmatrix}$
	Feedback Gains	$\begin{bmatrix}91.5 & 2300 & -247\end{bmatrix}$	$\begin{bmatrix}60.1 & 1098 & -48.5\end{bmatrix}$
Bass-Gura Pole Placement	Poles	$\begin{bmatrix}-9 & -8 & -4\end{bmatrix}$	$\begin{bmatrix}-10 & -11 & -12\end{bmatrix}$
	Feedback Gains	$\begin{bmatrix}-47.1 & -8 & 7.5\end{bmatrix}$	$\begin{bmatrix}-27.7 & -5.19 & 26.1\end{bmatrix}$
	Prefilter	-13.1	-17.85
LQR	Q	$\begin{bmatrix}0.5 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0.1\end{bmatrix}$	$\begin{bmatrix}0.5 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0.1\end{bmatrix}$
	R	0.05	0.05
	Poles	$\begin{bmatrix}-3.34 & -11.57 \pm 3.45i\end{bmatrix}$	$\begin{bmatrix}-13.8 \pm 12.0\,i & -0.84\end{bmatrix}$
	Feedback Gains	$\begin{bmatrix}-65 & -11.8 & 13\end{bmatrix}$	$\begin{bmatrix}-12.3 & -5.14 & 21.5\end{bmatrix}$
	Prefilter	-22.2	-3.8
Pole Placement with Integral Action	Poles	$\begin{bmatrix}-2 & -9 & -8 & -4\end{bmatrix}$	$\begin{bmatrix}-6 & -7 & -7.5 & -6.5\end{bmatrix}$
	Feedback Gains	$\begin{bmatrix}-63 & -9.9 & 9.5\end{bmatrix}$	$\begin{bmatrix}-24.6 & -3.99 & 20.1\end{bmatrix}$
	Integral Gain	-26.2	-27.7
	Prefilter	-25.86	-16.52

The variable setpoint gains and deadbands implemented in the system are listed in Table 20.

Table 20: Variable setpoint gains

	Bicycle System		Differential System	
Setpoint Gain	$k_{1,ref}$	0.02	$k_{2,ref}$	4.0
Deadband	±0.057		±0.062	

Note that all dynamic systems are derived with the appropriate deadbands.

5.3 Self-Balancing

This chapter presents all results regarding the self-balancing of the service platform.

5.3.1 Complementary Filter

Figure 132 illustrates the effects of the complementary filters with different values of α.

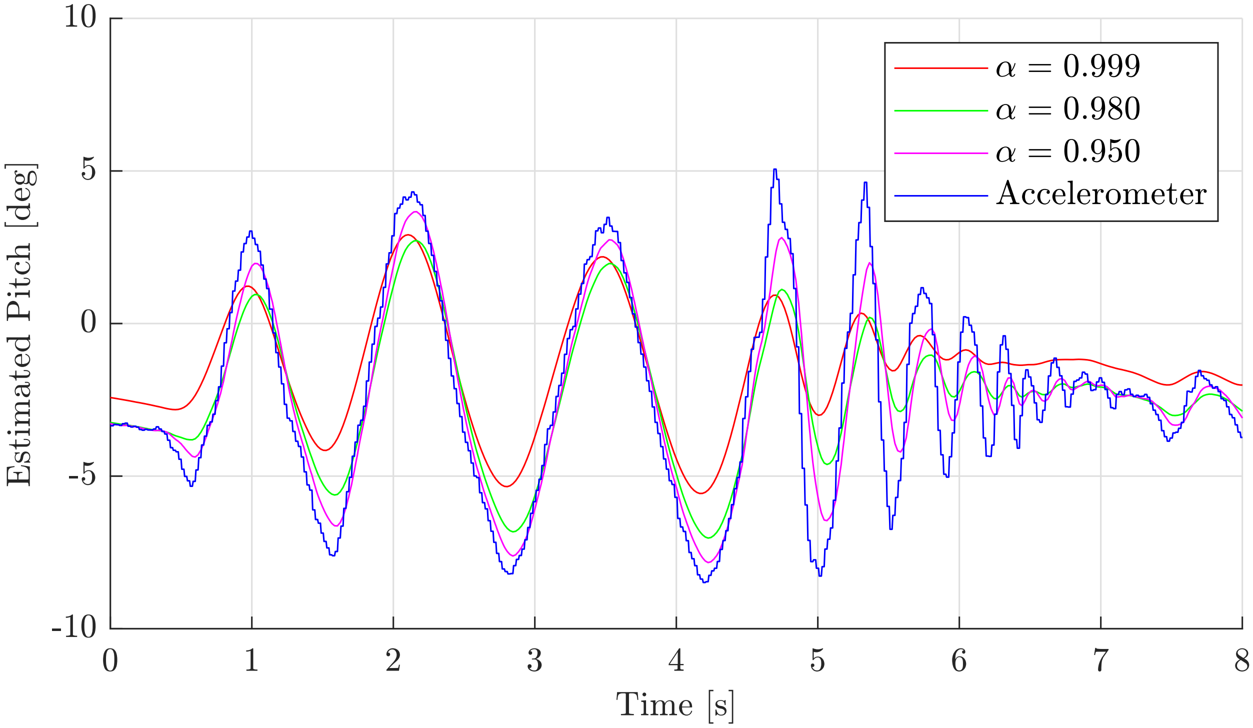


Figure 132: Complementary filter results

Where it can be seen that as α approaches one, the phase lag of the complementary filter will decrease, and more high frequency noise from the accelerometer is filtered out. Therefore, the value of $\alpha = 0.999$ is used to estimate the pitch displacement.

5.3.2 Bicycle Balancing

In order to find the best control approach of LQR, Bass-Gura pole placement, and Bass-Gura pole placement with integral action, the pitch angle data is plotted on top of each other for easy comparison. Figures 133 and 134 depict results from the different approaches where the variable setpoint control is excluded.

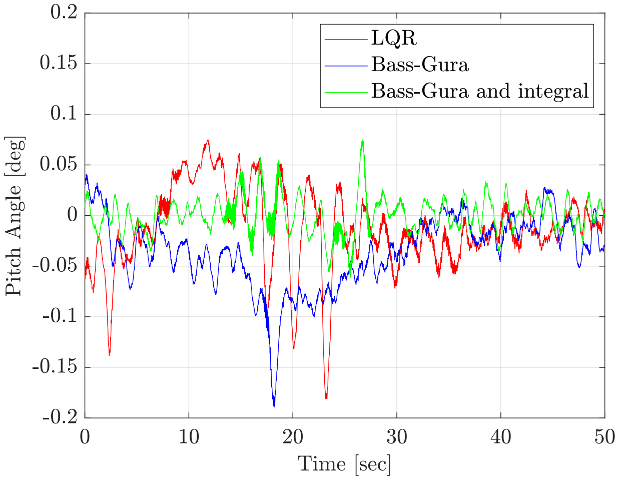


Figure 133: Comparing different control approaches

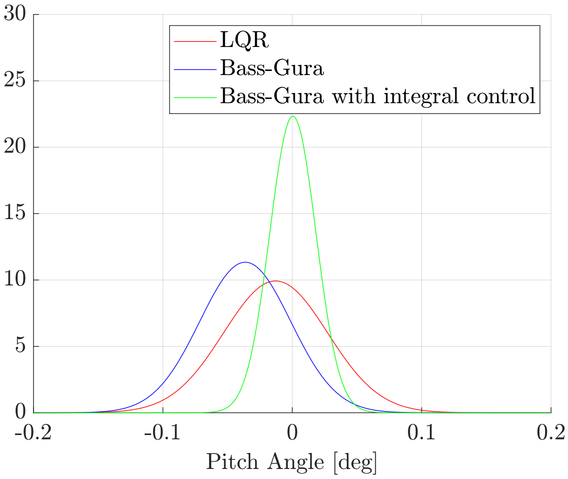


Figure 134: Normal probability density functions for all three approaches

It is seen from Figure 134 that the pole placement with integral action gives the most accurate and stable pitch. Using this approach and including the variable setpoint controls, the pitch angle, variable setpoint and inertia wheel velocity are plotted over one another for easy comparison in Figure 135.

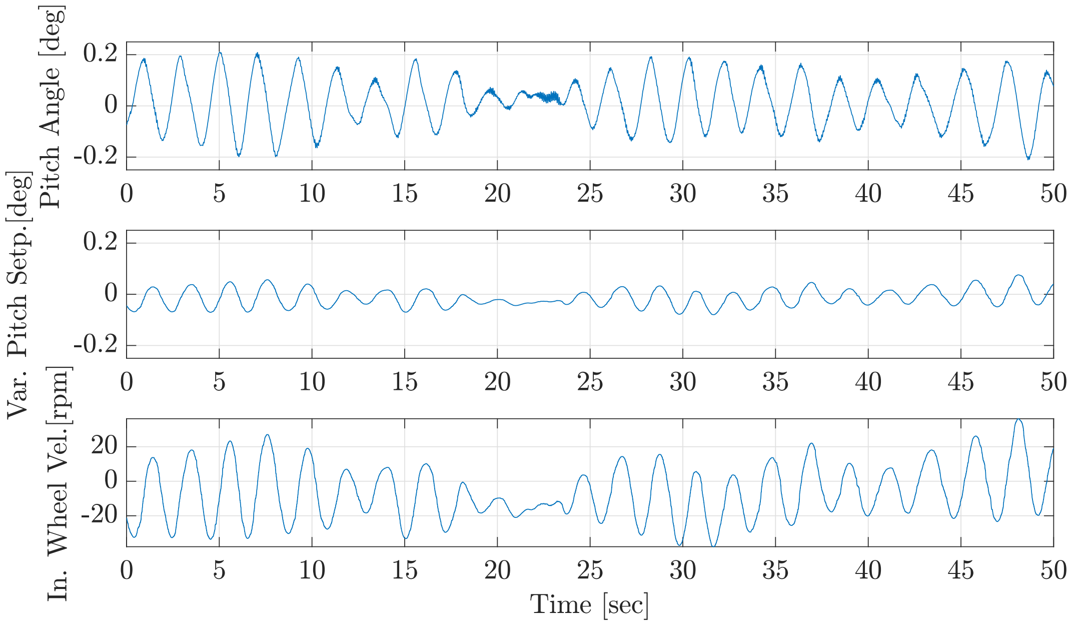


Figure 135: Pole placement with integral action

To test the best control approach, the system is subjected to a step of one degree as an input to $\theta_{1,ref}$ for ten seconds.

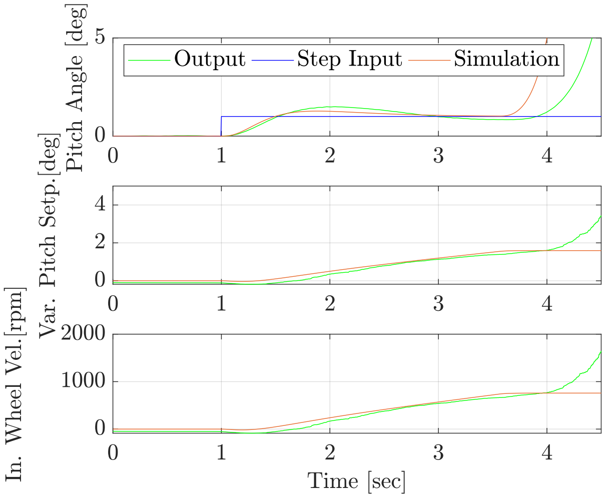


Figure 136: Step input without setpoint control

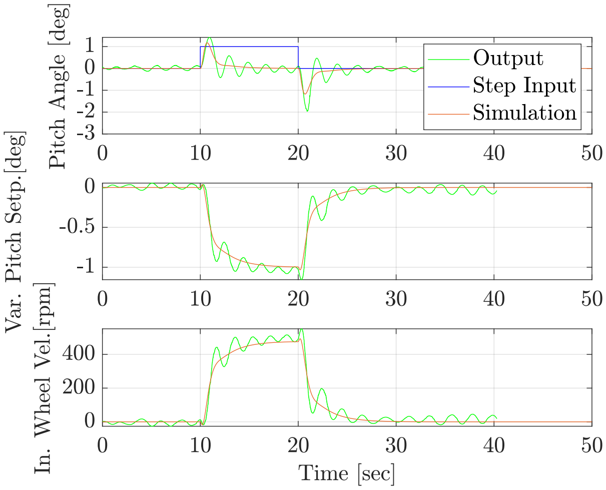


Figure 137: Double step input with setpoint control

As seen in Figure 136, the robot will be able to balance at one degree, but the motor speed will eventually saturate and fall over. Notice also that the inertia wheel velocity does not saturate because it is estimated through a transfer function. The step input is also applied when the variable setpoint control is active as seen in Figure 137, and the pitch angle is therefore controlled back to zero degrees, which is its ideal balancing point. The step input was also simulated as discussed in Chapter 4.6, and it is confirmed that the assumption of a highly realistic plant model and desirable control design is correct since the simulated behavior is roughly the same as the real behavior. All gains and other variables are equal for both the simulation and the physical system with values as shown previously in Tables 19 and 20.

5.3.3 Differential Drive Balancing

The resulting pitch angles from utilizing the different control approaches excluding the variable pitch control are illustrated in Figure 138, and the consequential wheel displacements during balancing are illustrated in Figure 139.

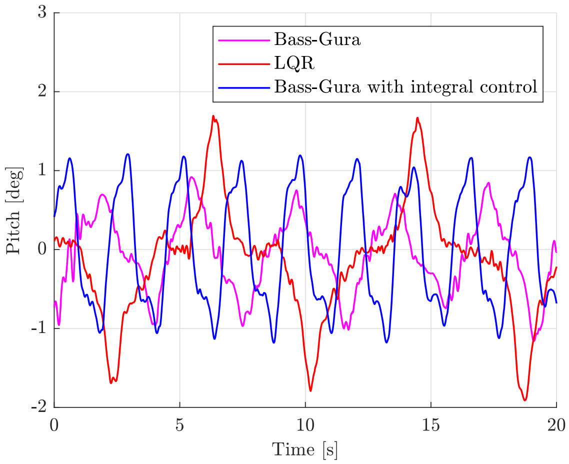


Figure 138: Pitch comparison

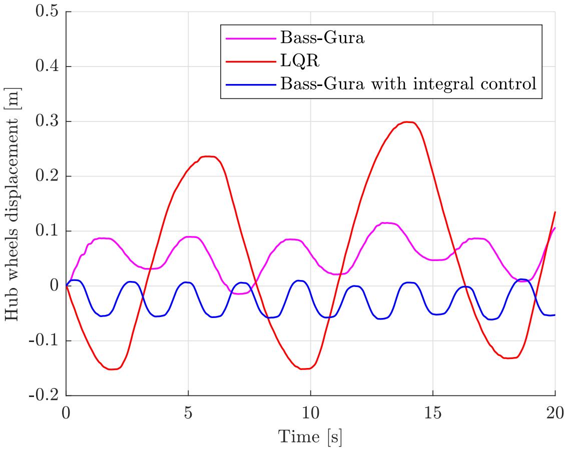


Figure 139: Wheel displacement comparison

These results are compared up against each other using normal distributions, as illustrated in Figures 140 and 141.

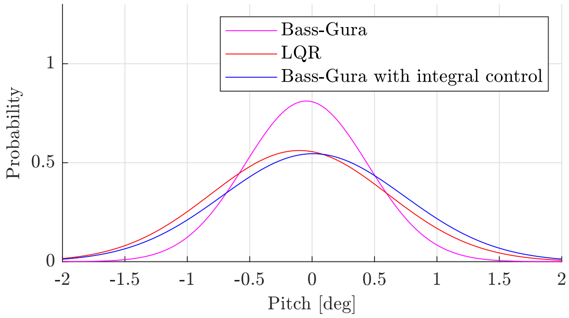


Figure 140: Pitch comparison

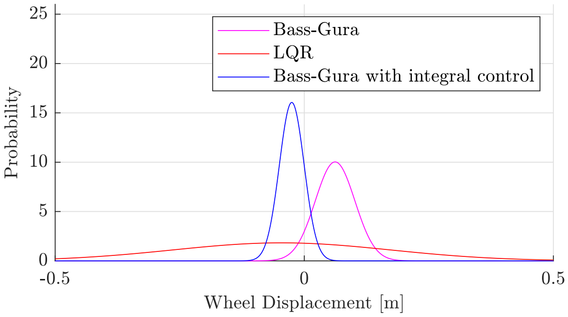


Figure 141: Wheel displacement comparison

The Bass-Gura pole placement approach yields a well-balanced robot with little displacement drift. It was however observed that small oscillations occur due to the poles of the system being placed relatively far left on the complex plane. Also the LQR controller gave good results in terms of balancing with no signs of instability or oscillations, but with high wheel displacement due to the relatively high weighing of pitch velocity in the Q-matrix. However, the best results came with the Bass-Gura pole placement approach with the inclusion of integral action with no high-frequency oscillation and little displacement drift.

Figure 142 illustrates the balancing results using the Bass-Gura with integral action approach when also including the variable pitch control while standing still, and Figure 143 shows how the robot acts after being subjected to a reference velocity step with the variable setpoint control active.

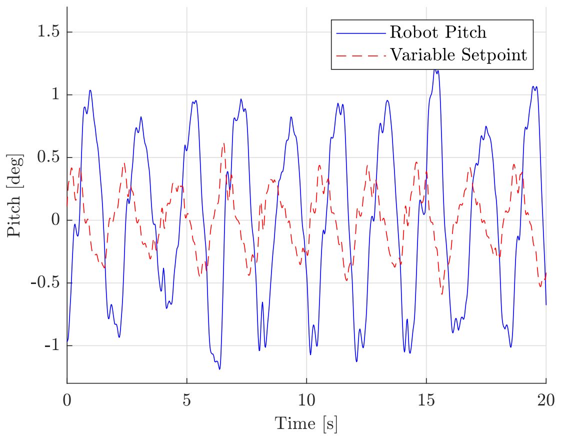


Figure 142: Balancing Results

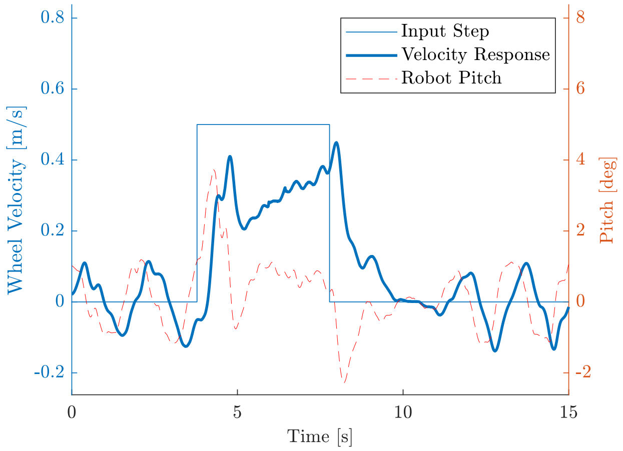


Figure 143: Velocity Response

The velocity response case from Figure 143 is also simulated in Simulink with all the same gains to verify the dynamic system. Simulation results are shown in Figure 144. Note that the simulation deals with an ideal system with no noise, which is why the oscillations in the simulation are not as severe as on the physical system.

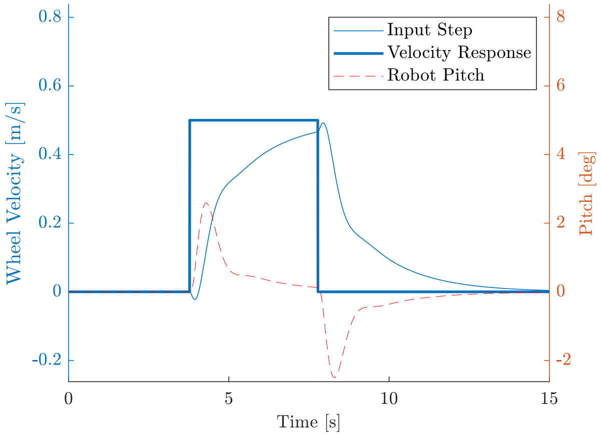


Figure 144: Simulation Response

5.3.4 Transition Balancing

Figure 145 shows the resulting pitch of the robot while transitioning between bicycle and differential configuration. It is again specified that the robot utilizes the inertia wheel balancing during the transition period.

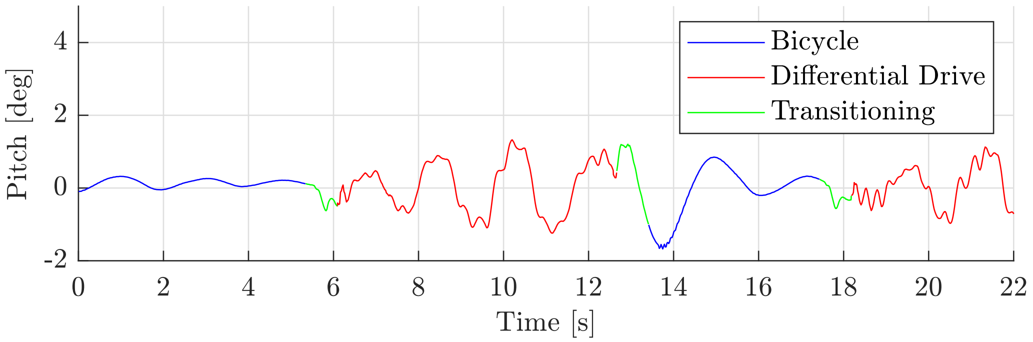


Figure 145: Pitch through transition

5.3.5 Observer Data

The state estimations from the Luenberger state observer is compared to the two measurable states for accuracy analysis. This is done for both the inertia wheel balancing and differential drive balancing as shown in Figure 147 and 148 respectively. The legend illustrated in Figure 146 applies for all figures in this chapter.

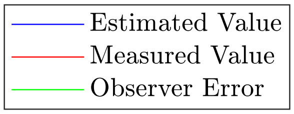


Figure 146: Legend

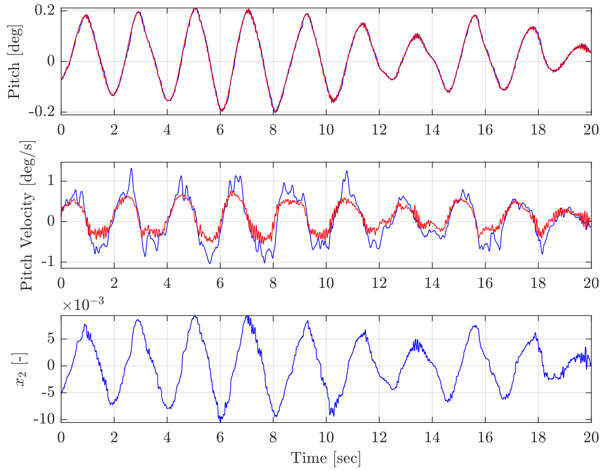


Figure 147: Bicycle balancing data

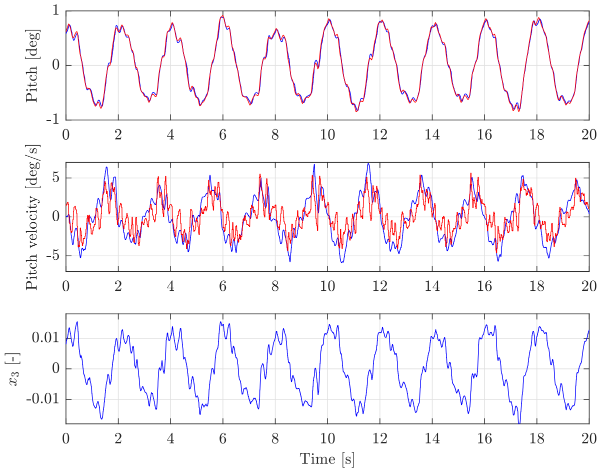


Figure 148: Differential drive balancing data

The observer error, which is the difference between the estimated and measured values as shown previously in Figures 149 and 150, gives an indication on how accurate the third observer states of the two systems, x_2 and x_3, are, which are the only estimated states applied in the control.

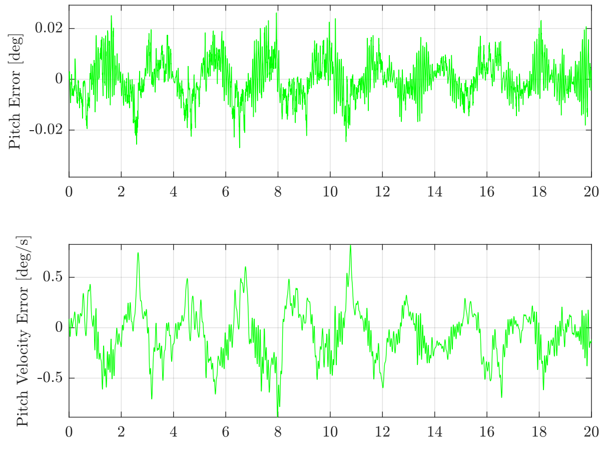


Figure 149: Bicycle state error

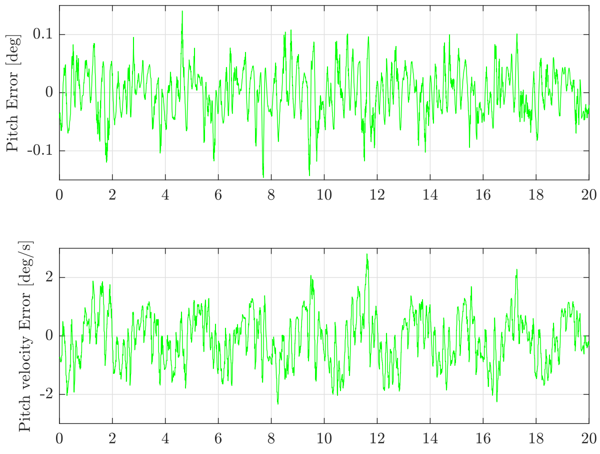


Figure 150: Differential drive state error

5.4 Odometry

Even though the robot localization is not an integral part of the thesis, the results regarding the robot localization through dead reckoning are included. Figures 151 and 152 illustrate the resulting odometry from running the robot through a certain course. Notice that some variations between the two figures because the course is followed differently in the two configurations.

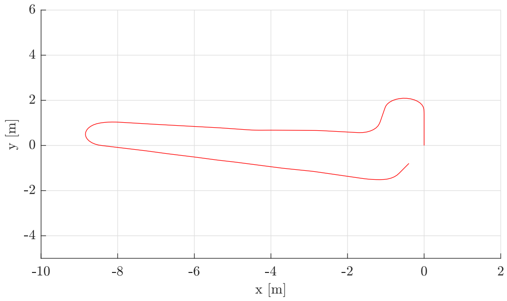


Figure 151: Bicycle odometry

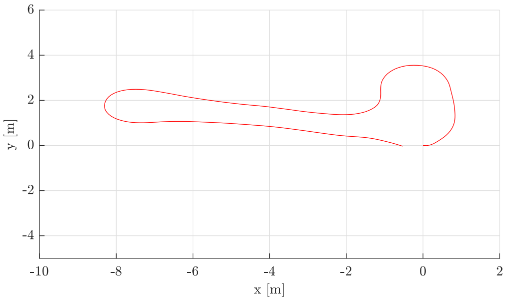


Figure 152: Differential drive odometry

The goal of the course is to finish in the same pose as the initial pose, which is not fully obtained as seen in the figures. Over time the pose estimation will drift and thereby yield false estimates. This is illustrated in Figures 153 and 154 by overlapping the same paths from Figures 151 and 152 multiple times to simulate multiple laps.

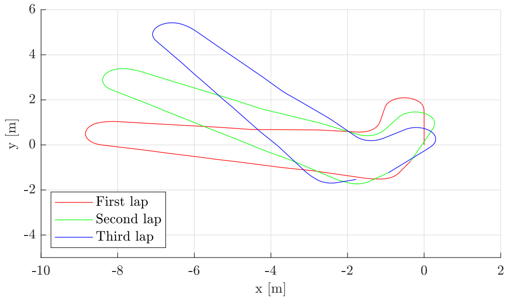


Figure 153: Bicycle odometry drift

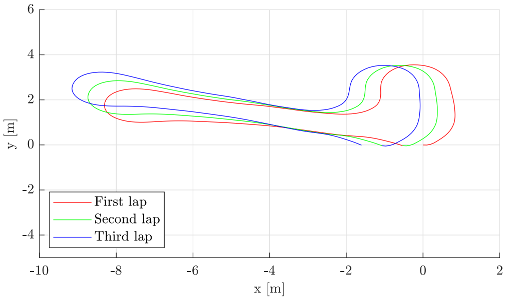


Figure 154: Differential drive odometry drift

5.5 Internal Acceptance Testing

As the final step in the development process, an internal acceptance test is executed in accordance with the V-model. The test is done through a cold reboot of the myRIO, and transferring the LabVIEW code to an independent laptop. From the independent laptop, the software is uploaded through Wi-Fi connection to the myRIO. The results of the internal acceptance test are listed below:

- The robot balances in bicycle mode
- The robot balances in differential drive mode
- The robot balances through the transition
- The robot is controllable through the joystick

In other words, all testing points are satisfied, and the internal acceptance test is considered successful.

6 Discussion

This chapter will discuss the results and potential improvements that could be implemented to improve the performance quality of the resulting service platform.

The results from Chapter 5 present a robot that is able to balance in both differential drive and bicycle configurations as well as in the transition between them. Especially the bicycle configuration with the inertia wheel balances very well with very small errors from the pitch reference. Also the differential drive mode balances well, but due to inaccurate system identification of the hub wheels system from the step response estimation, the maximum potential quality of the balancing is not achieved. This could be improved by performing a frequency response on the hub wheels as well, by for instance stabilizing the robot from a crane while performing the experiments. Generally, the results show that the best performance was obtained by the inclusion of an integral state to both control systems.

The possibilities of using a Kalman filter rather than a complementary filter to estimate the pitch angle of the robot could also be considered to potentially increase the accuracy of the pitch estimation. The main advantage of the Kalman filter is that the estimations start at the currently measured pitch opposed to the complementary filter where the estimations start at a provided initial condition and approach the current pitch of the robot over a short time span. However, the complementary filter is a sufficient and comfortable approach when the provided initial conditions correspond to the starting pitch angle.

During testing and result generation, it was observed that the loop time of the Dynamixel serial communication is slightly unstable. The main issue with loop delay is that the transition algorithm is timer based, which may culminate with a control mode being activated prematurely. This could be solved by further investigating the possibilities of extracting the angular position of the Dynamixel servo motors directly from the serial bus. Another solution would be to utilize external rotary sensors, such as encoders, to provide feedback on their angular positions. Nonetheless, this issue only occurs when the system has been running for a longer span of time due to memory overload and not initially.

The calculated odometry is not sufficient for localizing the robot and determining its pose over time. As a short term localization tool, the pose is accurately estimated, but as time and travel distance increase, the estimations will drift and become invalid. Nevertheless, the odometry described in this report lays down the foundation to be improved further with, for example, simultaneous localization and mapping (SLAM) in the future. Considering that localization is not an integral part of this assignment, the results are considered satisfactory.

Some mechanical features of the resulting service platform have the potential for improvements in terms of both functionality and stress distribution. Since the hub wheels only have output shafts on one side of the wheels, it results in single shear, and higher shear stresses will occur in the mechanical frame as a consequence of this. The design could be improved by redesigning the lower legs of the service platform to minimize the shear stress in the system. Also, the gear ratio in the inertia wheel system could be optimized using the theory of optimal gear ratio [44].

Another challenge with the robot is that the IMU is slightly temperature sensitive, which results in the calculated pitch being offset over time. Though the variable setpoint control contributes to maintaining the robot stable, it is highly recommended that the IMU is replaced in the future.

7 Conclusion and Further Work

A multi-mode, two-wheeled, self-balancing service platform that can transition between differential drive and bicycle configuration has been successfully designed and developed with satisfying results. The theory behind stabilizing the systems and the systems' kinematics are covered in depth, and the designed control structures were able to keep the pitch angle error satisfyingly low even when transitioning from one configuration to another. The kinematics makes the service robot useful in the sense that it can be actively driven in both bicycle and differential drive configurations with a wireless joystick controller and be roughly located with encoder feedback and dead reckoning.

All code except for the kinematic control through the wireless controller was implemented on the National Instruments *myRIO* and programmed in *LabVIEW*. Most of the analysis and simulation of the control system design was done in the software *MATLAB & Simulink*, while the mechanical construction design and analysis were done in *SOLIDWORKS*. The electronics revolve around tailoring a fitting power supply to the general system, but also around the choice of electric motors and sensors. Welding, soldering, CNC machining, and 3D printing among other constructional tools and machines were necessary to build this service robot prototype. Therefore, the group is satisfied with how this master's thesis has covered the different disciplines that mechatronics is defined by, as shown in Figure 155. It is also our belief that this interdisciplinary thesis can assist and be inspiring to push the limits of what a service robot has the potential to accomplish in the hopefully near future.

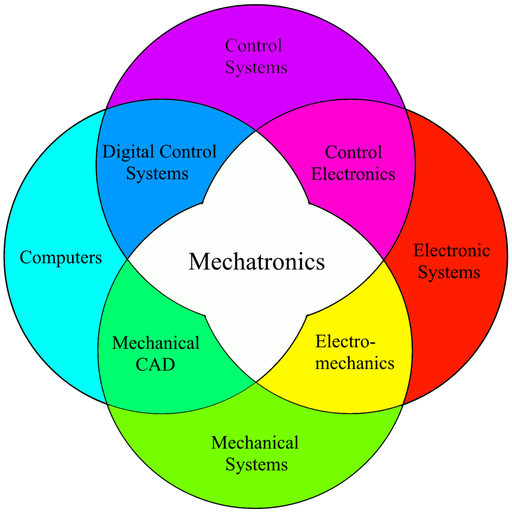


Figure 155: Kevin Craig's definition of mechatronics [7]

With regards to further work, the transition to the front wheel bicycle control for fast transportation should be easier to implement now that the service robot is stable during the slow inertia wheel bicycle mode. The front wheel bicycle mode could also be considered to be changed into a control where both wheels are utilized for better mobility. Also, the localization with pure encoder feedback is obviously not sufficient for such a service robot. Therefore a more robust method like SLAM should be implemented as well to account for drifting and uncertainties in the long run. There was also some issues with inaccurate pitch measurements during the first minutes when the IMU is powered up, and it should be replaced with a less temperature sensitive sensor in the future. A brief user's guide is included for further development in Appendix G.

References

[1] Sorawuth Vatanashevanopakorn and Manukid Parnichkun. Steering control based balancing of a bicycle robot, 2011. **Link to PDF**.

[2] Michael Ruderman. Lecture: State-space methods, 2018.

[3] Massachusetts institute of technology. Understanding poles and zeros. **Link to PDF**.

[4] AMESWEB. Preferred fits and tolerances charts (iso & ansi metric standards), 2010. **Link to page**.

[5] ESKATING. Electric skateboard speed controller (maytech), -. **Link to page**.

[6] Willpeters21. Xbox 360 controller clipart, -. **Link to image**.

[7] Alan S. Brown. Mechatronics and the role of engineers, 2011. **Link to page**.

[8] linmotors. Bldc - nema 17, -. **Link to page**.

[9] David Kiley. Ford tests delivery robots that could one day replace human workers, 2019. **Link to page**.

[10] Markforged. Onyx, -. **Link to page**.

[11] Markforged. Kevlar, -. **Link to page**.

[12] Mark Pedley. Tilt sensing using a three-axis accelerometer, 2013. **Link to PDF**.

[13] OlliW's Bastelseiten. Imu data fusing: Complementary, kalman, and mahony filter, 2013. **Link to page**.

[14] MIT. Discrete approximation of continuous-time systems, 2011. **Link to PDF**.

[15] Erik Cheever. Transformation: Transfer function to state space, -. **Link to page**.

[16] SIU Carbondale. Lesson 14: Transfer functions of dc motors, - produced. **Link to PDF**.

[17] The University of Newcastle. Lecture 10: Elements of system identification, -. **Link to PDF**.

[18] Finn Haugen. *Regulering av dynamiske systemer*. Tapir forlag, 1994.

[19] Abbas Emami-Naeini Gene F. Franklin, J. David Powell. *Feedback Control of Dynamic Systems (Sixth edition)*. PEARSON, 2010.

[20] Norman S. NISE. *Control Systems Engineering (Sixth edition)*. John Wiley & Sons, Inc., 2011.

[21] Perry Y.Li. State feedback and observer feedback. **Link to page**.

[22] Dr. Radhakant Padhi. Pole placement control design, 2013. **Link to page**.

[23] Steve Brunton. Control bootcamp: Linear quadratic regulator (lqr) control for the inverted pendulum on a cart, 2017. **Link to page**.

[24] Kjell G. Robbersmyr. *Maskinkonstruksjon Konstruksjonsanalyse*. Kjell G. Robbersmyr, 2008.

[25] AHR international. Bearing material properties chart, -. **Link to page**.

[26] Edvard Osakue. General tolerance dimensioning, -. **Link to PDF**.

[27] MatWeb. Steels, general properties, -. **Link to page**.

[28] PROF. R. VELMURUGAN. Composite materials, module iii - micro-mechanics of lamina, -. **Link to PDF**.

[29] MatWeb. Markforged onyx 3d printed polymer, -. **Link to page**.

[30] MatWeb. Overview of materials for acrylonitrile butadiene styrene (abs), molded, -. **Link to page**.

[31] Ian Poole. Lithium ion battery advantages & disadvantages, -. **Link to page**.

[32] LG. Product specification, 2014. **Link to PDF**.

[33] VESC Project. Electric skateboard speed controller (maytech), -. **Link to page**.

[34] ELPROCUS. Difference between pam, pwm and ppm, -. **Link to page**.

[35] Robotis. Dynamixel, 2019. **Link to page**.

[36] Trossen Robotics. Dynamixel mx-64t robot actuator, -. **Link to page**.

[37] ROBOTIS. Roboplus 1.0. **Link to PDF**.

[38] RJ Gross. Robotis dynamixel actuators in labview, Jan 13, 2011. **Link to PDF**.

[39] Robotis. Mx-64t / mx-64r / mx-64at / mx-64ar, 2017. **Link to page**.

[40] Arduino. A brief introduction to the serial peripheral interface, -. **Link to page**.

[41] IllinoisControls. Mpu-9250 myrio, 2014. **Link to page**.

[42] Katsuhiko Ogata. *Modern Control Engineering*. Pearson, 5th edition, 2010.

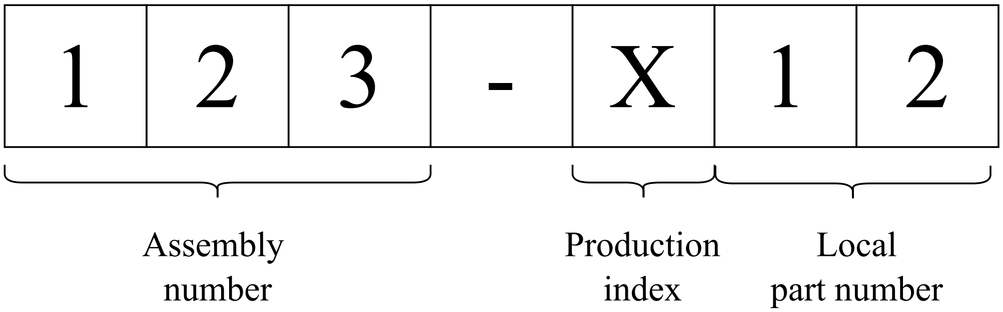
[43] Michael Ruderman. Lecture: Digital control, 2018.

[44] Keith Knight. Understanding inertia and reflected inertia, -. **Link to PDF**.

A Technical Drawings

The drawings in this appendix are sorted into one main assembly and several sub-assemblies. Each of them will have separate bills of material (BOM) and subchapters. The drawings are made in A3 paper scale which has been scaled down to fit the report's standard A4 setup.

The following yields for part numbers:



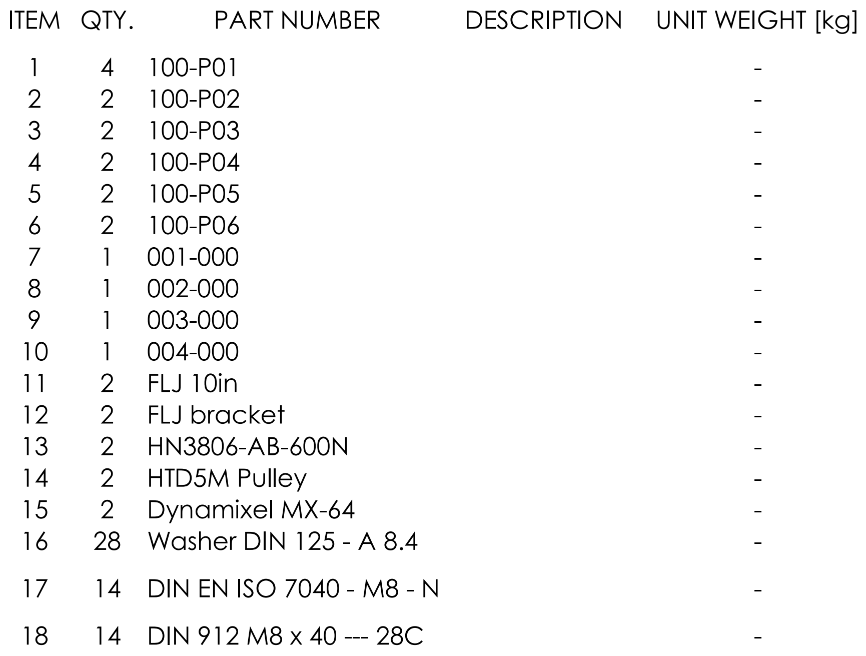
Where the assembly number tells which main assembly it is part of, production index tells how it is produced;

0 = Steel part, welded and machined
C = Laser cut part in acrylics
P = 3D printed part, mainly plastic

And the local part number tells which part it is in the given assembly.

A.1 100-000: Full System

Bill of materials for the full system:

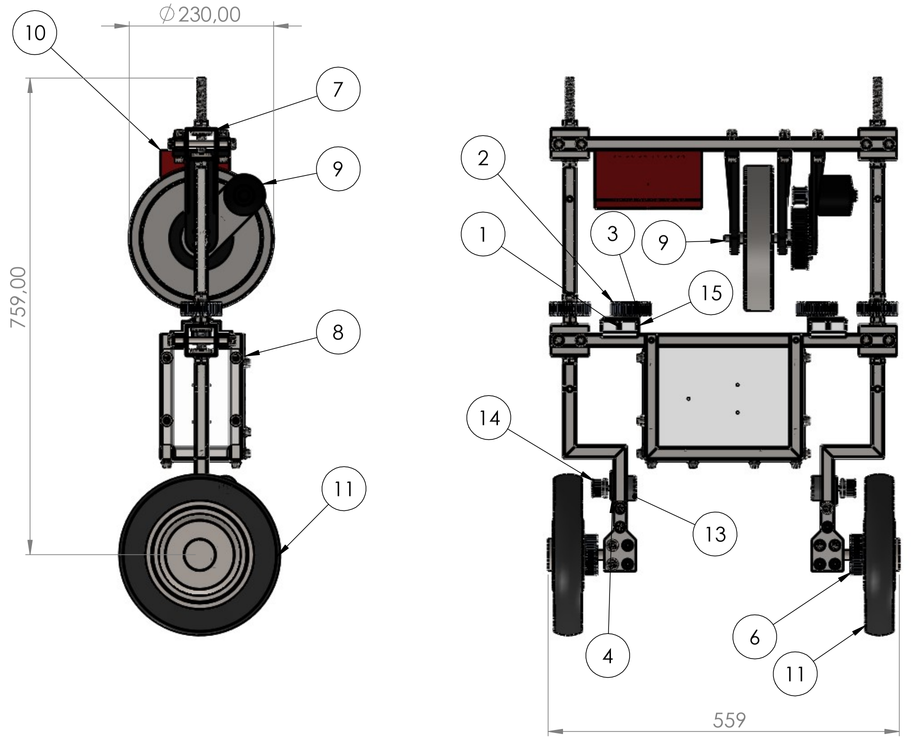


ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	4	100-P01		-
2	2	100-P02		-
3	2	100-P03		-
4	2	100-P04		-
5	2	100-P05		-
6	2	100-P06		-
7	1	001-000		-
8	1	002-000		-
9	1	003-000		-
10	1	004-000		-
11	2	FLJ 10in		-
12	2	FLJ bracket		-
13	2	HN3806-AB-600N		-
14	2	HTD5M Pulley		-
15	2	Dynamixel MX-64		-
16	28	Washer DIN 125 - A 8.4		-
17	14	DIN EN ISO 7040 - M8 - N		-
18	14	DIN 912 M8 x 40 --- 28C		-

Next page will include the assembly of the full system, with its respective parts.

This document and all information and data herein or herewith are confedencial and propriertary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF



ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	4	100-P01		
2	2	100-P02		
3	2	100-P03		
4	2	100-P04		
5	2	100-P05		
6	2	100-P06		
7	1	001-000		
8	1	002-000		
9	1	003-000		
10	1	004-000		
11	2	FLJ 10in		
12	2	FLJ bracket		
13	2	HN3806-AB-600N		
14	2	HTD5M Pulley		
15	2	Dynamixel MX-64		
16	28	Washer DIN 125 - A 8.4		
17	14	DIN EN ISO 7040 - M8 - N		
18	14	DIN 912 M8 x 40 --- 28C		

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

DWG NO. 100-000

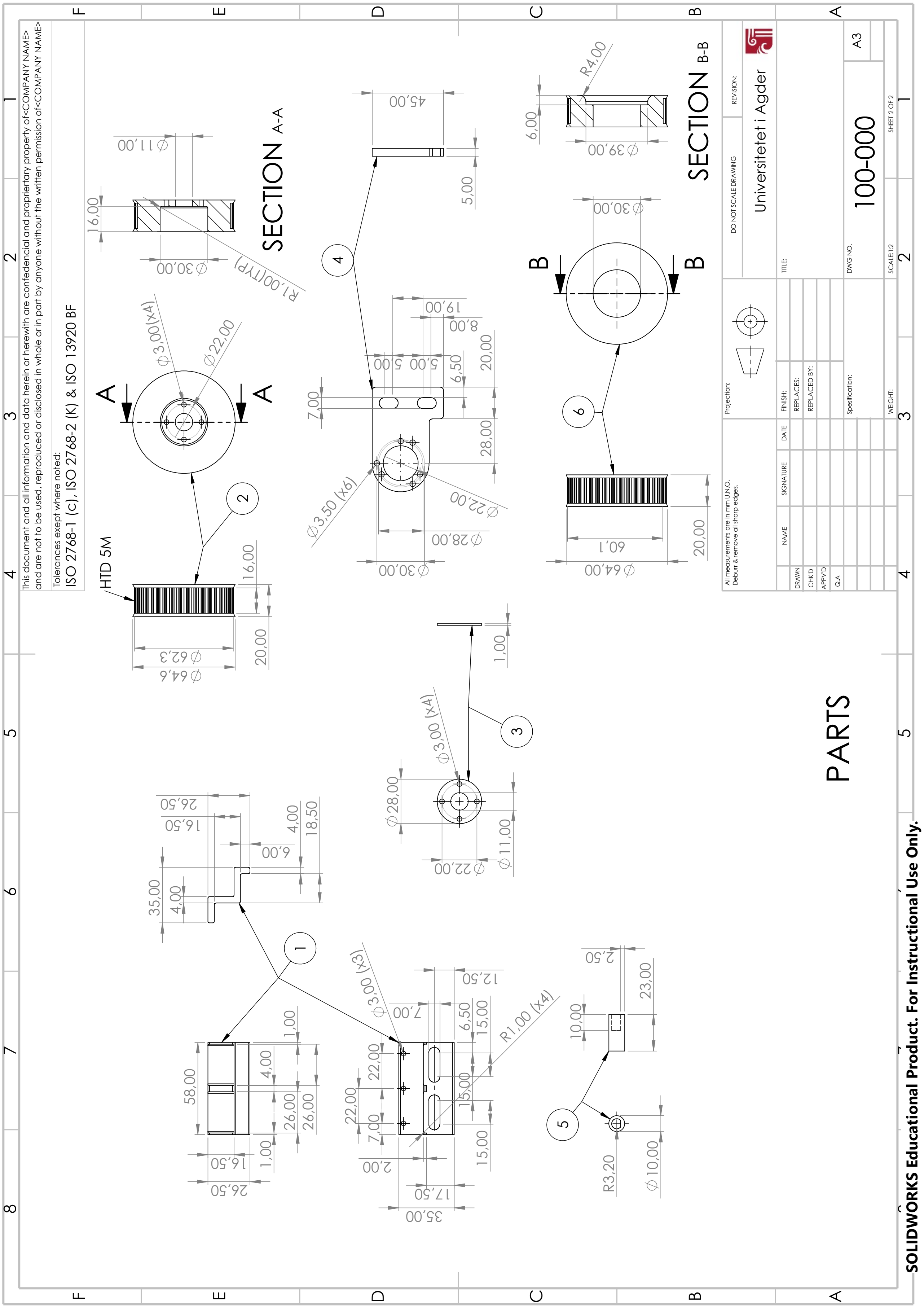
A3

WEIGHT:

SCALE:1:7

SHEET 1 OF 2

ASSEMBLY


This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME>
and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>
Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF
HTD 5M
SECTION A-A
SECTION B-B
A
A
B
B
1
2
3
4
5
6
PARTS
16,00
Ø11,00
Ø30,00
R1,00(TYP)
Ø3,00(x4)
Ø22,00
16,00
20,00
Ø62,3
Ø64,6
45,00
5,00
19,00
8,00
5,00
5,00
6,50
20,00
7,00
28,00
Ø3,50 (x6)
Ø22,00
Ø28,00
Ø30,00
R4,00
6,00
Ø39,00
Ø30,00
60,1
20,00
Ø64,00
1,00
Ø3,00 (x4)
Ø28,00
Ø22,00
Ø11,00
26,50
16,50
6,00
4,00
18,50
35,00
4,00
58,00
1,00
4,00
26,00
26,00
1,00
16,50
26,50
Ø3,00 (x3)
12,50
7,00
6,50
15,00
R1,00 (x4)
22,00
22,00
15,00
7,00
15,00
2,00
17,50
35,00
2,50
23,00
10,00
R3,20
Ø10,00
All measurements are in mm U.N.O.
Deburr & remove all sharp edges.
Projection:
DO NOT SCALE DRAWING
REVISION:
Universitetet i Agder
NAME
SIGNATURE
DATE
FINISH:
REPLACES:
REPLACED BY:
TITLE:
DRAWN
CHK'D
APPV'D
Q.A
Spesification:
DWG NO.
100-000
A3
WEIGHT:
SCALE:1:2
SHEET 2 OF 2
SOLIDWORKS Educational Product. For Instructional Use Only.
F
E
D
C
B
A
1
2
3
4
5
6
7
8

A.2 001-000: Base Frame

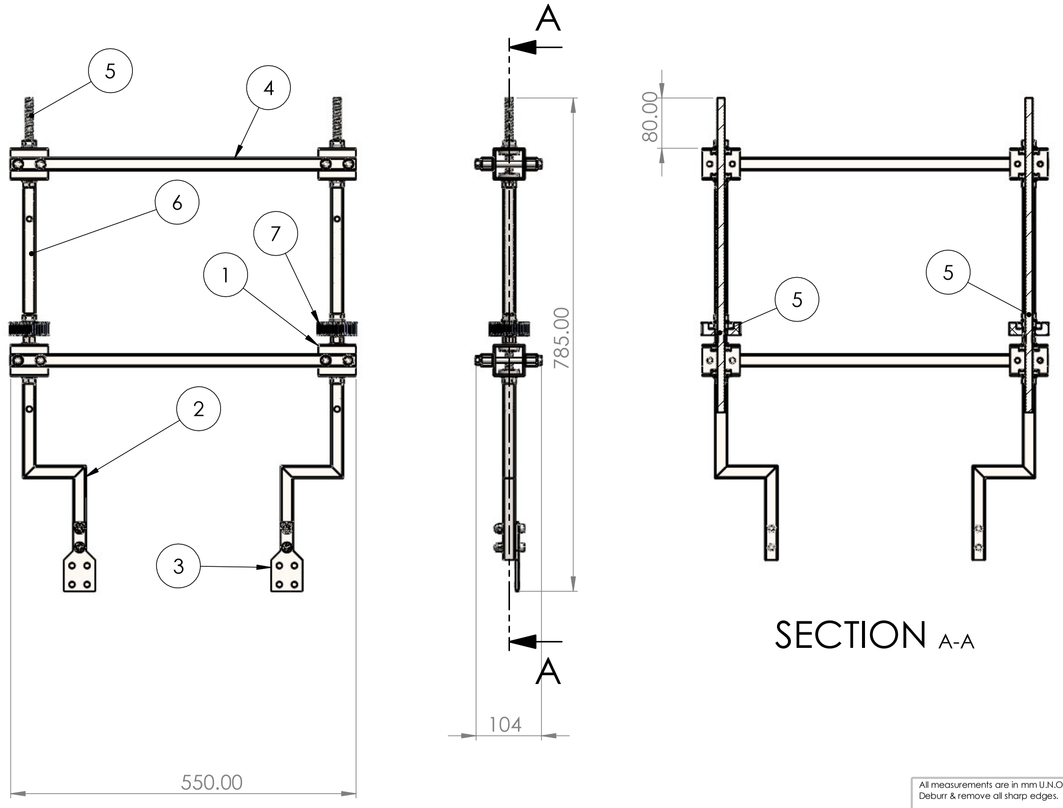
Bill of materials for base frame:

ITEM	QTY.	PART NUMBER	UNIT WEIGHT [kg]
1	4	001-001	0.22839
2	2	001-002	0.30949
3	2	001-003	0.13805
4	4	001-004	0.45551
5	2	001-005	0.4227
6	2	001-006	0.17854
7	2	001-P07	0.04332
8	11	Washer DIN 125 - A 13	-
9	24	Washer DIN 125 - A 8.4	-
10	4	DIN EN ISO 7040 - M8 - N	-
11	2	DIN EN ISO 7040 - M12 - C	-
12	4	DIN 912 M8 x 40 --- 28C	-
13	8	ISO 4017 - M8 x 100-C	-
14	8	ISO - 4032 - M8 - W - N	-
15	12	DIN934_M12	-

Next page will include the assembly of the base frame, and after that the subassembly-drawings of the base frame will be presented.

This document and all information and data herein or herewith are confedencial and propriertary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF



ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	4	001-001		0.22839
2	2	001-002		0.30949
3	2	001-003		0.13805
4	4	001-004		0.45551
5	2	001-005		0.4227
6	2	001-006		0.17854
7	2	001-P07		0.04332
8	11	Washer DIN 125 - A 13		
9	24	Washer DIN 125 - A 8.4		
10	4	DIN EN ISO 7040 - M8 - N		
11	2	DIN EN ISO 7040 - M12 - C		
12	4	DIN 912 M8 x 40 --- 28C		
13	8	ISO 4017 - M8 x 100-C		
14	8	ISO - 4032 - M8 - W - N		
15	12	DIN934_M12		

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:

REPLACES:

REPLACED BY:

TITLE:

Spesification:

DWG NO. 001-000

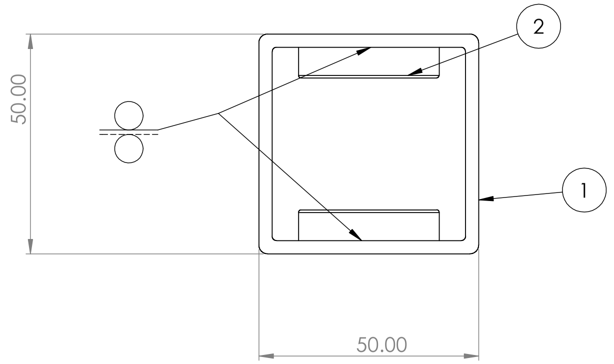
A3

WEIGHT: 5.18983

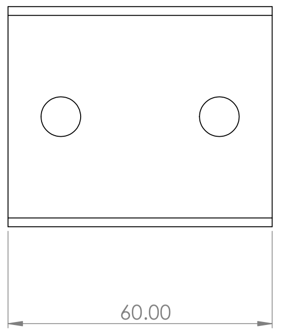
SCALE:1:7

SHEET 1 OF 1

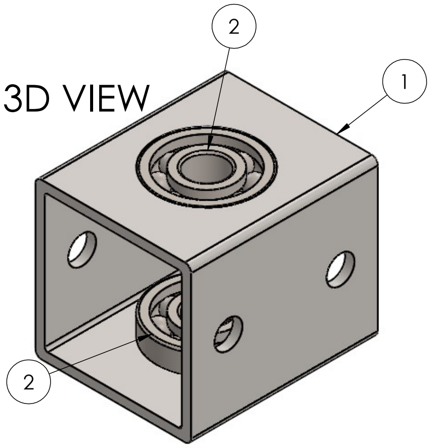
Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF



NOTE: Align bearings according to holes in item 1.



3D VIEW



FASTENING

ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	1	SHS (50x50x4)		
2	2	SKF - 6201 - 8,DE,NC,8_68		

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

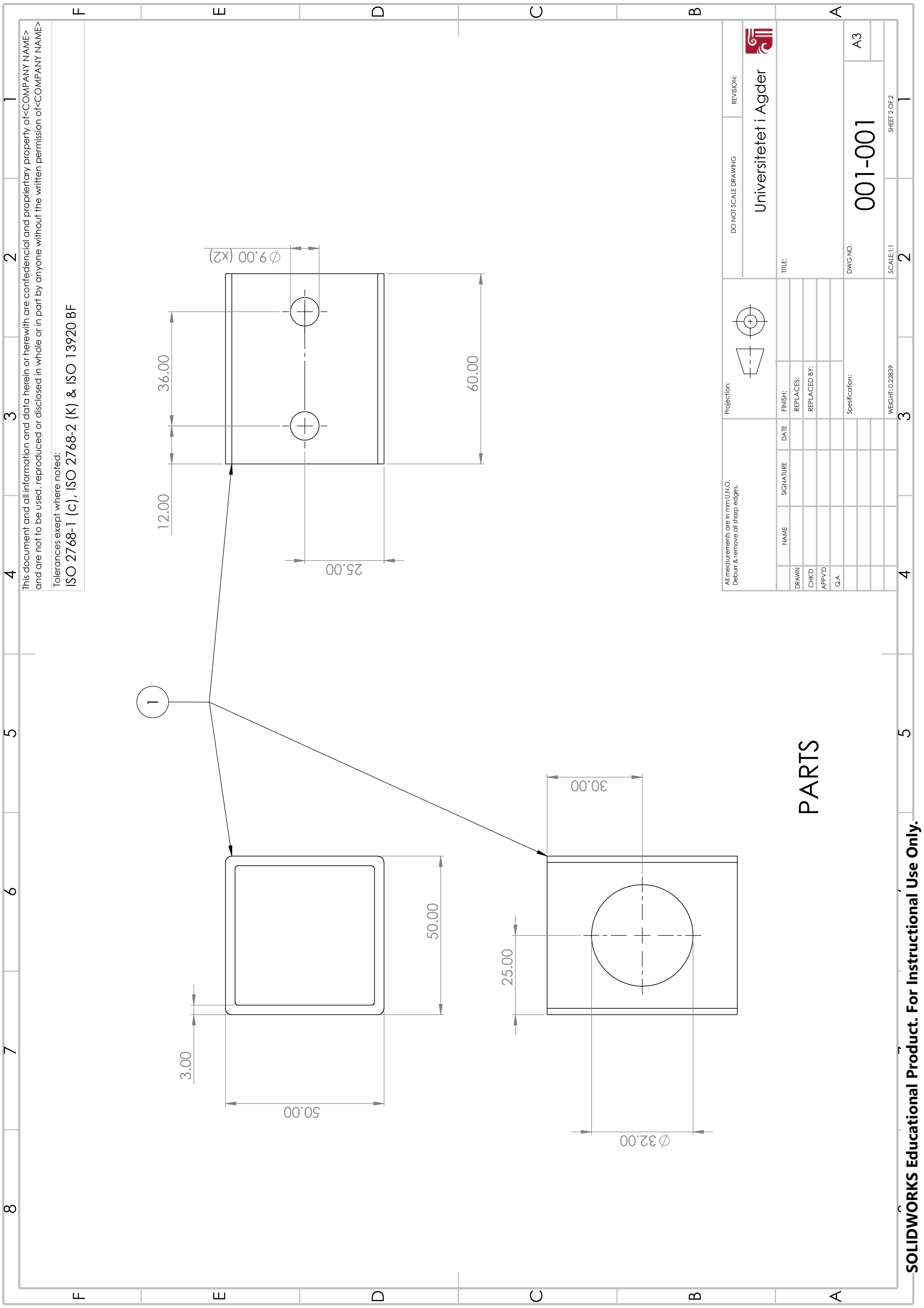
DWG NO. 001-001

A3

WEIGHT: 0.22839

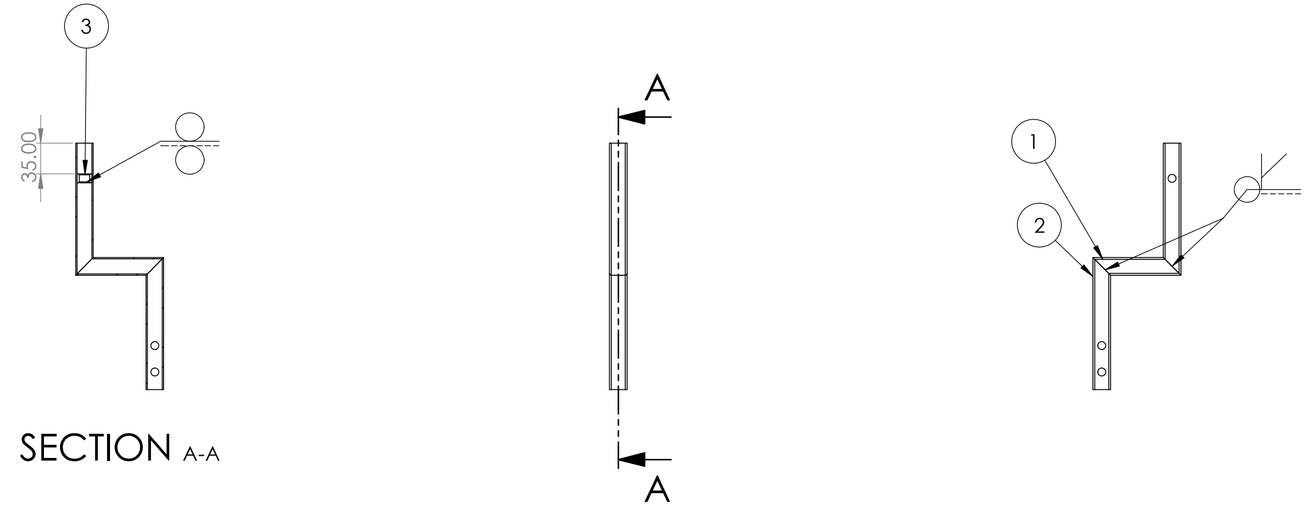
SCALE:1:1

SHEET 1 OF 2


This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME>
and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permision of<COMPANY NAME>
Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF
⌀9.00 (x2)
36.00
12.00
60.00
25.00
1
30.00
50.00
25.00
3.00
50.00
⌀32.00
PARTS
All measurements are in mm U.N.O.
Deburr & remove all sharp edges.
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
Q.A
Projection:
FINISH:
REPLACES:
REPLACED BY:
Spesification:
WEIGHT: 0.22839
DO NOT SCALE DRAWING
REVISION:
Universitetet i Agder
TITLE:
DWG NO.
001-001
SCALE:1:1
SHEET 2 OF 2
A3
F
E
D
C
B
A
1
2
3
4
5
6
7
8

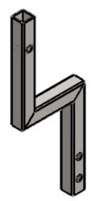
This document and all information and data herein or herewith are confedencial and propriertary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF



SECTION A-A

3D VIEW



WELDING

ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	1	SHS (20x20x1.5)		
2	1	SHS (20x20x1.5)		
3	1	M12 nut		
4	1	SHS (20x20x1.5)		

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:

REPLACES:

REPLACED BY:

TITLE:

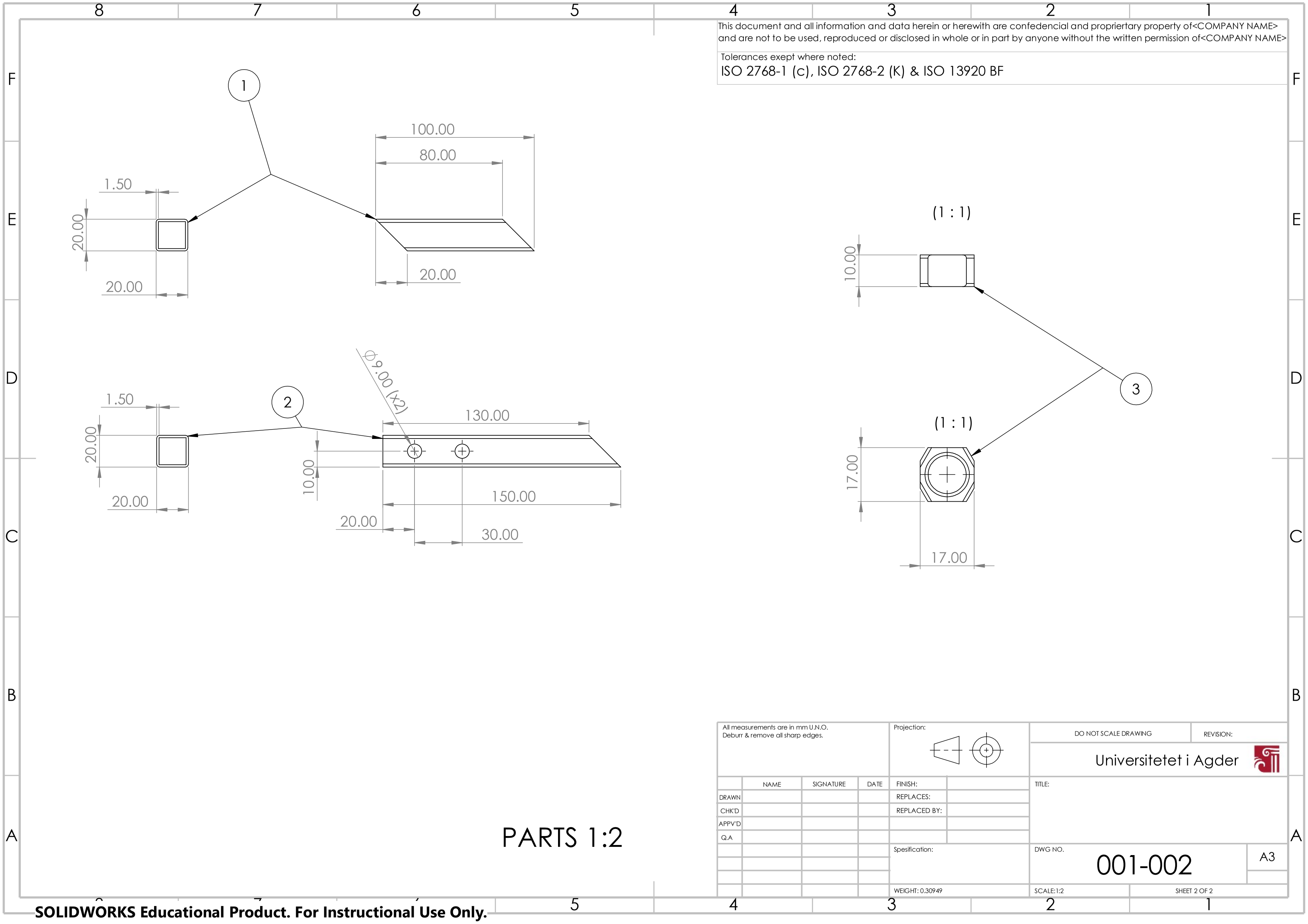
Spesification:

DWG NO. 001-002 A3

WEIGHT: 0.30949

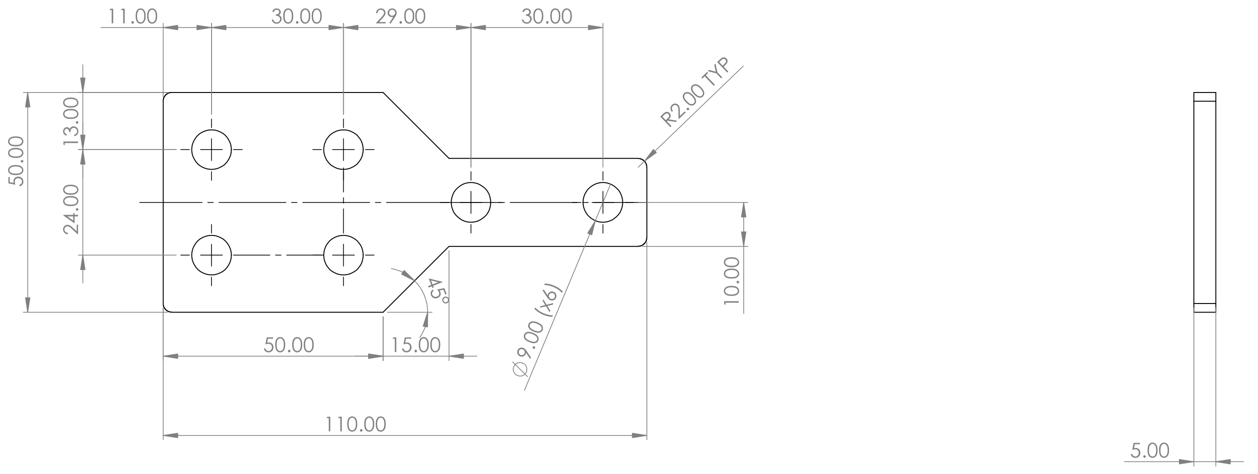
SCALE:1:5

SHEET 1 OF 2

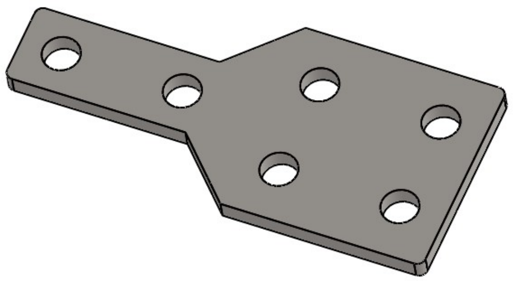

8
7
6
5
4
3
2
1
F
E
D
C
B
A
This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME>
and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>
Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF
1
1.50
20.00
20.00
100.00
80.00
20.00
2
1.50
20.00
20.00
⌀9.00 (x2)
130.00
10.00
150.00
20.00
30.00
(1 : 1)
10.00
3
(1 : 1)
17.00
17.00
PARTS 1:2
All measurements are in mm U.N.O.
Deburr & remove all sharp edges.
Projection:
DO NOT SCALE DRAWING
REVISION:
Universitetet i Agder
NAME
SIGNATURE
DATE
FINISH:
REPLACES:
REPLACED BY:
TITLE:
DRAWN
CHK'D
APPV'D
Q.A
Spesification:
DWG NO.
001-002
A3
WEIGHT: 0.30949
SCALE:1:2
SHEET 2 OF 2
SOLIDWORKS Educational Product. For Instructional Use Only.

This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF



3D VIEW



ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	1	PLATE 5mm		0.13805

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

Spesification:

WEIGHT: 0.13805

TITLE:

DWG NO. 001-003

A3

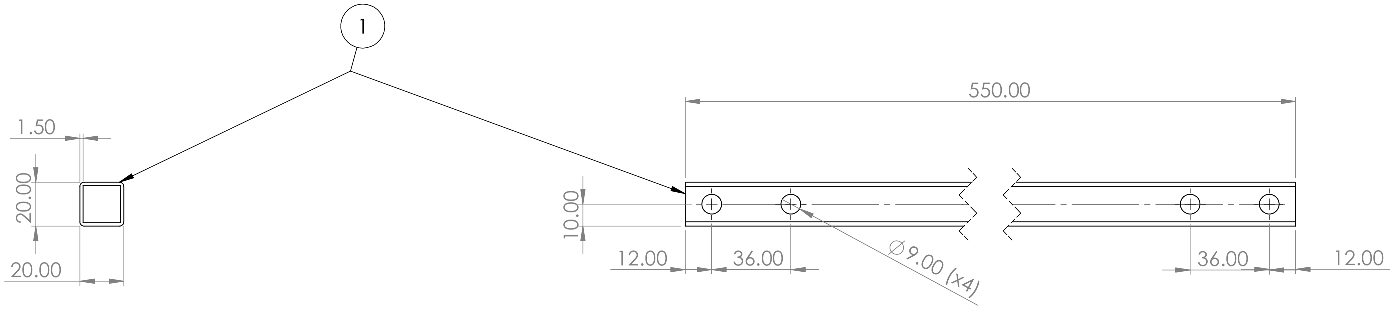
SCALE:1:1

SHEET 1 OF 1

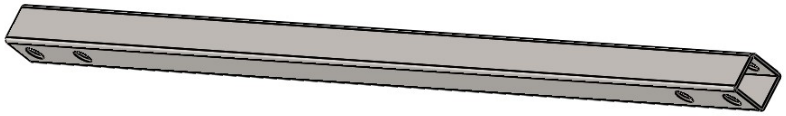
SOLIDWORKS Educational Product. For Instructional Use Only.

This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF



3D VIEW



ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	1	SHS 20x20x1.5		0.45551

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

DWG NO. 001-004

A3

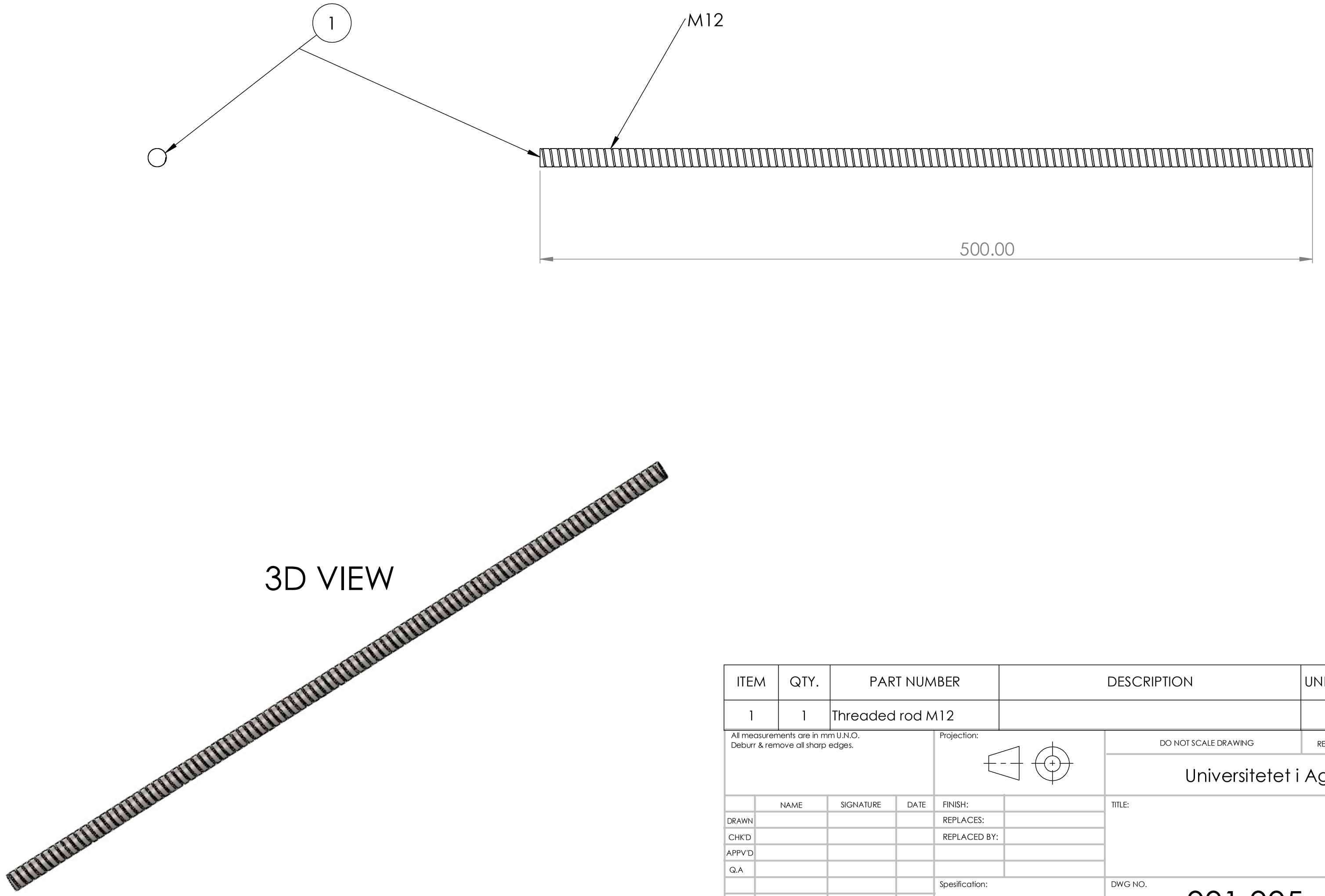
WEIGHT: 0.45551

SCALE:1:2

SHEET 1 OF 1

This document and all information and data herein or herewith are confedencial and propriertary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF



ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	1	Threaded rod M12		0.4227

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

DWG NO.
001-005

A3

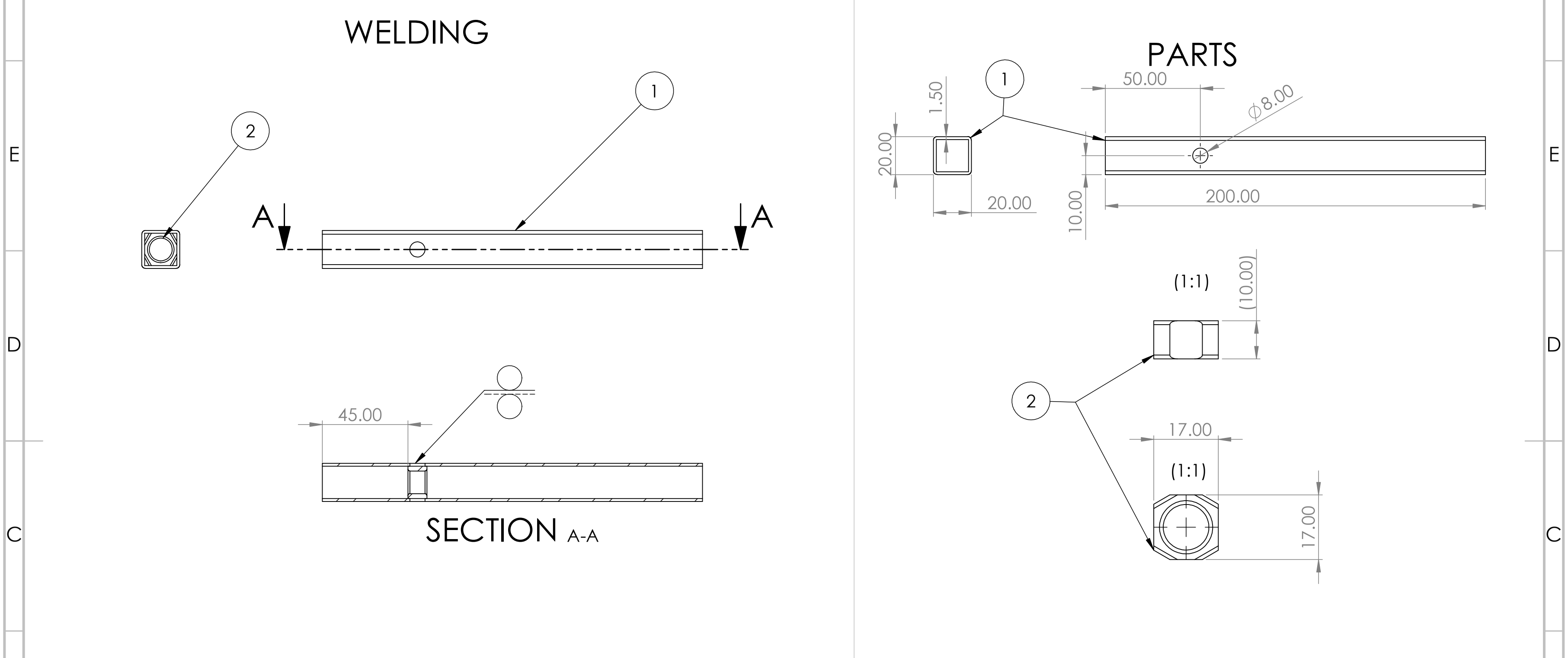
WEIGHT: 0.4227

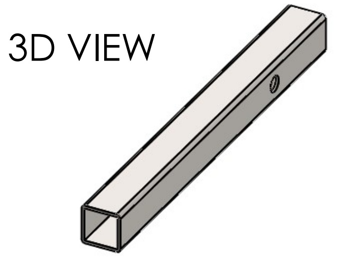
SCALE:1:2.5

SHEET 1 OF 1

This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF





ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	1	HUP (20x20x1.5)		
2	1	M12 nut		

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:

REPLACES:

REPLACED BY:

TITLE:

Spesification:

DWG NO. 001-006

A3

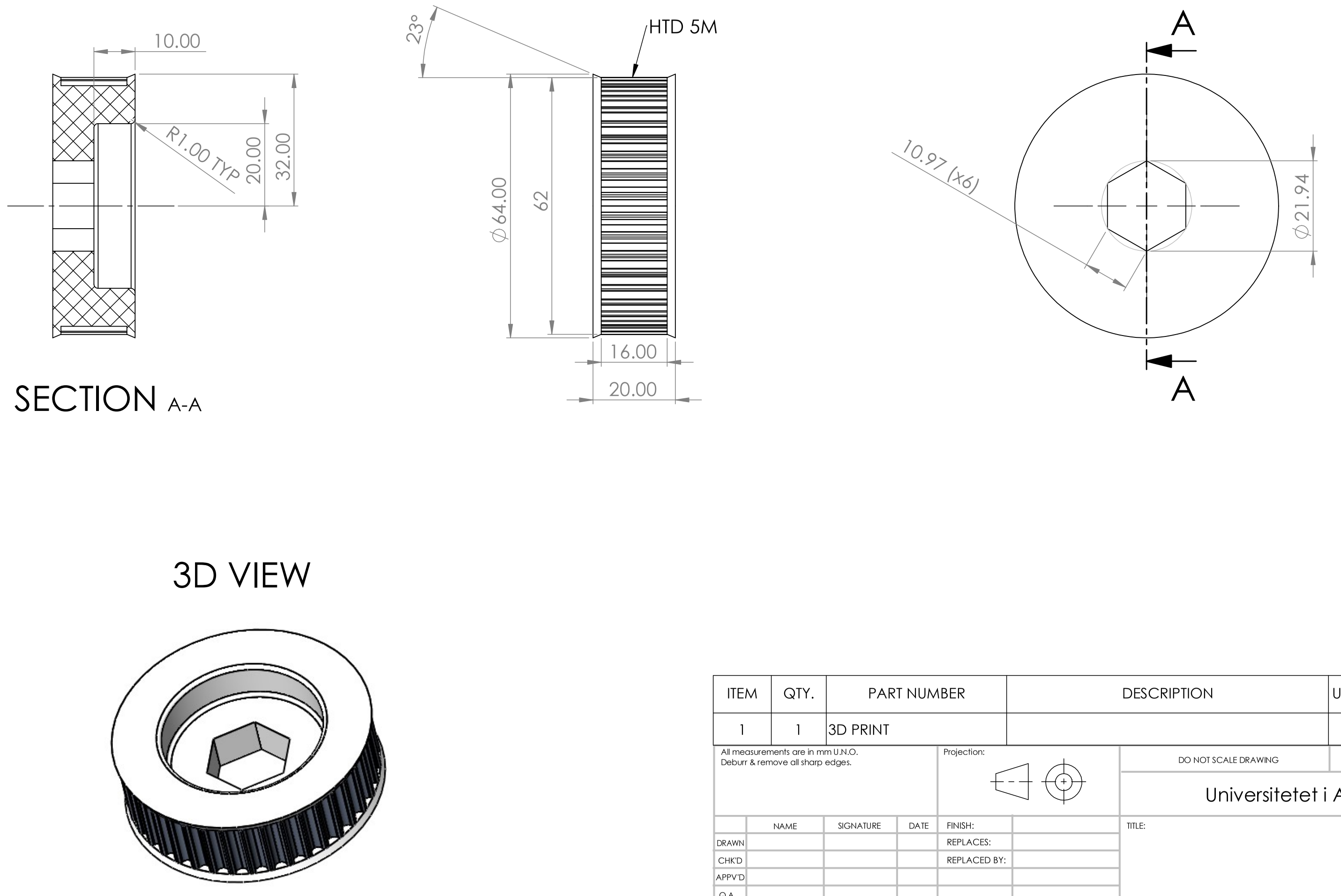
WEIGHT: 0.17854

SCALE:1:2

SHEET 1 OF 1

This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF



ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	1	3D PRINT		0.04332

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

Spesification:

WEIGHT: 0.04332

TITLE:

DWG NO. 001-P07

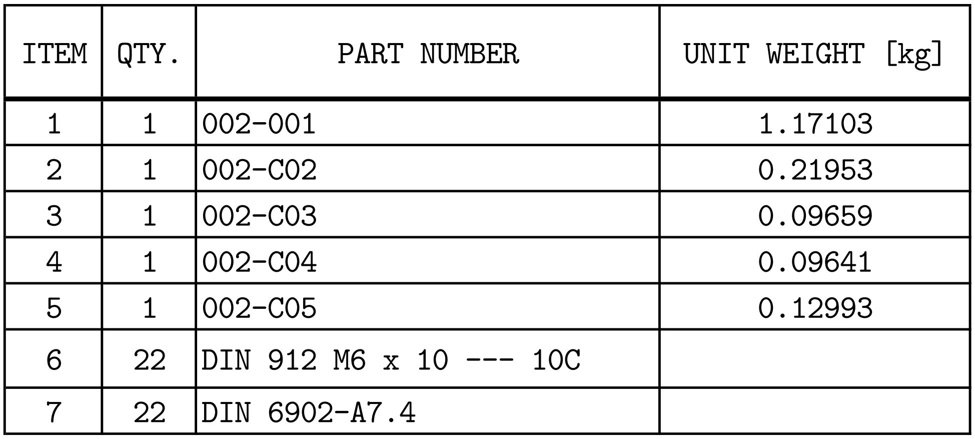
A3

SCALE:1:1

SHEET 1 OF 1

A.3 002-000: Component Box

Bill of materials for component box:

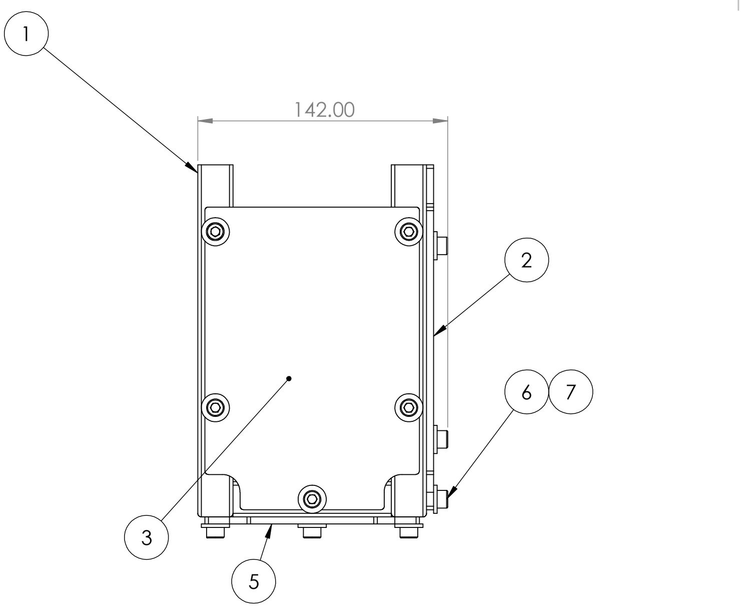


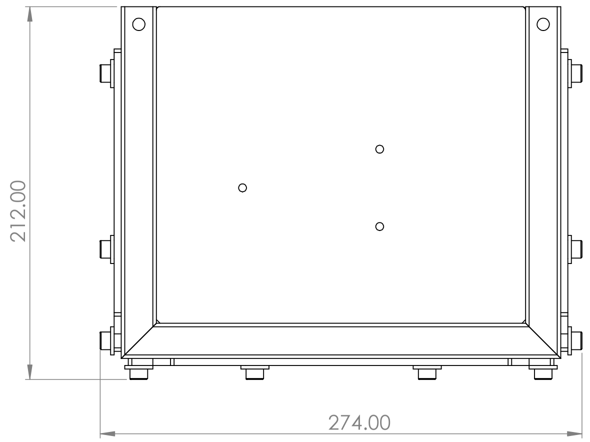
ITEM	QTY.	PART NUMBER	UNIT WEIGHT [kg]
1	1	002-001	1.17103
2	1	002-C02	0.21953
3	1	002-C03	0.09659
4	1	002-C04	0.09641
5	1	002-C05	0.12993
6	22	DIN 912 M6 x 10 --- 10C	
7	22	DIN 6902-A7.4	

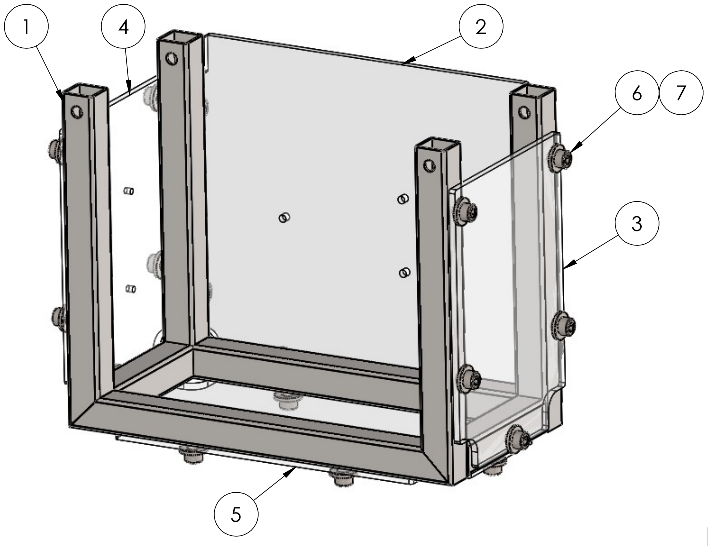
Next page will include the assembly of the base frame, and after that the subassembly-drawings of component box will be presented.

This document and all information and data herein or herewith are confedencial and propriertary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF







ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	1	002-001		1.17103
2	1	002-C02		0.21953
3	1	002-C03		0.09659
4	1	002-C04		0.09641
5	1	002-C05		0.12993
6	22	DIN 912 M6 x 10 --- 10C		
7	22	DIN 6902-A7.4		

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

Spesification:

WEIGHT: 1.73528

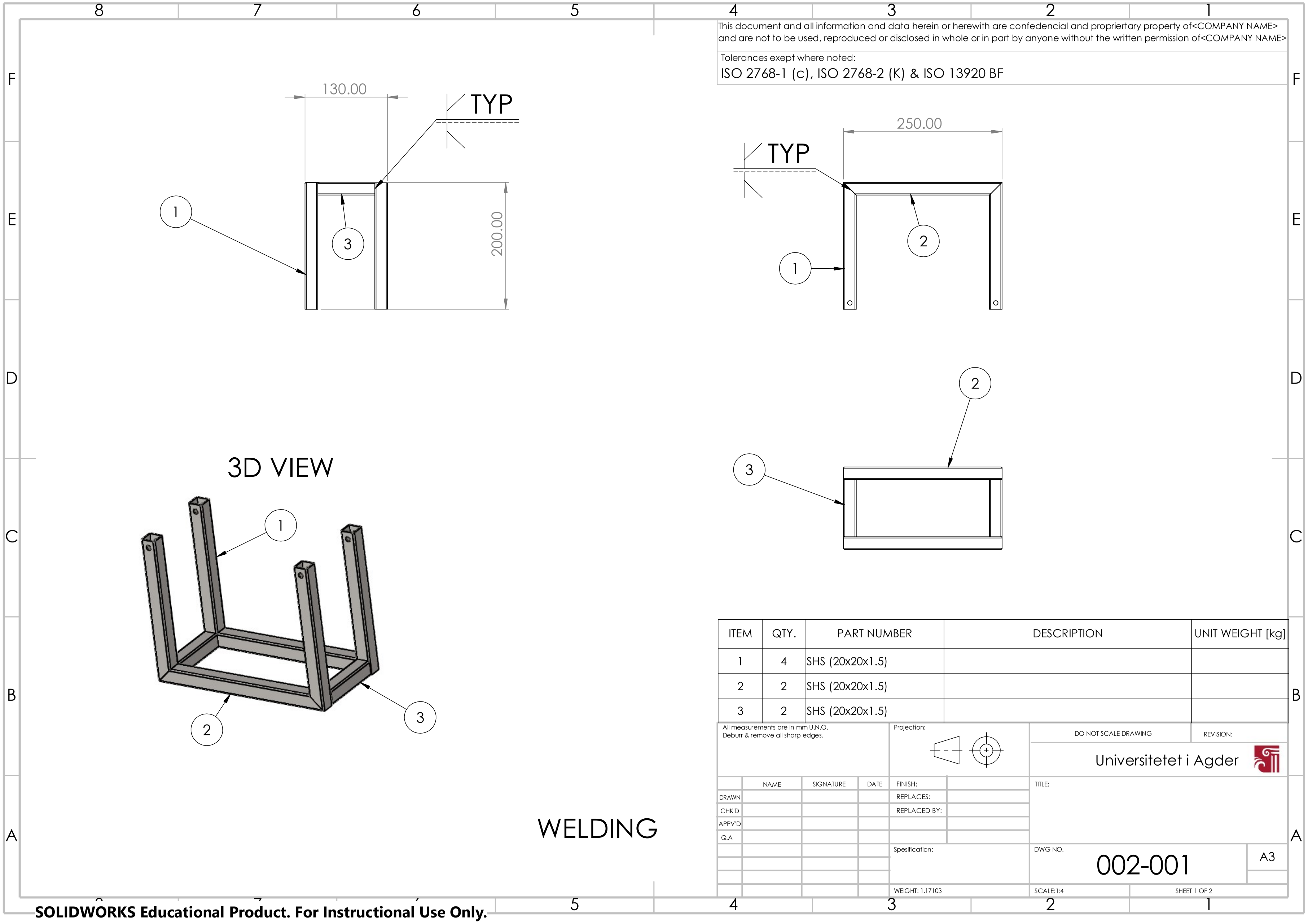
TITLE:

DWG NO.
002-000

A3

SCALE:1:2.5

SHEET 1 OF 1



This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF

ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	4	SHS (20x20x1.5)		
2	2	SHS (20x20x1.5)		
3	2	SHS (20x20x1.5)		

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

DWG NO.
002-001

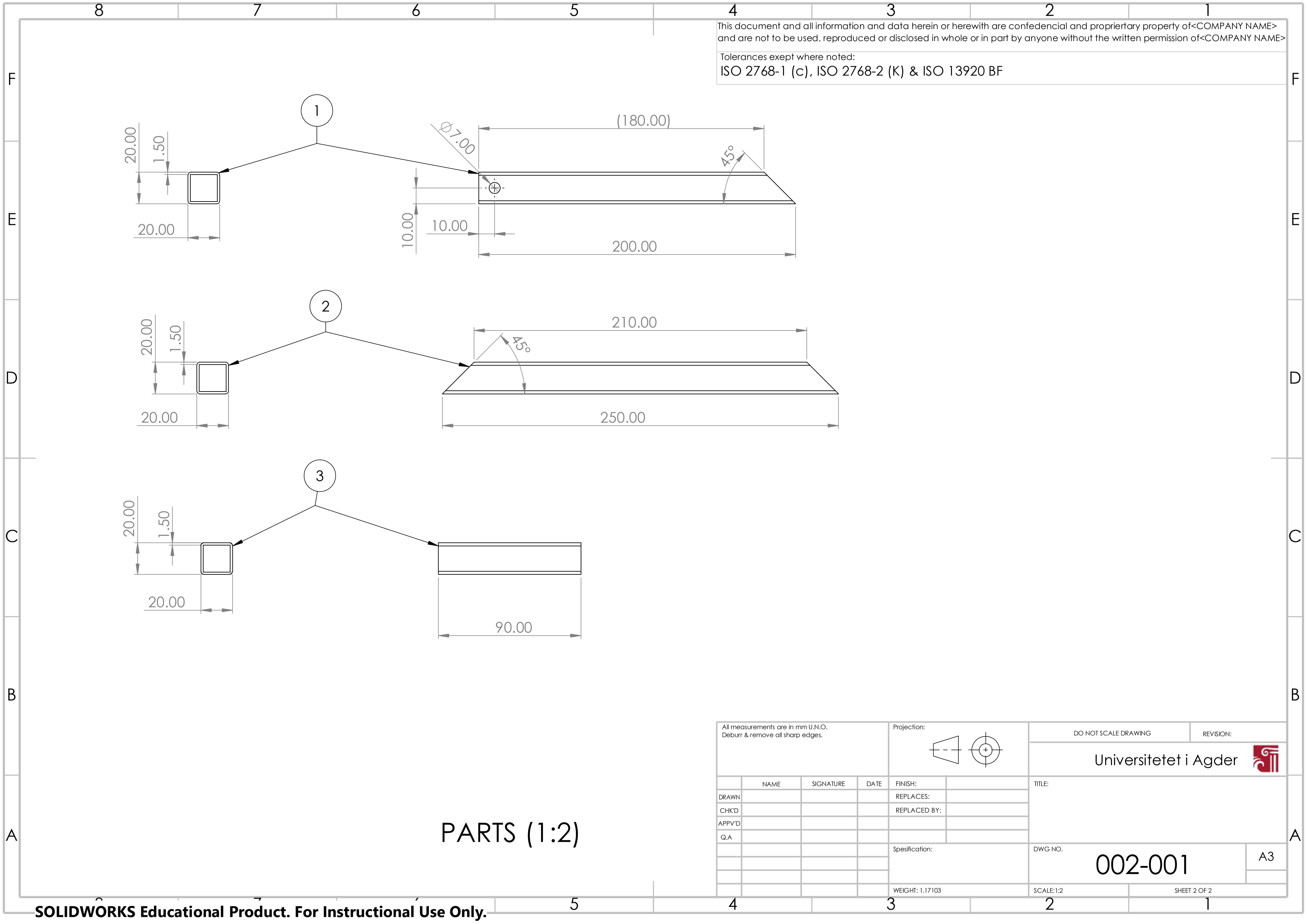
A3

WEIGHT: 1.17103

SCALE:1:4

SHEET 1 OF 2

SOLIDWORKS Educational Product. For Instructional Use Only.



This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF

PARTS (1:2)

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

	NAME	SIGNATURE	DATE	FINISH:	
DRAWN				REPLACES:	
CHK'D				REPLACED BY:	
APPV'D					
Q.A					

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

TITLE:

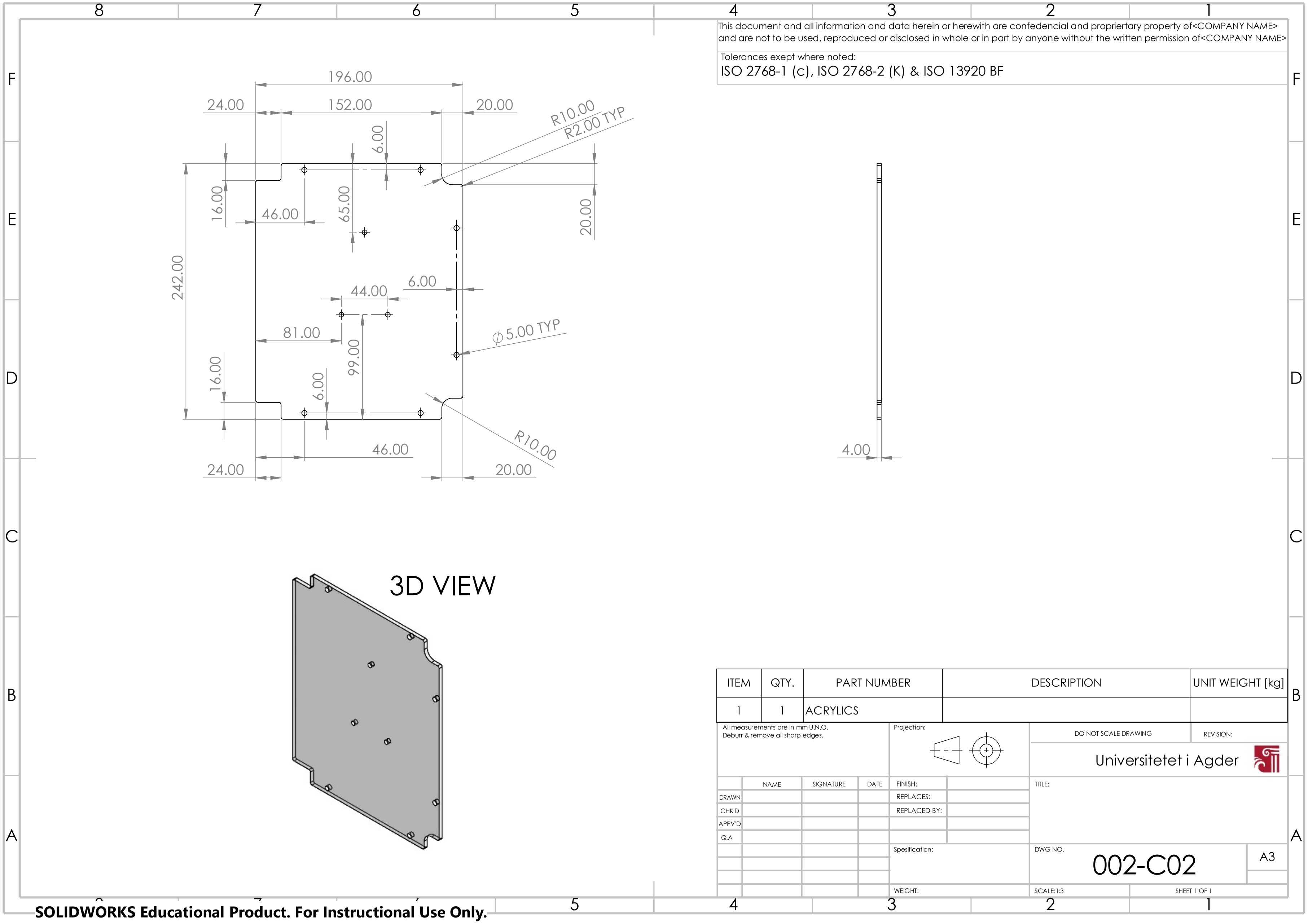
Spesification:

DWG NO. 002-001 A3

WEIGHT: 1.17103

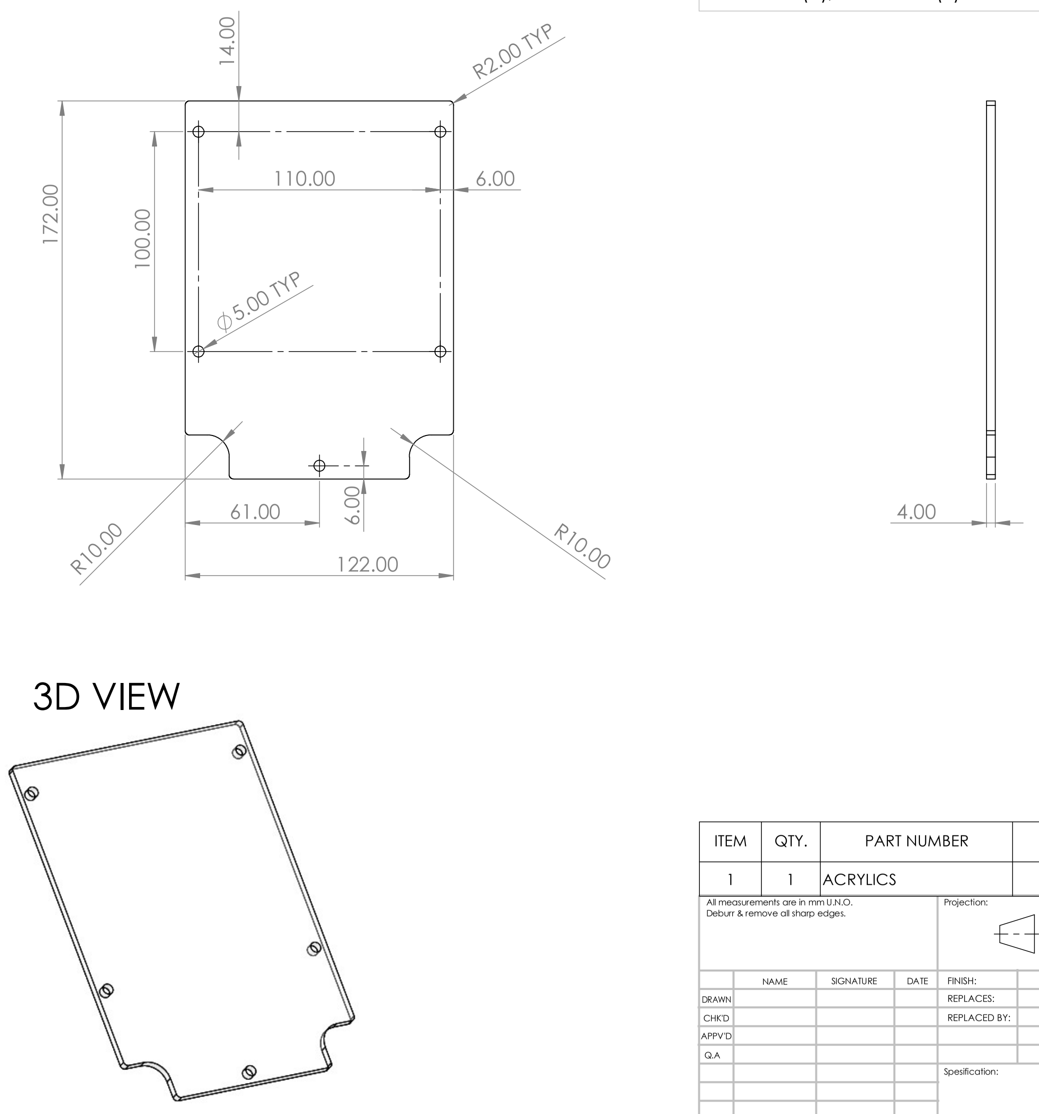
SCALE:1:2

SHEET 2 OF 2


This document and all information and data herein or herewith are confedencial and propriertary property of<COMPANY NAME>
and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>
Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF
196.00
24.00
152.00
20.00
R10.00
R2.00 TYP
6.00
16.00
46.00
65.00
20.00
242.00
44.00
6.00
81.00
ϕ5.00 TYP
16.00
6.00
99.00
46.00
R10.00
24.00
20.00
4.00
3D VIEW
ITEM
QTY.
PART NUMBER
DESCRIPTION
UNIT WEIGHT [kg]
1
1
ACRYLICS
All measurements are in mm U.N.O.
Deburr & remove all sharp edges.
Projection:
DO NOT SCALE DRAWING
REVISION:
Universitetet i Agder
NAME
SIGNATURE
DATE
FINISH:
REPLACES:
REPLACED BY:
TITLE:
DRAWN
CHK'D
APPV'D
Q.A
Spesification:
DWG NO.
002-C02
A3
WEIGHT:
SCALE:1:3
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.
8
7
6
5
4
3
2
1
F
E
D
C
B
A

This document and all information and data herein or herewith are confedencial and propriertary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF



ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	1	ACRYLICS		

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

DWG NO.
002-C03

A3

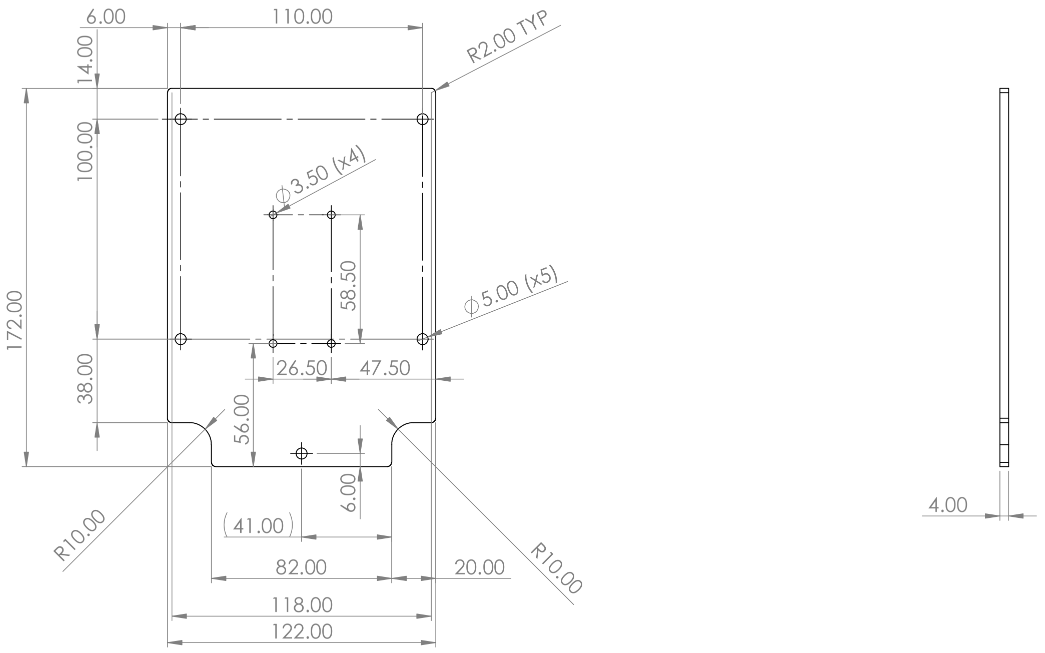
WEIGHT:

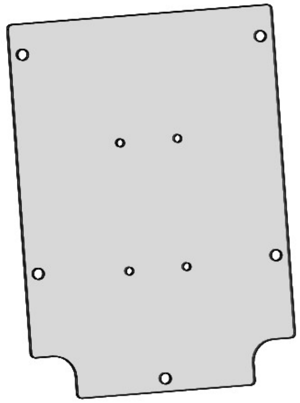
SCALE:1:2

SHEET 1 OF 1

This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF




3D VIEW

ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	1	ACRYLICS		

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

DWG NO.
002-C04

A3

WEIGHT:

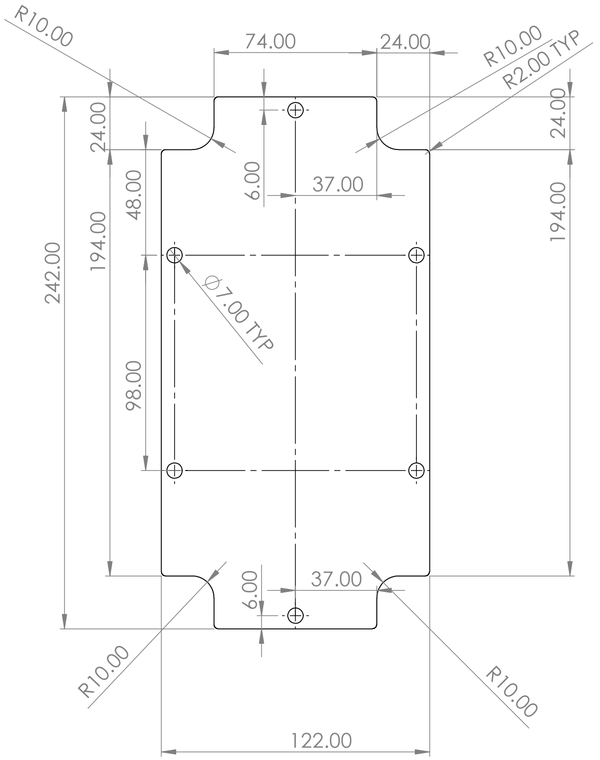
SCALE:1:2

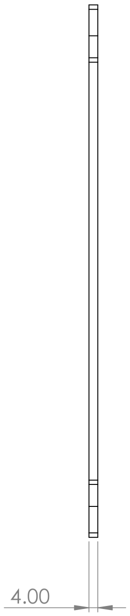
SHEET 1 OF 1

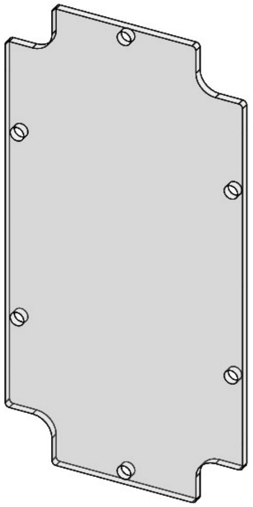
SOLIDWORKS Educational Product. For Instructional Use Only.

This document and all information and data herein or herewith are confedencial and proprirtary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF







3D VIEW

ITEM	QTY.	PART NUMBER	DESCRIPTION	UNIT WEIGHT [kg]
1	1	ACRYLICS		

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

DWG NO.
002-C05

A3

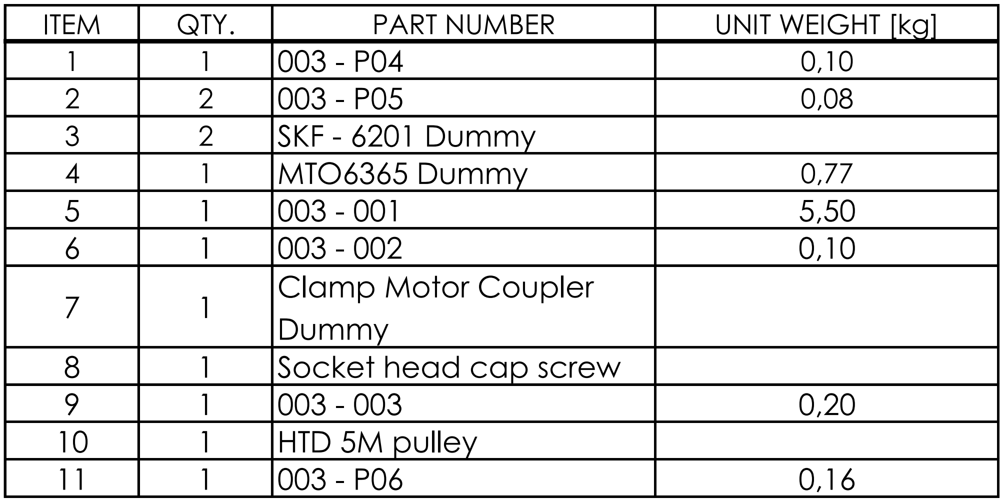
WEIGHT:

SCALE:1:2

SHEET 1 OF 1

A.4 003-000: Inertia Wheel System

Bill of materials for the inertia wheel system:

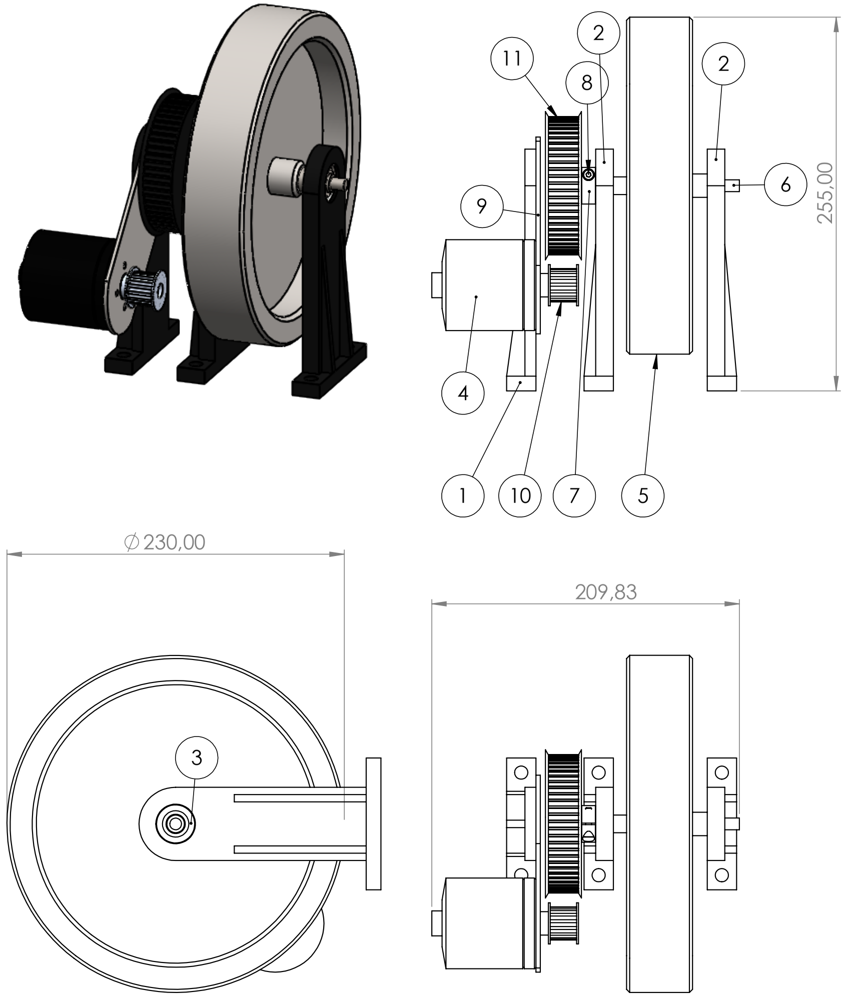


ITEM	QTY.	PART NUMBER	UNIT WEIGHT [kg]
1	1	003 - P04	0,10
2	2	003 - P05	0,08
3	2	SKF - 6201 Dummy	
4	1	MTO6365 Dummy	0,77
5	1	003 - 001	5,50
6	1	003 - 002	0,10
7	1	Clamp Motor Coupler Dummy	
8	1	Socket head cap screw	
9	1	003 - 003	0,20
10	1	HTD 5M pulley	
11	1	003 - P06	0,16

Next page will include the assembly of the total system, and after that the subassembly-drawings will be presented.

This document and all information and data herein or herewith are confedencial and proprierty property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF



ITEM NO.	PART NUMBER	DESCRIPTION	Default/ QTY.
1	003 - P04		1
2	003 - P05		2
3	SKF - 6201		2
4	MTO6365		1
5	003 - 001		1
6	003 - 002		1
7	Clamp Motor Coupler Dummy		1
8	Socket head cap screw		1
9	003 - 003		1
10	HTD 5M pulley	d = 24mm w = 15mm	1
11	003 - P06		1

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

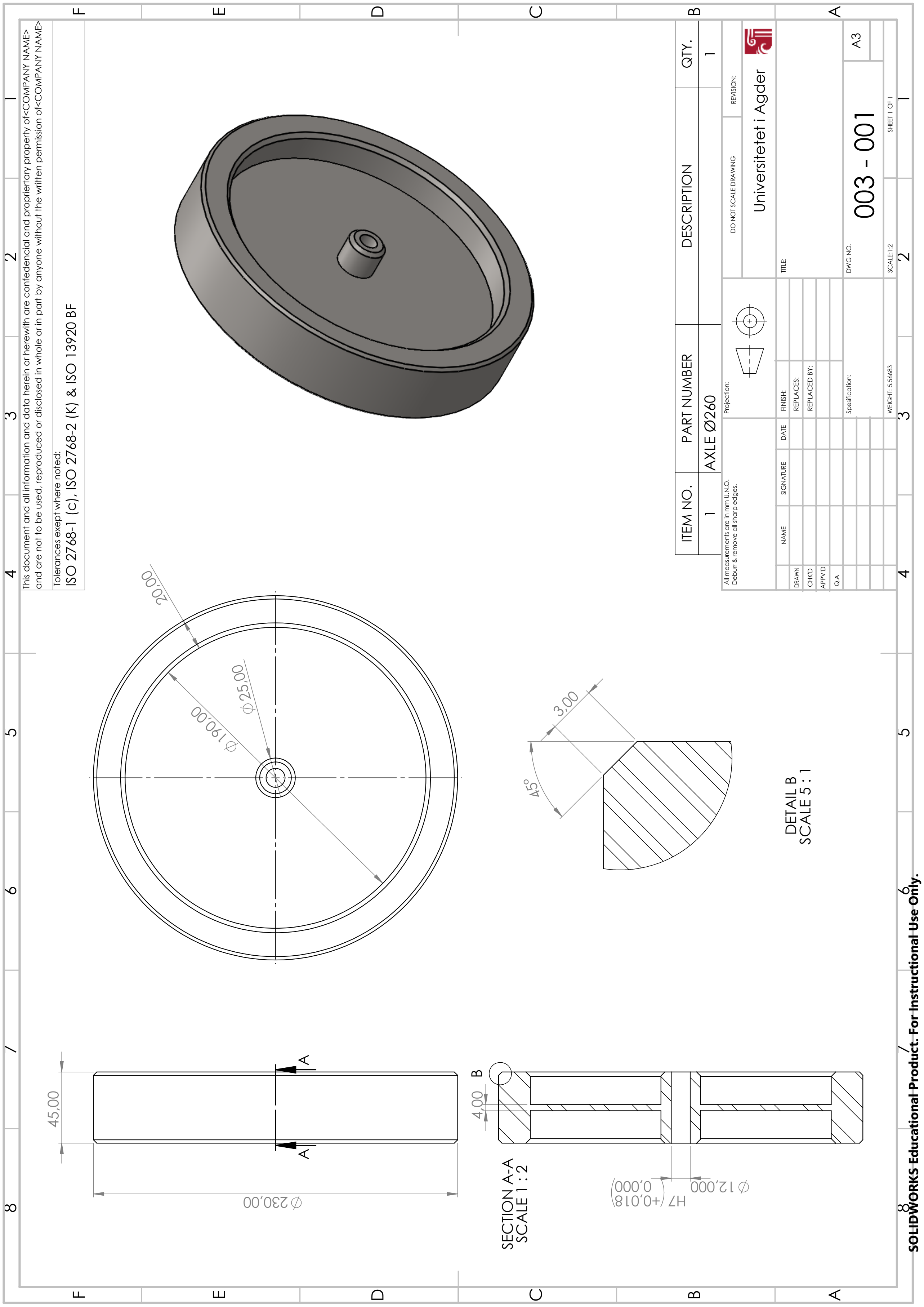
DWG NO. 003 - 000

A3

WEIGHT: 7.15193

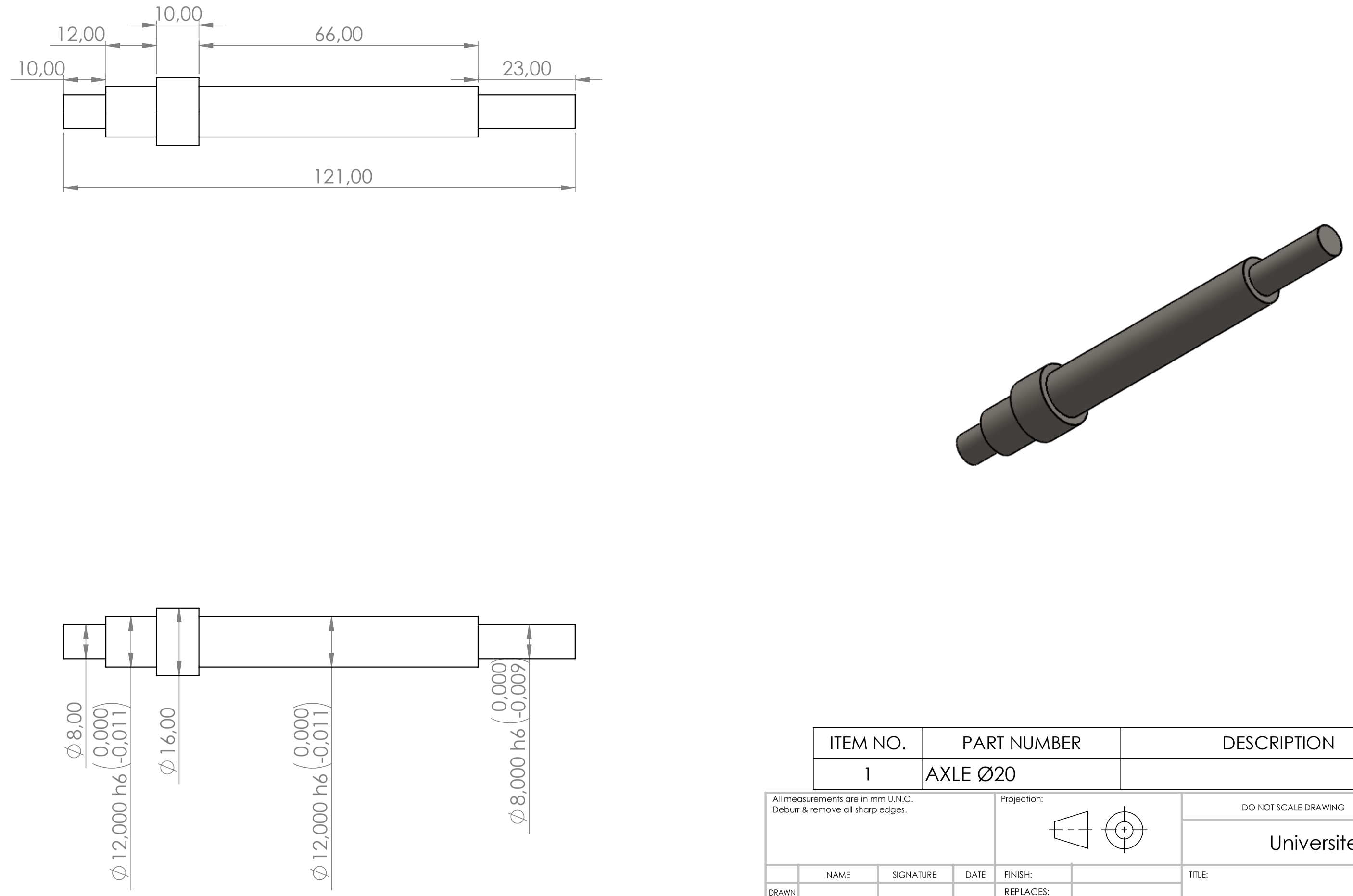
SCALE:1:3

SHEET 1 OF 1


This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME>
and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>
Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF
ITEM NO.
PART NUMBER
DESCRIPTION
QTY.
1
AXLE Ø260
1
All measurements are in mm U.N.O.
Debur & remove all sharp edges.
Projection:
DO NOT SCALE DRAWING
REVISION:
Universitetet i Agder
NAME
SIGNATURE
DATE
FINISH:
REPLACES:
REPLACED BY:
DRAWN
CHK'D
APPV'D
Q.A
TITLE:
Spesification:
DWG NO.
003 - 001
A3
SCALE:1:2
WEIGHT: 5.56683
SHEET 1 OF 1
20,00
Ø25,00
Ø190,00
3,00
45°
DETAIL B
SCALE 5 : 1
45,00
Ø230,00
A
A
B
4,00
Ø12,000
H7 (+0,018 / 0,000)
SECTION A-A
SCALE 1 : 2
SOLIDWORKS Educational Product. For Instructional Use Only.
A
B
C
D
E
F
1
2
3
4
5
6
7
8

This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF


10,00
12,00
10,00
66,00
23,00
121,00
⌀8,00
⌀12,000 h6 (0,000 / -0,011)
⌀16,00
⌀12,000 h6 (0,000 / -0,011)
⌀8,000 h6 (0,000 / -0,009)

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	AXLE Ø20		1

All measurements are in mm U.N.O.
Debur & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

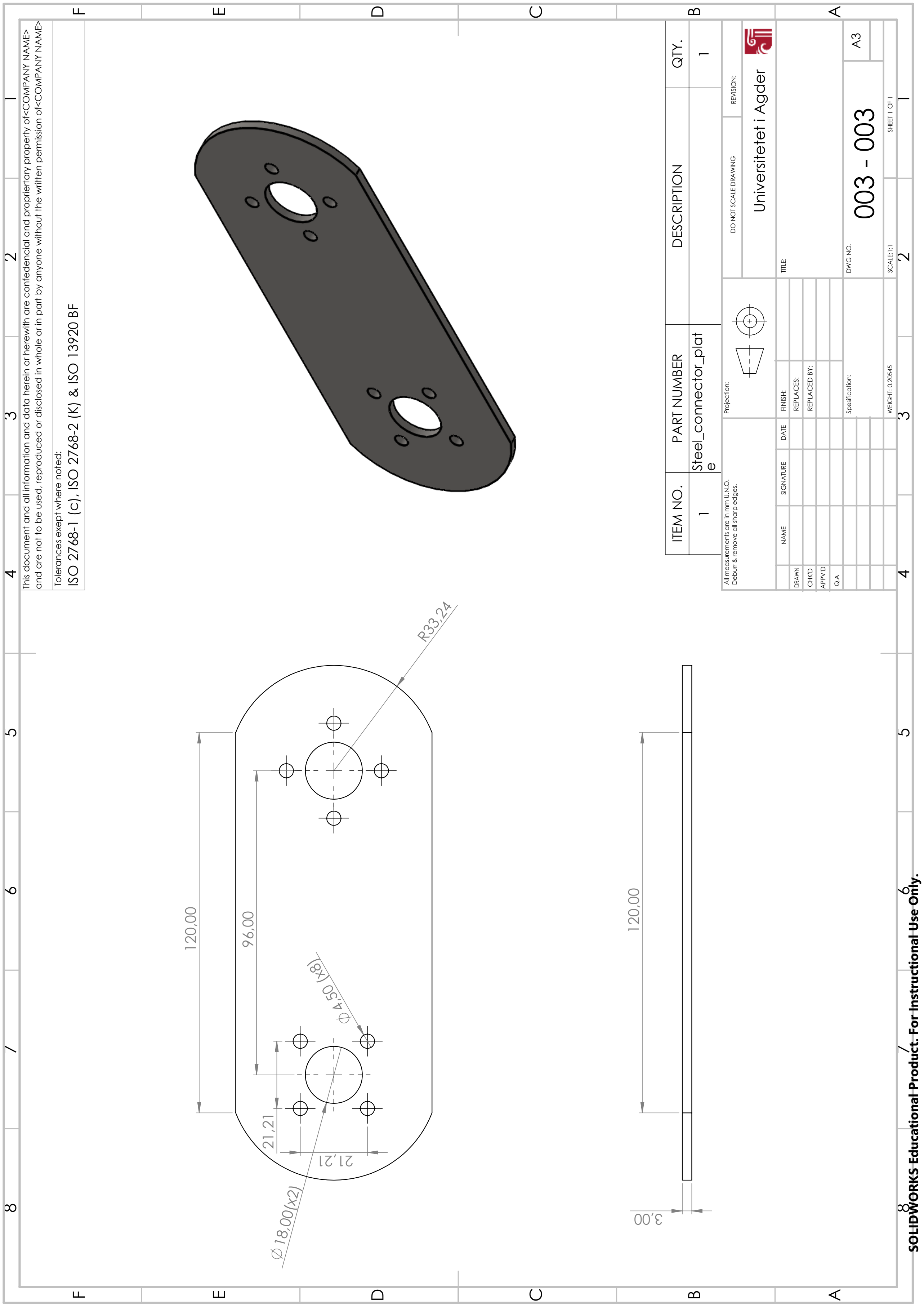
DWG NO. 003 - 002

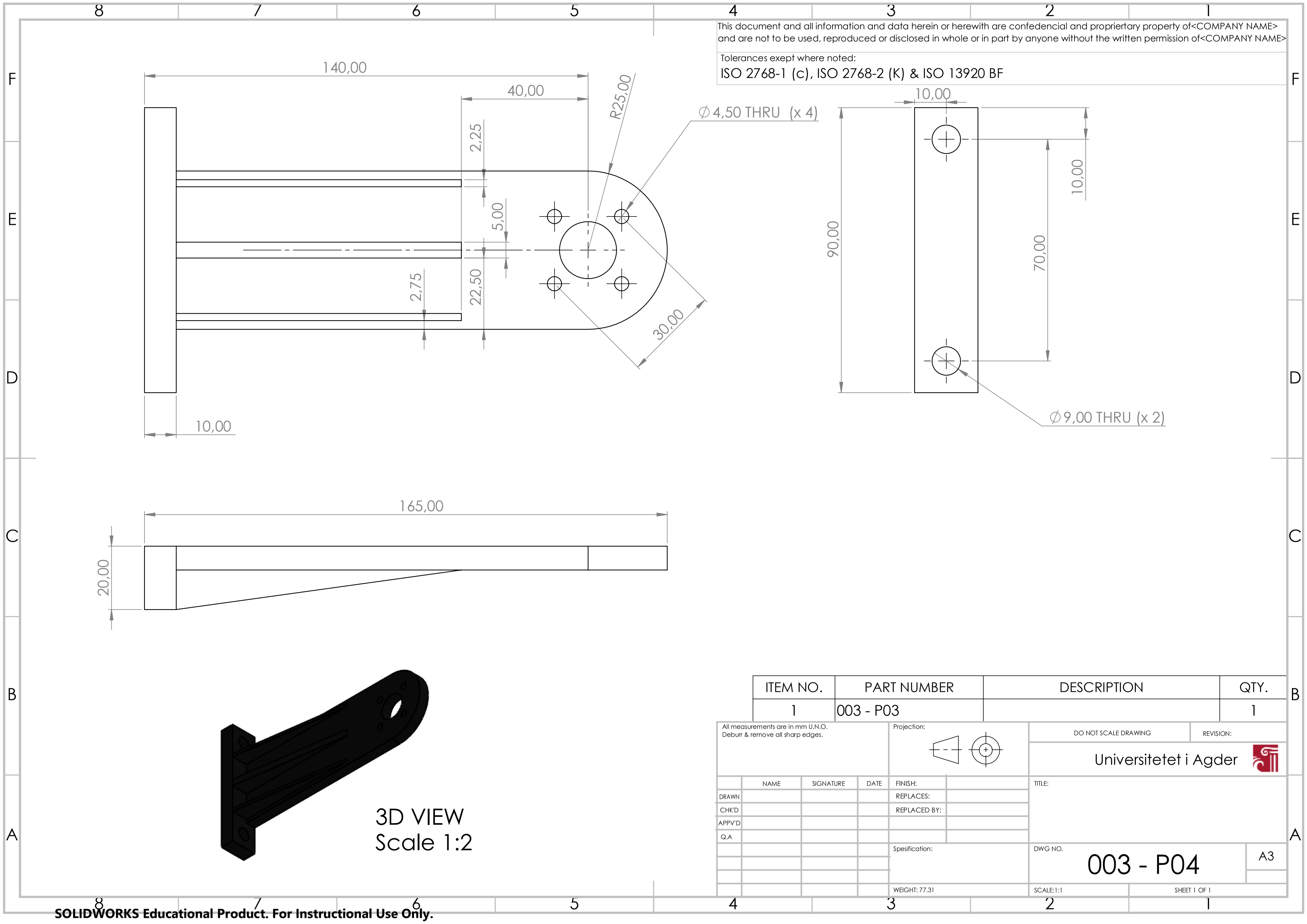
A3

WEIGHT: 0.09743

SCALE:1:1

SHEET 1 OF 1


This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME>
and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>
Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF
ITEM NO.
PART NUMBER
DESCRIPTION
QTY.
1
Steel_connector_plate
1
All measurements are in mm U.N.O.
Debur & remove all sharp edges.
Projection:
DO NOT SCALE DRAWING
REVISION:
Universitetet i Agder
NAME
SIGNATURE
DATE
FINISH:
REPLACES:
REPLACED BY:
TITLE:
DRAWN
CHK'D
APPV'D
Q.A
Spesification:
DWG NO.
003 - 003
A3
WEIGHT: 0.20545
SCALE:1:1
SHEET 1 OF 1
R33,24
120,00
96,00
⌀4,50 (x8)
21,21
21,21
⌀18,00(x2)
120,00
3,00
SOLIDWORKS Educational Product. For Instructional Use Only.
A
B
C
D
E
F
1
2
3
4
5
6
7
8



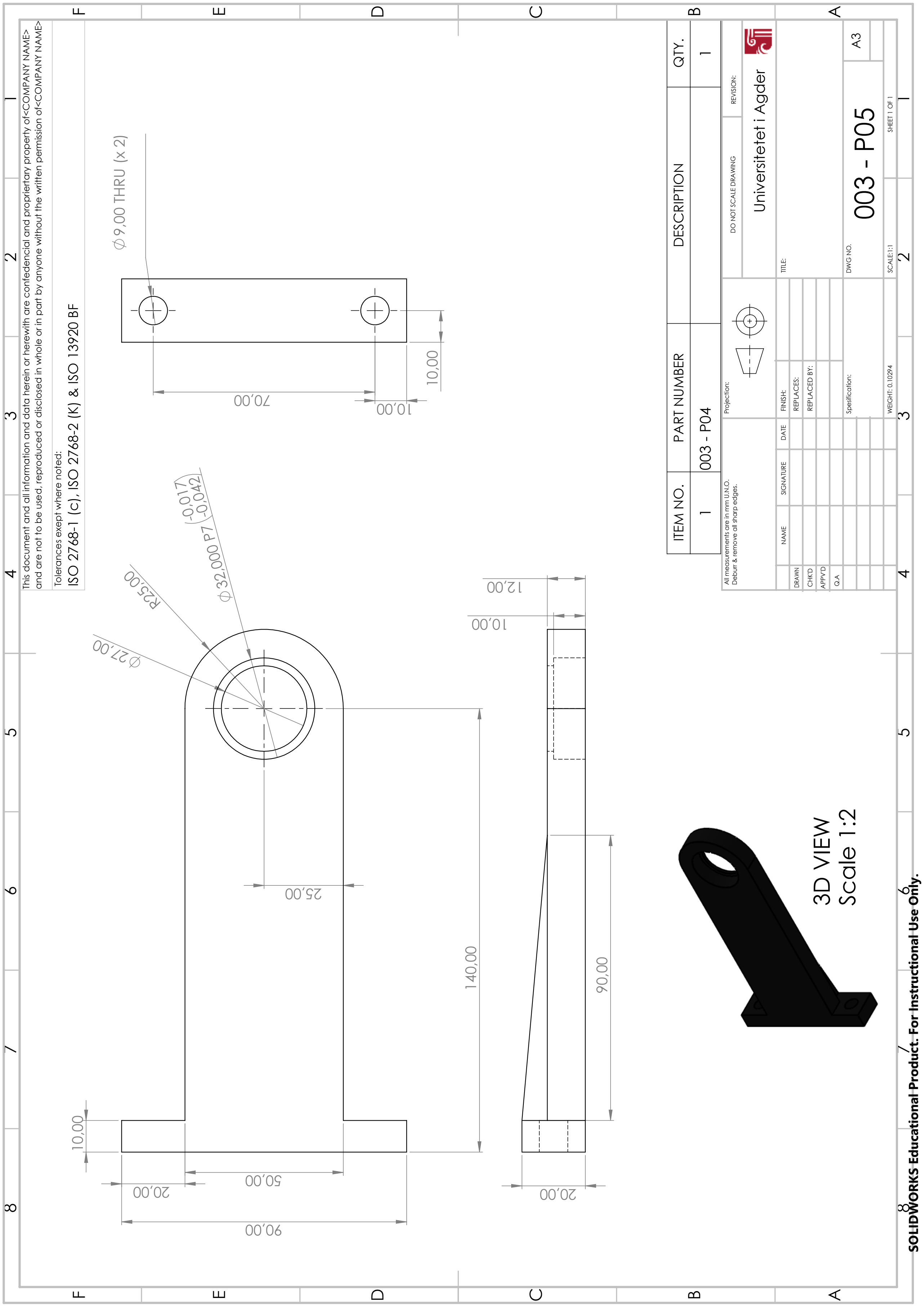
This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

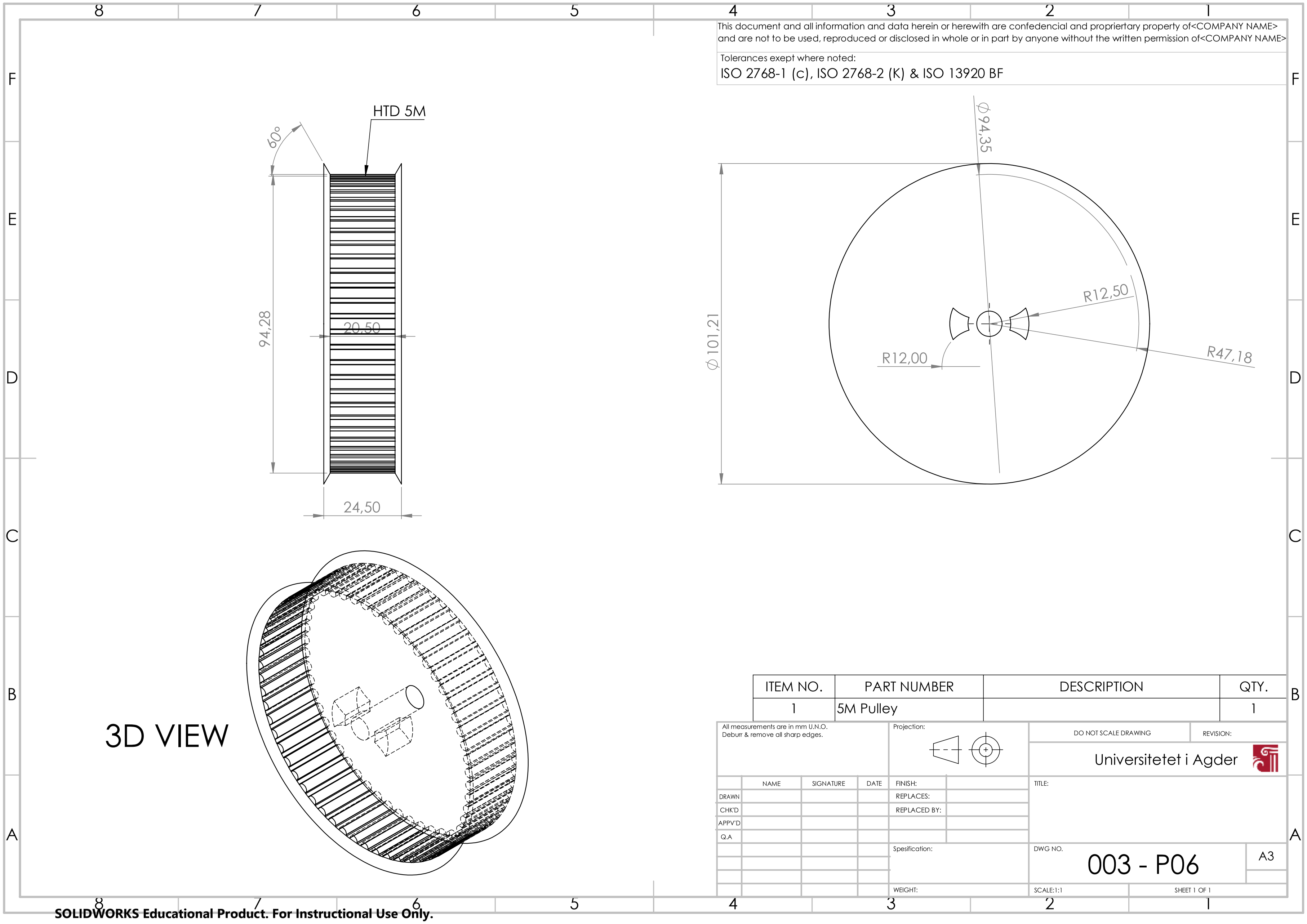
Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	003 - P03		1

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			


This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME>
and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>
Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF
⌀ 9,00 THRU (x 2)
70,00
10,00
10,00
R25,00
⌀ 27,00
⌀ 32,000 P7 (-0,017 / -0,042)
25,00
140,00
10,00
20,00
50,00
90,00
12,00
10,00
90,00
20,00
3D VIEW
Scale 1:2
ITEM NO.
PART NUMBER
DESCRIPTION
QTY.
1
003 - P04
1
All measurements are in mm U.N.O.
Debure & remove all sharp edges.
Projection:
DO NOT SCALE DRAWING
REVISION:
Universitetet i Agder
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
Q.A
FINISH:
REPLACES:
REPLACED BY:
TITLE:
Spesification:
DWG NO.
003 - P05
A3
WEIGHT: 0.10294
SCALE:1:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.
A
B
C
D
E
F
1
2
3
4
5
6
7
8



This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	5M Pulley		1

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

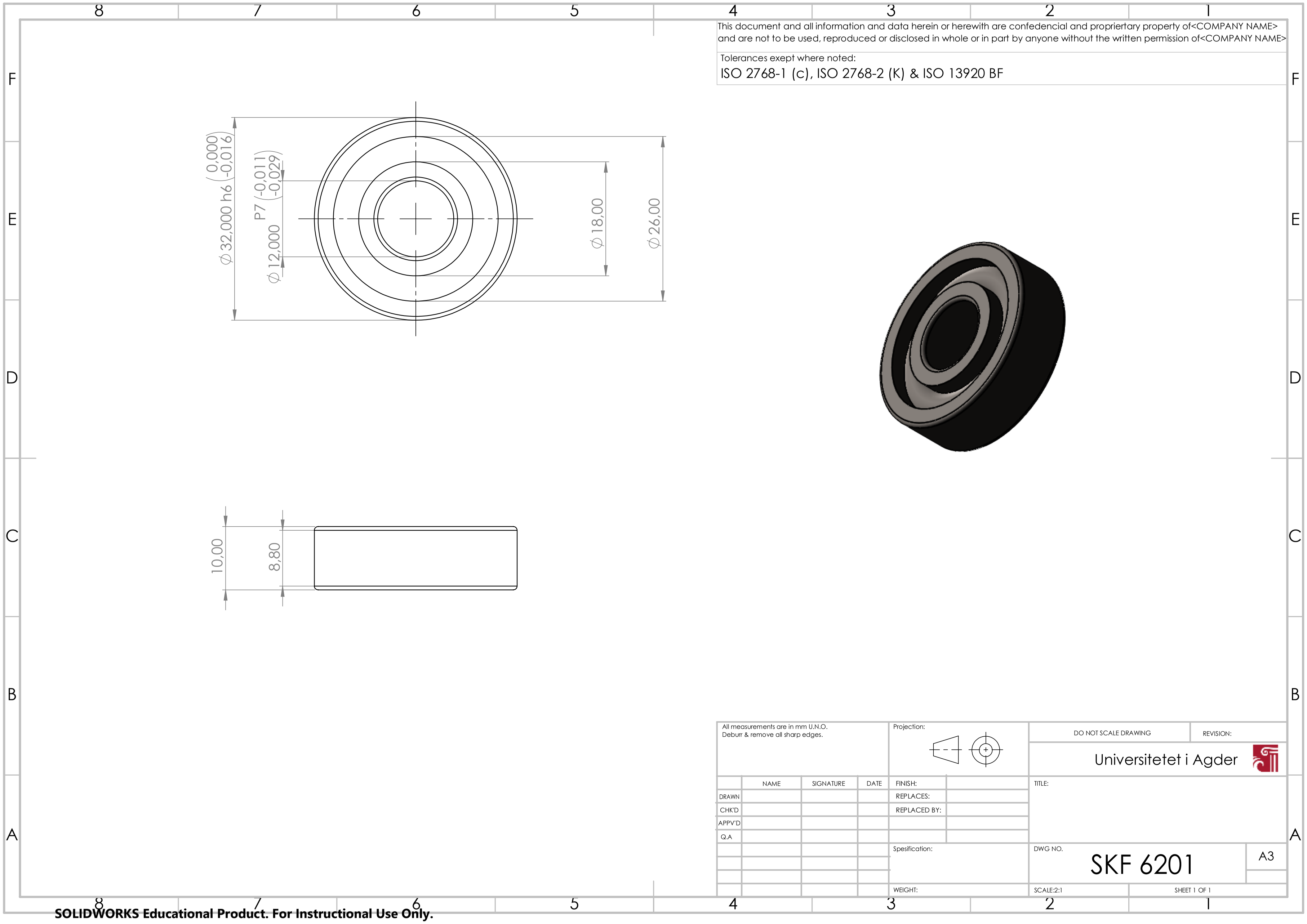
DWG NO.
003 - P06

A3

WEIGHT:

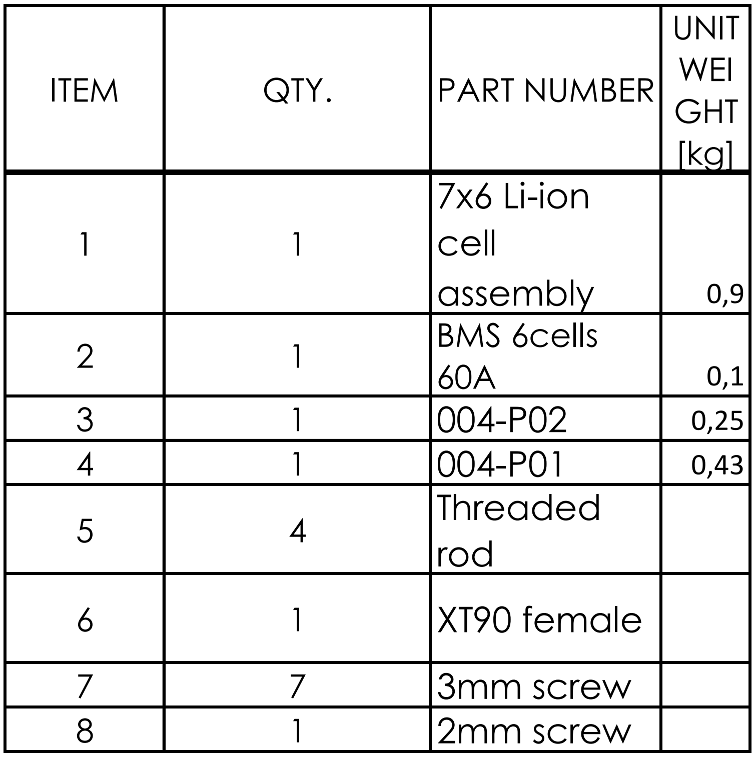
SCALE:1:1

SHEET 1 OF 1


8
7
6
5
4
3
2
1
F
E
D
C
B
A
This document and all information and data herein or herewith are confedencial and proprierary property of<COMPANY NAME>
and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>
Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF
∅32,000 h6 (0,000 -0,016)
∅12,000 P7 (-0,011 -0,029)
∅18,00
∅26,00
10,00
8,80
All measurements are in mm U.N.O.
Deburr & remove all sharp edges.
Projection:
DO NOT SCALE DRAWING
REVISION:
Universitetet i Agder
NAME
SIGNATURE
DATE
FINISH:
REPLACES:
REPLACED BY:
TITLE:
DRAWN
CHK'D
APPV'D
Q.A
Spesification:
DWG NO.
SKF 6201
A3
WEIGHT:
SCALE:2:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.

A.5 004-000: Battery Package

Bill of materials for the battery package assembly:

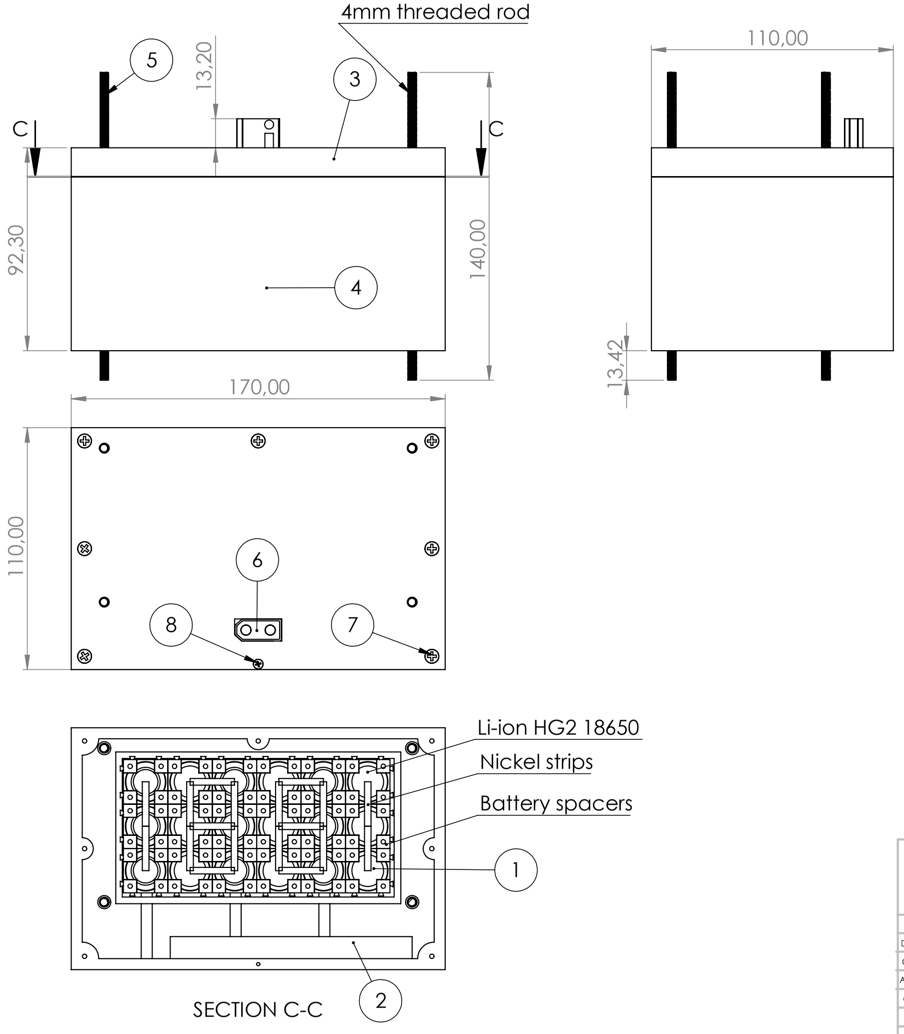


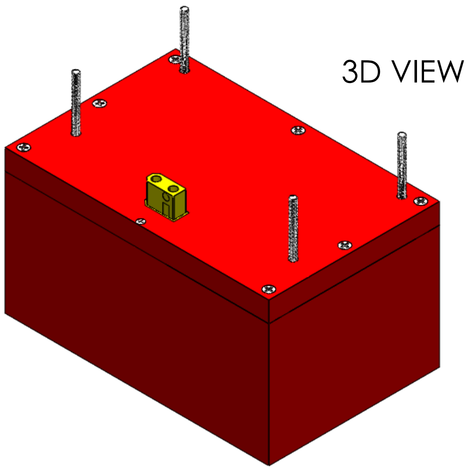
ITEM	QTY.	PART NUMBER	UNIT WEIGHT [kg]
1	1	7x6 Li-ion cell assembly	0,9
2	1	BMS 6cells 60A	0,1
3	1	004-P02	0,25
4	1	004-P01	0,43
5	4	Threaded rod	
6	1	XT90 female	
7	7	3mm screw	
8	1	2mm screw	

Next page will include the assembly of the total system, and after that the subassembly-drawings will be presented.

This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF




3D VIEW

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	7x6 Li-ion cell assembly		1
2	BMS 6cells 60A		1
3	004-P02		1
4	004-P01		1
5	Threaded rod		4
6	XT90 female		1
7	3mm screw		7
8	2mm screw		1

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

DWG NO. 004-000

A3

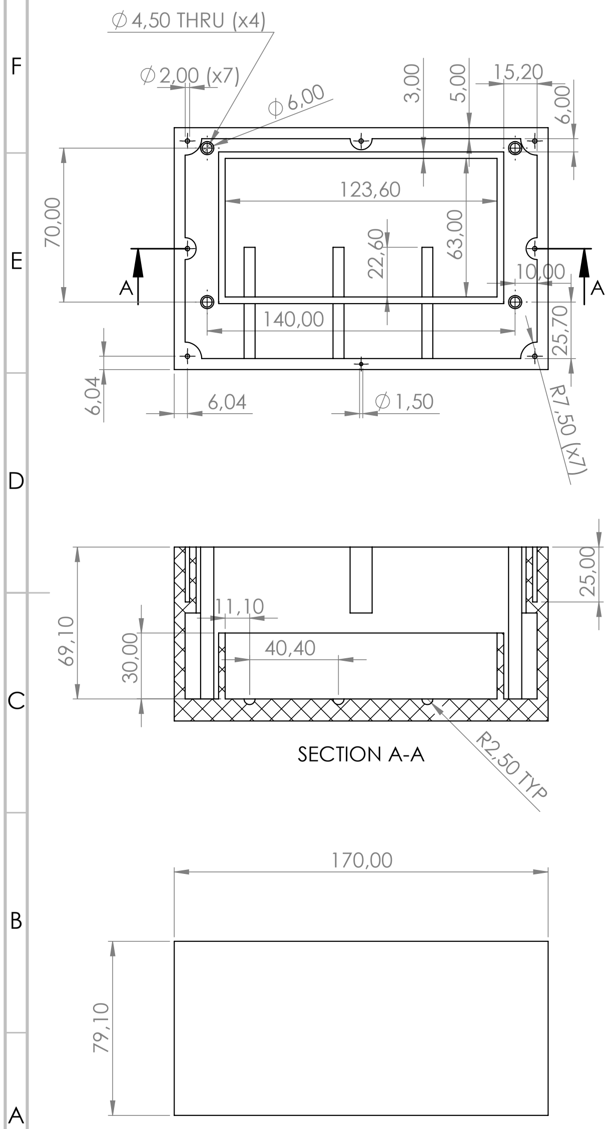
WEIGHT: 1.66679

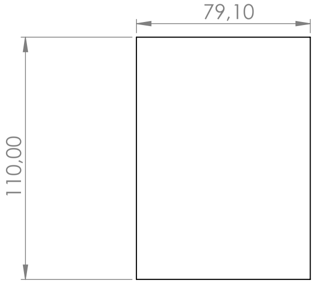
SCALE:1:2

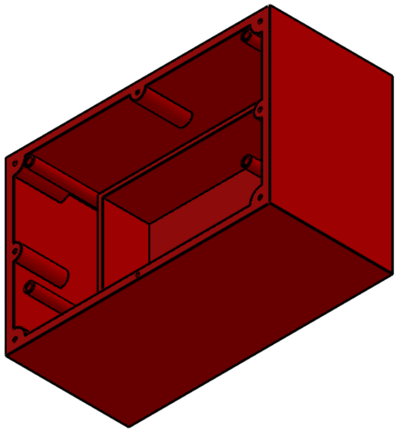
SHEET 1 OF 1

This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF







ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Bottom cover		1

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

DWG NO. 004-P01 A3

WEIGHT: 0.42604

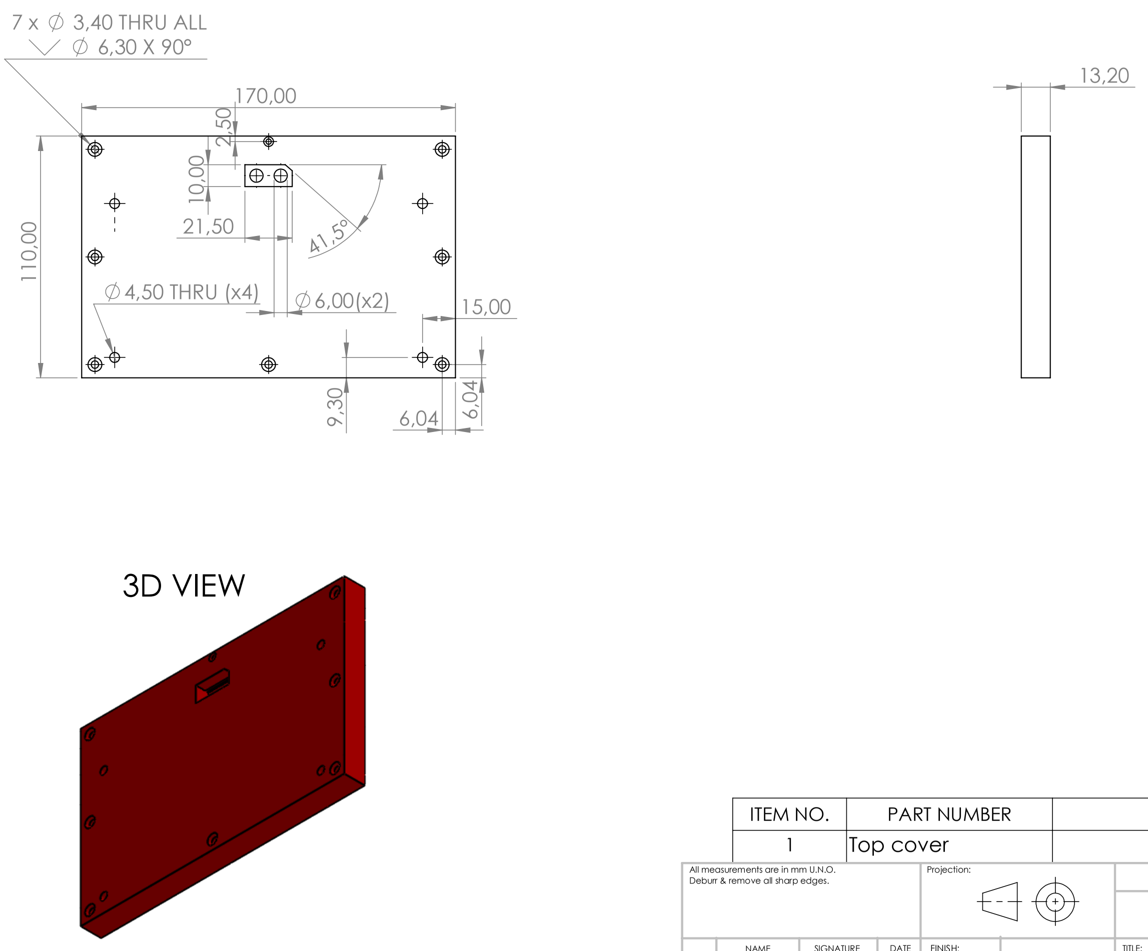
SCALE:1:2

SHEET 1 OF 1

SOLIDWORKS Educational Product. For Instructional Use Only.

This document and all information and data herein or herewith are confedencial and proprietary property of<COMPANY NAME> and are not to be used, reproduced or disclosed in whole or in part by anyone without the written permission of<COMPANY NAME>

Tolerances exept where noted:
ISO 2768-1 (c), ISO 2768-2 (K) & ISO 13920 BF



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Top cover		1

All measurements are in mm U.N.O.
Deburr & remove all sharp edges.

Projection:

DO NOT SCALE DRAWING

REVISION:

Universitetet i Agder

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
Q.A			

FINISH:
REPLACES:
REPLACED BY:

TITLE:

Spesification:

DWG NO. 004-P02

A3

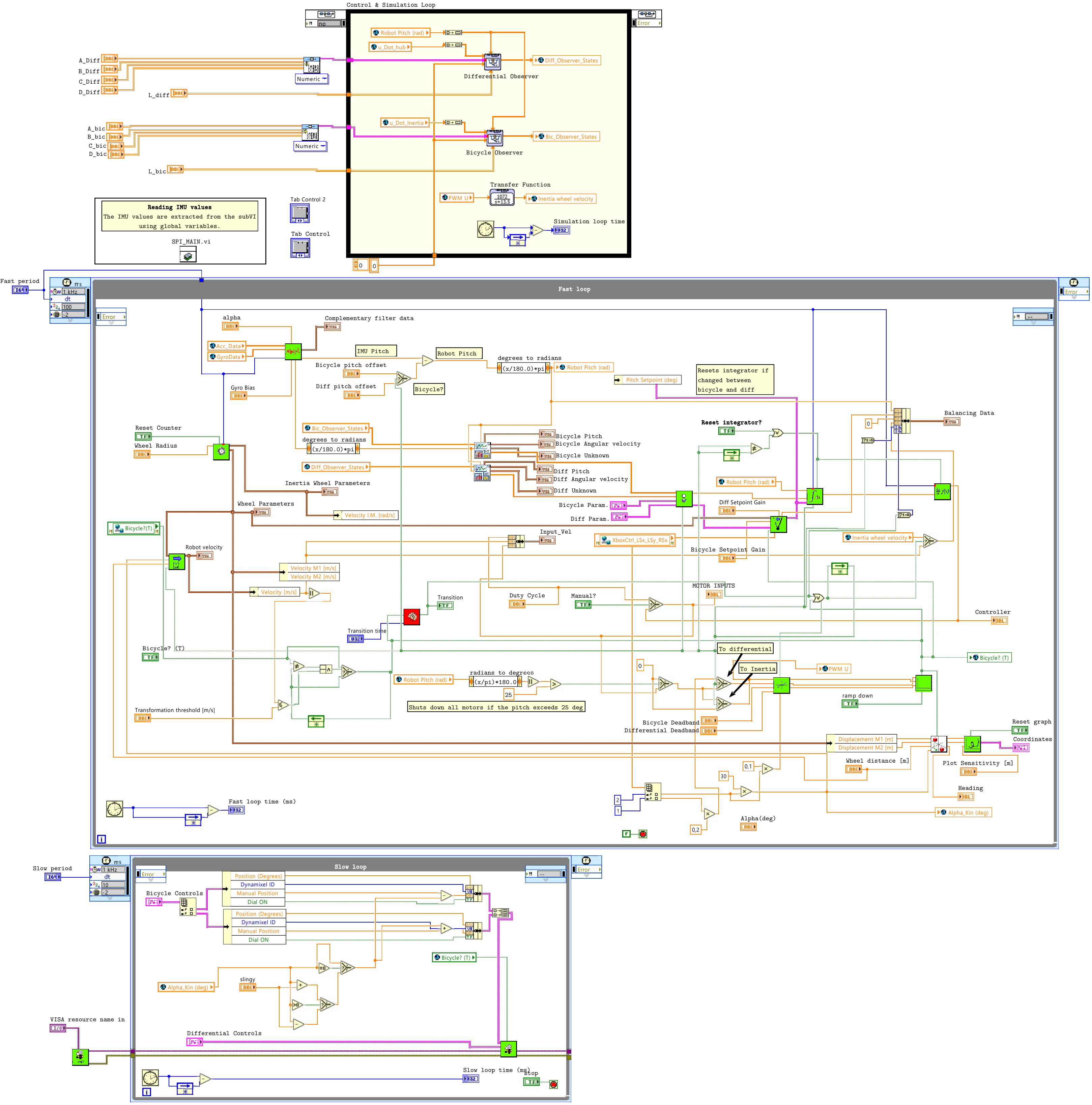
WEIGHT: 0.24569

SCALE:1:2

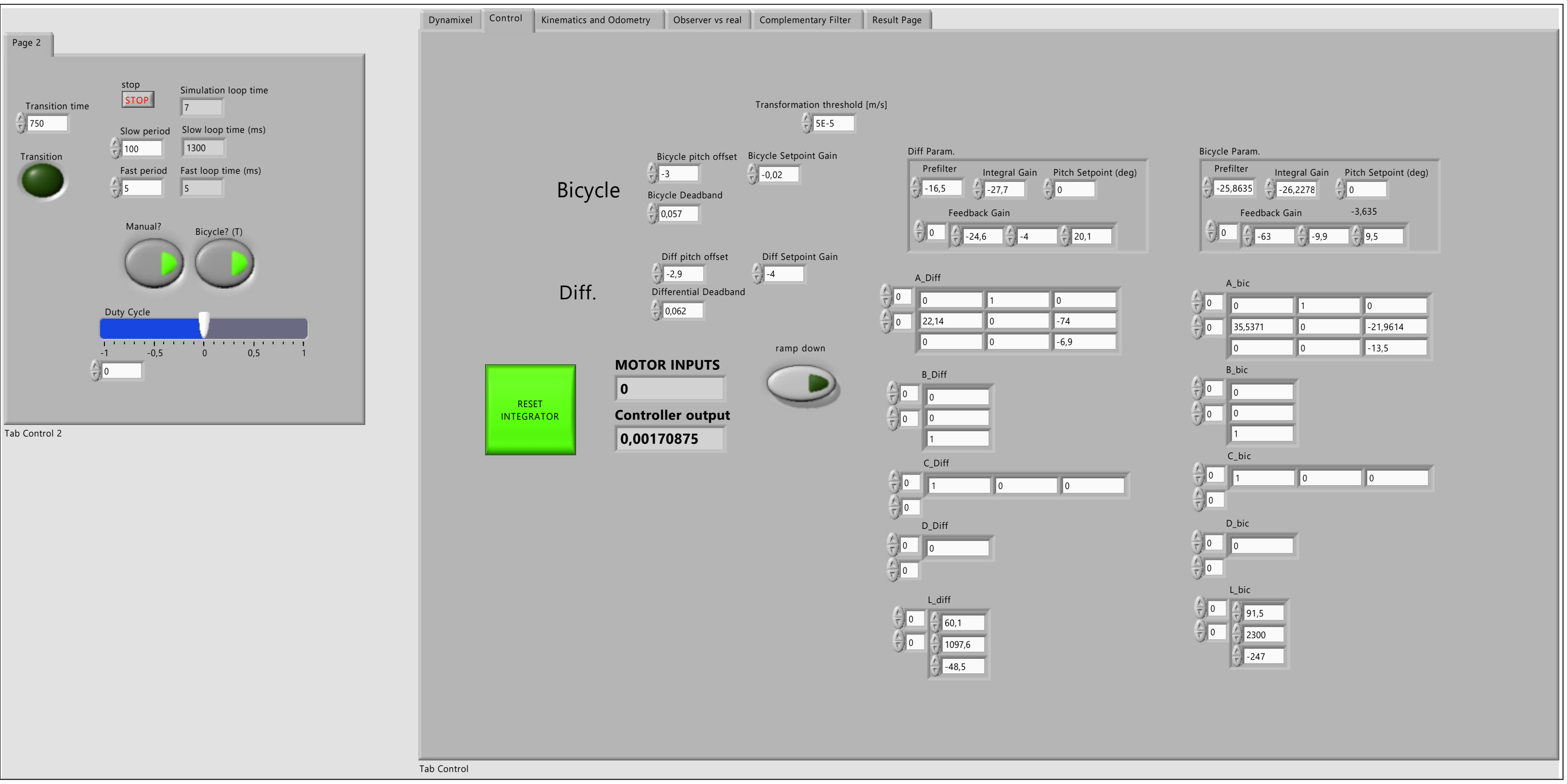
SHEET 1 OF 1

B LabVIEW Code

B.1 Block Diagram


Control & Simulation Loop
Robot Pitch (rad)
u_Dot_hub
A_Diff
B_Diff
C_Diff
D_Diff
Numeric
L_diff
Diff_Observer_States
Differential Observer
u_Dot_Inertia
A_bic
B_bic
C_bic
D_bic
Numeric
L_bic
Bic_Observer_States
Bicycle Observer
Transfer Function
PWM U
1072
s+13.5
Inertia wheel velocity
Simulation loop time
Reading IMU values
The IMU values are extracted from the subVI using global variables.
SPI_MAIN.vi
Tab Control 2
Tab Control
Error
Fast period
1 kHz
dt
100
-2
Fast loop
alpha
Complementary filter data
Acc_Data
GyroData
IMU Pitch
Robot Pitch
degrees to radians
(x/180.0)*pi
Robot Pitch (rad)
Bicycle pitch offset
Diff pitch offset
Bicycle?
Pitch Setpoint (deg)
Resets integrator if changed between bicycle and diff
Gyro Bias
Balancing Data
Reset integrator?
Reset Counter
Wheel Radius
Bic_Observer_States
degrees to radians
(x/180.0)*pi
Bicycle Pitch
Bicycle Angular velocity
Bicycle Unknown
Diff_Observer_States
Diff Pitch
Diff Angular velocity
Diff Unknown
Inertia Wheel Parameters
Velocity I.M. [rad/s]
Wheel Parameters
Bicycle Param.
Diff Param.
Diff Setpoint Gain
Robot Pitch (rad)
Bicycle?(T)
Robot velocity
Input_Vel
XboxCtrl_LSx_LSy_RSx
Inertia wheel velocity
Bicycle Setpoint Gain
Velocity M1 [m/s]
Velocity M2 [m/s]
Velocity [m/s]
Transition
Duty Cycle
Manual?
MOTOR INPUTS
Controller
Transition time
Bicycle? (T)
To differential
To Inertia
PWM U
Bicycle? (T)
Robot Pitch (rad)
radians to degrees
(x/pi)*180.0
25
Shuts down all motors if the pitch exceeds 25 deg
ramp down
Transformation threshold [m/s]
Bicycle Deadband
Differential Deadband
Reset graph
Displacement M1 [m]
Displacement M2 [m]
Coordinates
Wheel distance [m]
Plot Sensitivity [m]
0,1
30
Heading
2
1
Alpha_Kin (deg)
0,2
Alpha(deg)
Fast loop time (ms)
Slow period
1 kHz
dt
10
-2
Slow loop
Error
Position (Degrees)
Dynamixel ID
Manual Position
Dial ON
Bicycle Controls
Position (Degrees)
Dynamixel ID
Manual Position
Dial ON
Bicycle? (T)
Alpha_Kin (deg)
slingy
VISA resource name in
Differential Controls
Slow loop time (ms)
Stop

B.2 Front Panel



Transition time: 750

stop

Simulation loop time: 7

Slow period: 100

Slow loop time (ms): 1300

Transition

Fast period: 5

Fast loop time (ms): 5

Manual?

Bicycle? (T)

Duty Cycle: 0

Transformation threshold [m/s]: 5E-5

Bicycle

Bicycle pitch offset: -3

Bicycle Setpoint Gain: -0,02

Bicycle Deadband: 0,057

Diff.

Diff pitch offset: -2,9

Diff Setpoint Gain: -4

Differential Deadband: 0,062

MOTOR INPUTS: 0

Controller output: 0,00170875

ramp down

Diff Param.

Prefilter: -16,5

Integral Gain: -27,7

Pitch Setpoint (deg): 0

Feedback Gain: 0 | -24,6 | -4 | 20,1

Bicycle Param.

Prefilter: -25,8635

Integral Gain: -26,2278

Pitch Setpoint (deg): 0

Feedback Gain: 0 | -63 | -9,9 | 9,5

-3,635

A_Diff (0, 0)

0	1	0
22,14	0	-74
0	0	-6,9

B_Diff (0, 0)

0
0
1

C_Diff (0, 0)

1	0	0

D_Diff (0, 0)

0

L_diff (0, 0)

60,1
1097,6
-48,5

A_bic (0, 0)

0	1	0
35,5371	0	-21,9614
0	0	-13,5

B_bic (0, 0)

0
0
1

C_bic (0, 0)

1	0	0

D_bic (0, 0)

0

L_bic (0, 0)

91,5
2300
-247

C MATLAB Scripts

C.1 Bicycle Tuning

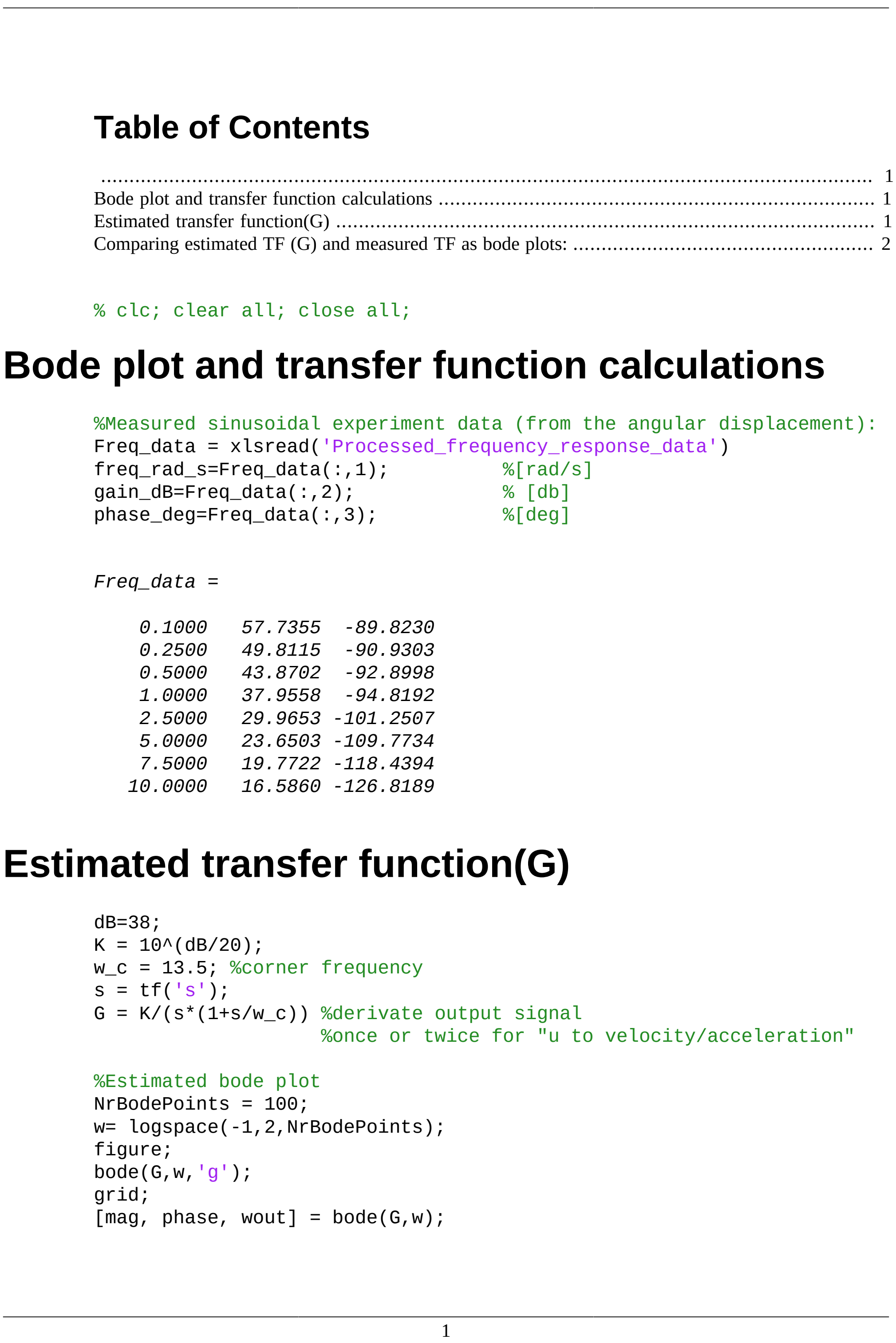


Table of Contents

```
% clc; clear all; close all;
```

Bode plot and transfer function calculations

```
%Measured sinusoidal experiment data (from the angular displacement):
Freq_data = xlsread('Processed_frequency_response_data')
freq_rad_s=Freq_data(:,1);          %[rad/s]
gain_dB=Freq_data(:,2);             % [db]
phase_deg=Freq_data(:,3);           %[deg]
```

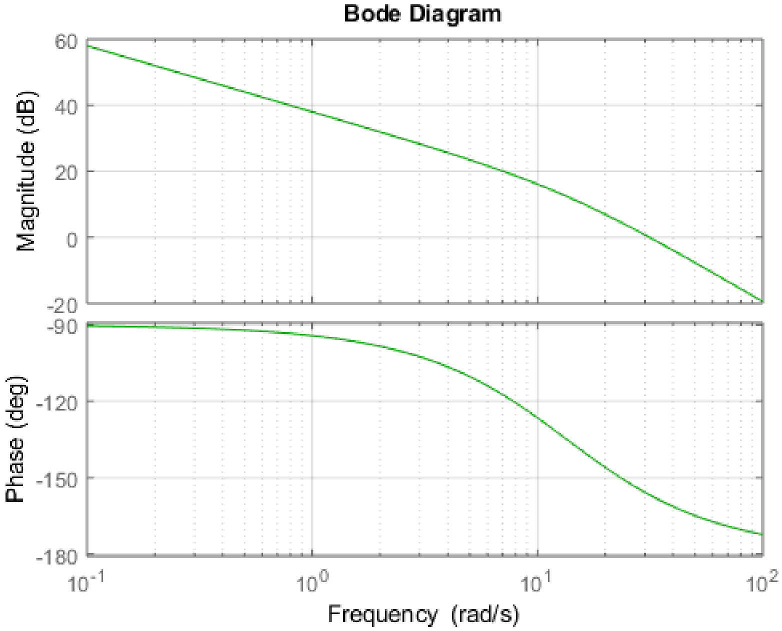
```
Freq_data =

    0.1000   57.7355  -89.8230
    0.2500   49.8115  -90.9303
    0.5000   43.8702  -92.8998
    1.0000   37.9558  -94.8192
    2.5000   29.9653 -101.2507
    5.0000   23.6503 -109.7734
    7.5000   19.7722 -118.4394
   10.0000   16.5860 -126.8189
```

Estimated transfer function(G)

```
dB=38;
K = 10^(dB/20);
w_c = 13.5; %corner frequency
s = tf('s');
G = K/(s*(1+s/w_c)) %derivate output signal
                    %once or twice for "u to velocity/acceleration"

%Estimated bode plot
NrBodePoints = 100;
w= logspace(-1,2,NrBodePoints);
figure;
bode(G,w,'g');
grid;
[mag, phase, wout] = bode(G,w);
```

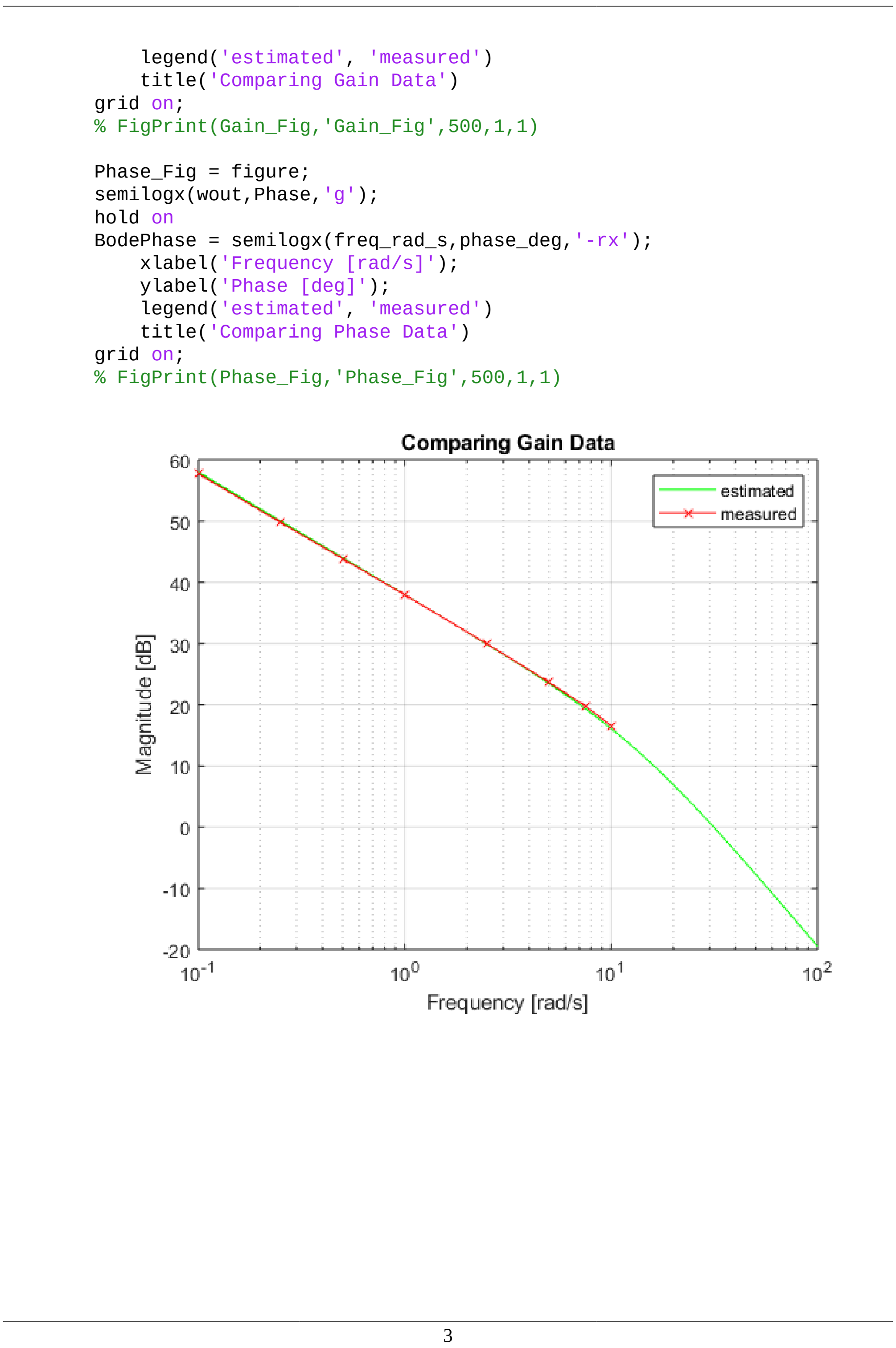
```
%reshape bode plot values
Phase = reshape(phase,[NrBodePoints,1]);
Magnitude_gain = reshape(mag,[NrBodePoints,1]);
Magnitude_dB = 20*log10(Magnitude_gain);
```

```
G =

      1072
  ------------
  s^2 + 13.5 s

Continuous-time transfer function.
```



Comparing estimated TF (G) and measured TF as bode plots:

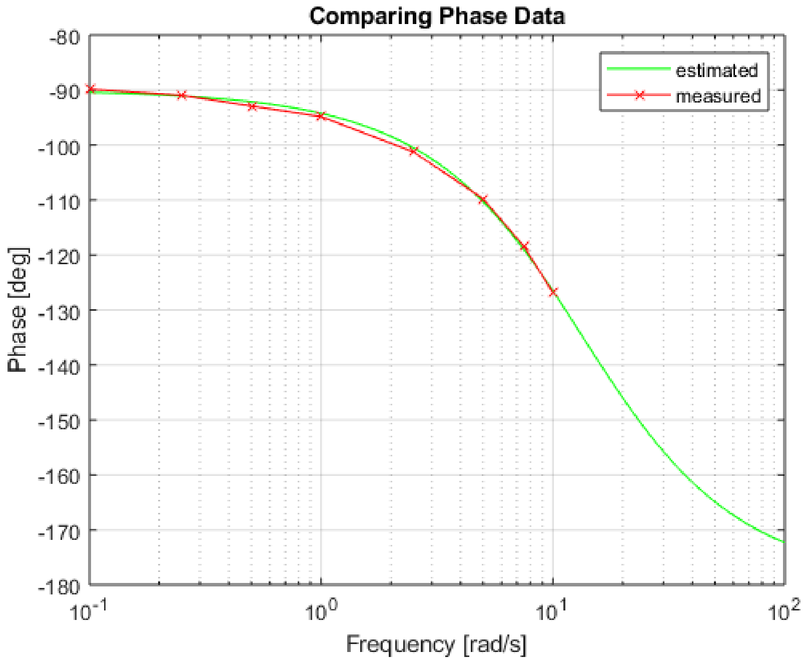
```
Gain_Fig = figure;
semilogx(wout,Magnitude_dB,'g');
hold on
BodeGain = semilogx(freq_rad_s,gain_dB,'-rx'); %gjør om til log. x
 akse.
    xlabel('Frequency [rad/s]');
    ylabel('Magnitude [dB]');
```



```
    legend('estimated', 'measured')
    title('Comparing Gain Data')
grid on;
% FigPrint(Gain_Fig,'Gain_Fig',500,1,1)

Phase_Fig = figure;
semilogx(wout,Phase,'g');
hold on
BodePhase = semilogx(freq_rad_s,phase_deg,'-rx');
    xlabel('Frequency [rad/s]');
    ylabel('Phase [deg]');
    legend('estimated', 'measured')
    title('Comparing Phase Data')
grid on;
% FigPrint(Phase_Fig,'Phase_Fig',500,1,1)
```

Comparing Gain Data
60
50
40
30
20
10
0
-10
-20
Magnitude [dB]
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency [rad/s]
estimated
measured

3



Published with MATLAB® R2018a

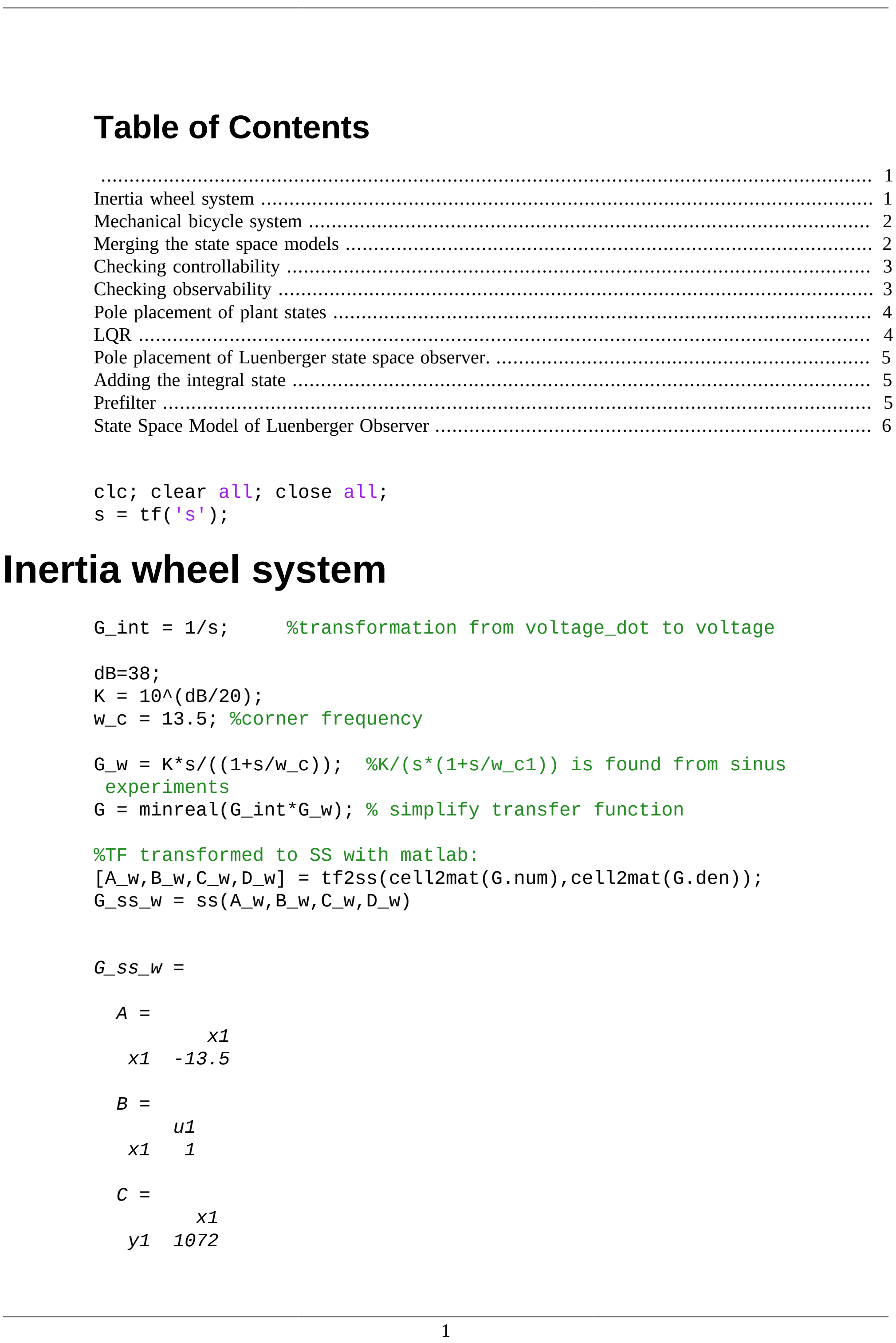


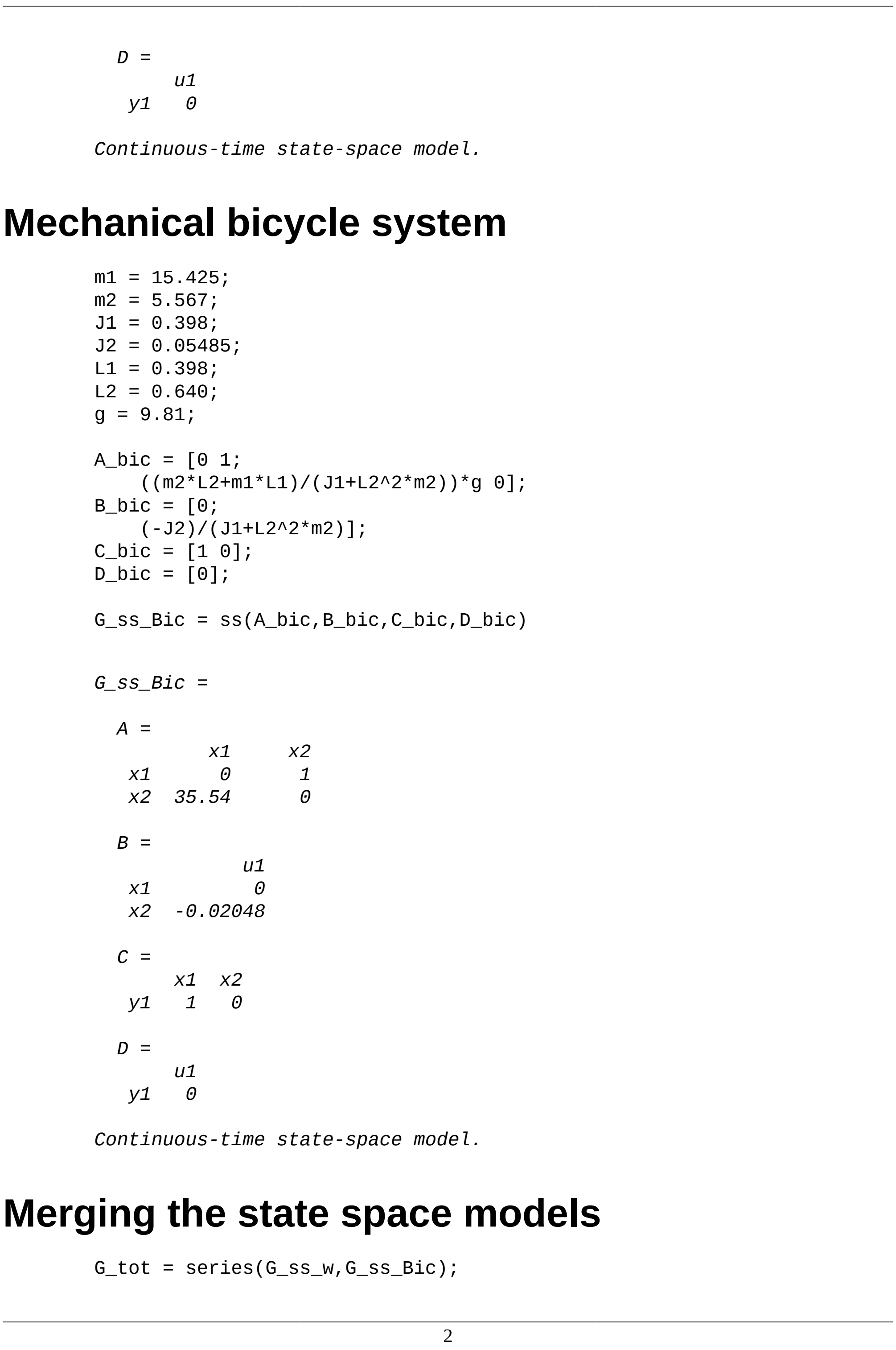
Table of Contents

```
clc; clear all; close all;
s = tf('s');
```

Inertia wheel system

```
G_int = 1/s;     %transformation from voltage_dot to voltage

dB=38;
K = 10^(dB/20);
w_c = 13.5; %corner frequency

G_w = K*s/((1+s/w_c));  %K/(s*(1+s/w_c1)) is found from sinus
 experiments
G = minreal(G_int*G_w); % simplify transfer function

%TF transformed to SS with matlab:
[A_w,B_w,C_w,D_w] = tf2ss(cell2mat(G.num),cell2mat(G.den));
G_ss_w = ss(A_w,B_w,C_w,D_w)
```

```
G_ss_w =

  A =
          x1
   x1  -13.5

  B =
       u1
   x1   1

  C =
         x1
   y1  1072
```



```
  D = 
       u1
   y1   0
 
Continuous-time state-space model.
```

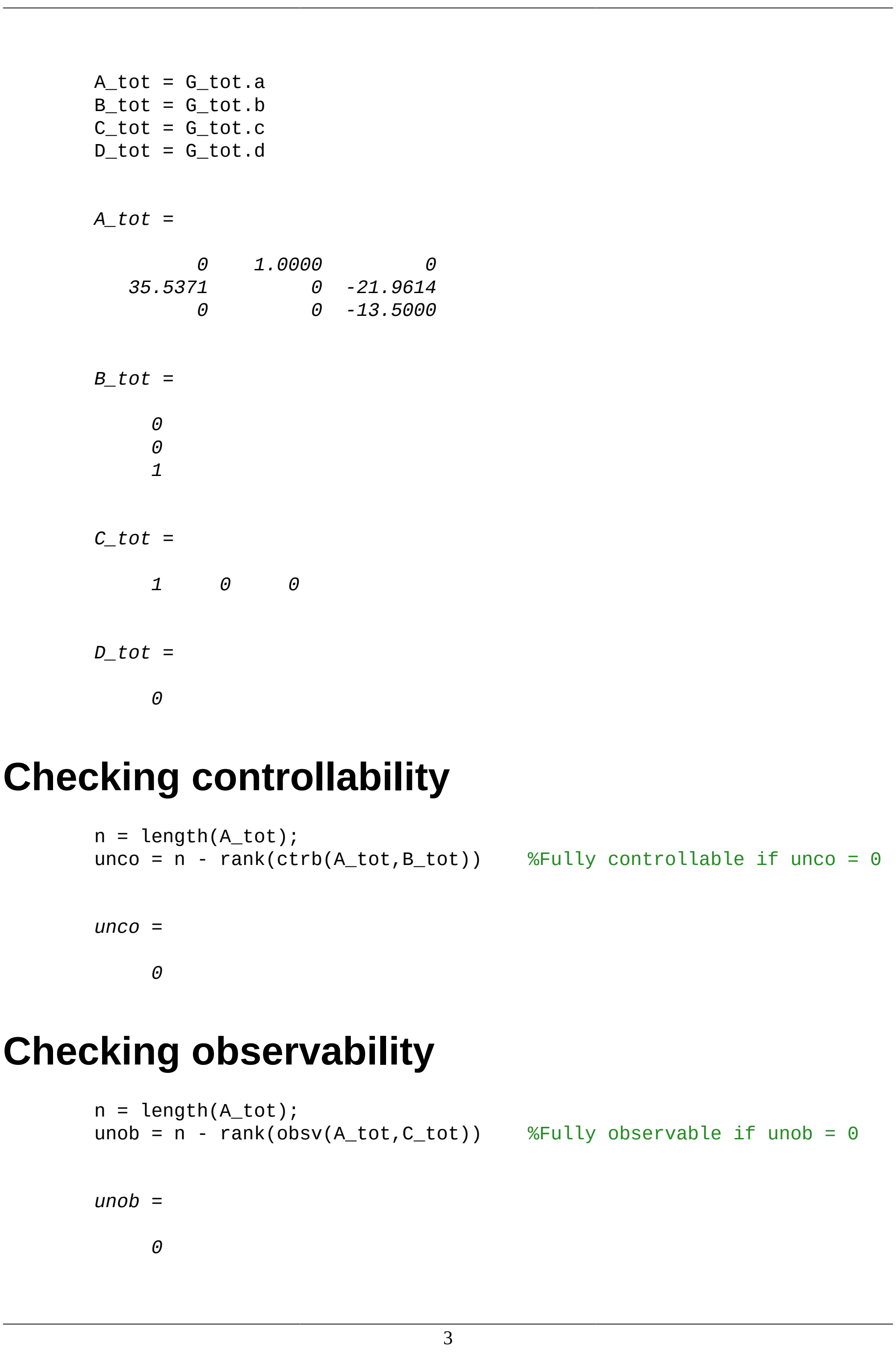
Mechanical bicycle system

```
m1 = 15.425;
m2 = 5.567;
J1 = 0.398;
J2 = 0.05485;
L1 = 0.398;
L2 = 0.640;
g = 9.81;

A_bic = [0 1;
    ((m2*L2+m1*L1)/(J1+L2^2*m2))*g 0];
B_bic = [0;
    (-J2)/(J1+L2^2*m2)];
C_bic = [1 0];
D_bic = [0];

G_ss_Bic = ss(A_bic,B_bic,C_bic,D_bic)
```

```

G_ss_Bic =
 
  A = 
          x1     x2
   x1      0      1
   x2  35.54      0
 
  B = 
             u1
   x1         0
   x2  -0.02048
 
  C = 
       x1  x2
   y1   1   0
 
  D = 
       u1
   y1   0
 
Continuous-time state-space model.
```

Merging the state space models

```
G_tot = series(G_ss_w,G_ss_Bic);
```



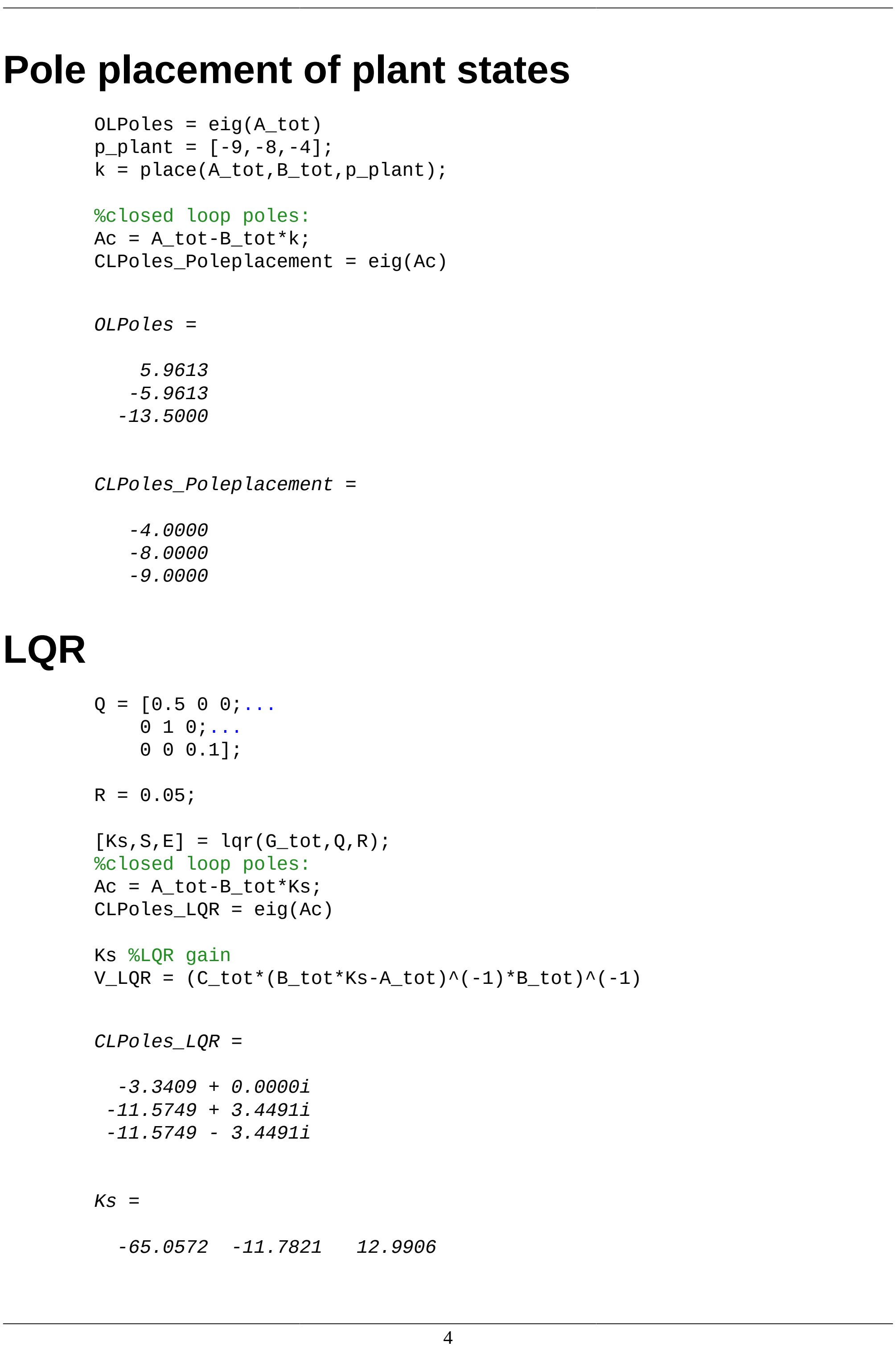
```
A_tot = G_tot.a
B_tot = G_tot.b
C_tot = G_tot.c
D_tot = G_tot.d


A_tot =

         0    1.0000         0
   35.5371         0  -21.9614
         0         0  -13.5000


B_tot =

     0
     0
     1


C_tot =

     1     0     0


D_tot =

     0
```

Checking controllability

```
n = length(A_tot);
unco = n - rank(ctrb(A_tot,B_tot))    %Fully controllable if unco = 0


unco =

     0
```

Checking observability

```
n = length(A_tot);
unob = n - rank(obsv(A_tot,C_tot))    %Fully observable if unob = 0


unob =

     0
```

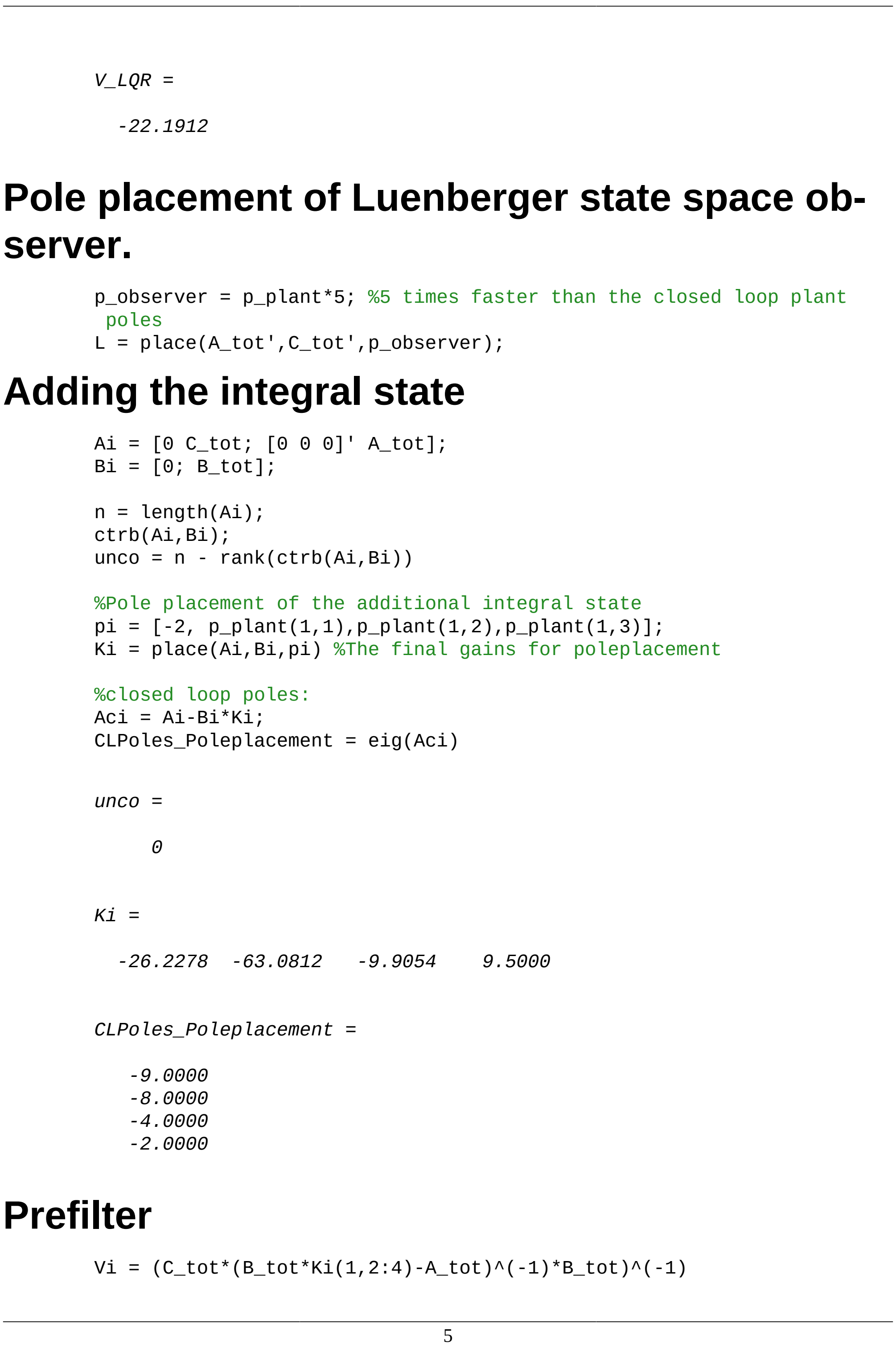


Pole placement of plant states

```
OLPoles = eig(A_tot)
p_plant = [-9,-8,-4];
k = place(A_tot,B_tot,p_plant);

%closed loop poles:
Ac = A_tot-B_tot*k;
CLPoles_Poleplacement = eig(Ac)


OLPoles =

    5.9613
   -5.9613
  -13.5000


CLPoles_Poleplacement =

   -4.0000
   -8.0000
   -9.0000
```

LQR

```
Q = [0.5 0 0;...
    0 1 0;...
    0 0 0.1];

R = 0.05;

[Ks,S,E] = lqr(G_tot,Q,R);
%closed loop poles:
Ac = A_tot-B_tot*Ks;
CLPoles_LQR = eig(Ac)

Ks %LQR gain
V_LQR = (C_tot*(B_tot*Ks-A_tot)^(-1)*B_tot)^(-1)


CLPoles_LQR =

  -3.3409 + 0.0000i
 -11.5749 + 3.4491i
 -11.5749 - 3.4491i


Ks =

  -65.0572  -11.7821   12.9906
```



```
V_LQR =

  -22.1912
```

Pole placement of Luenberger state space observer.

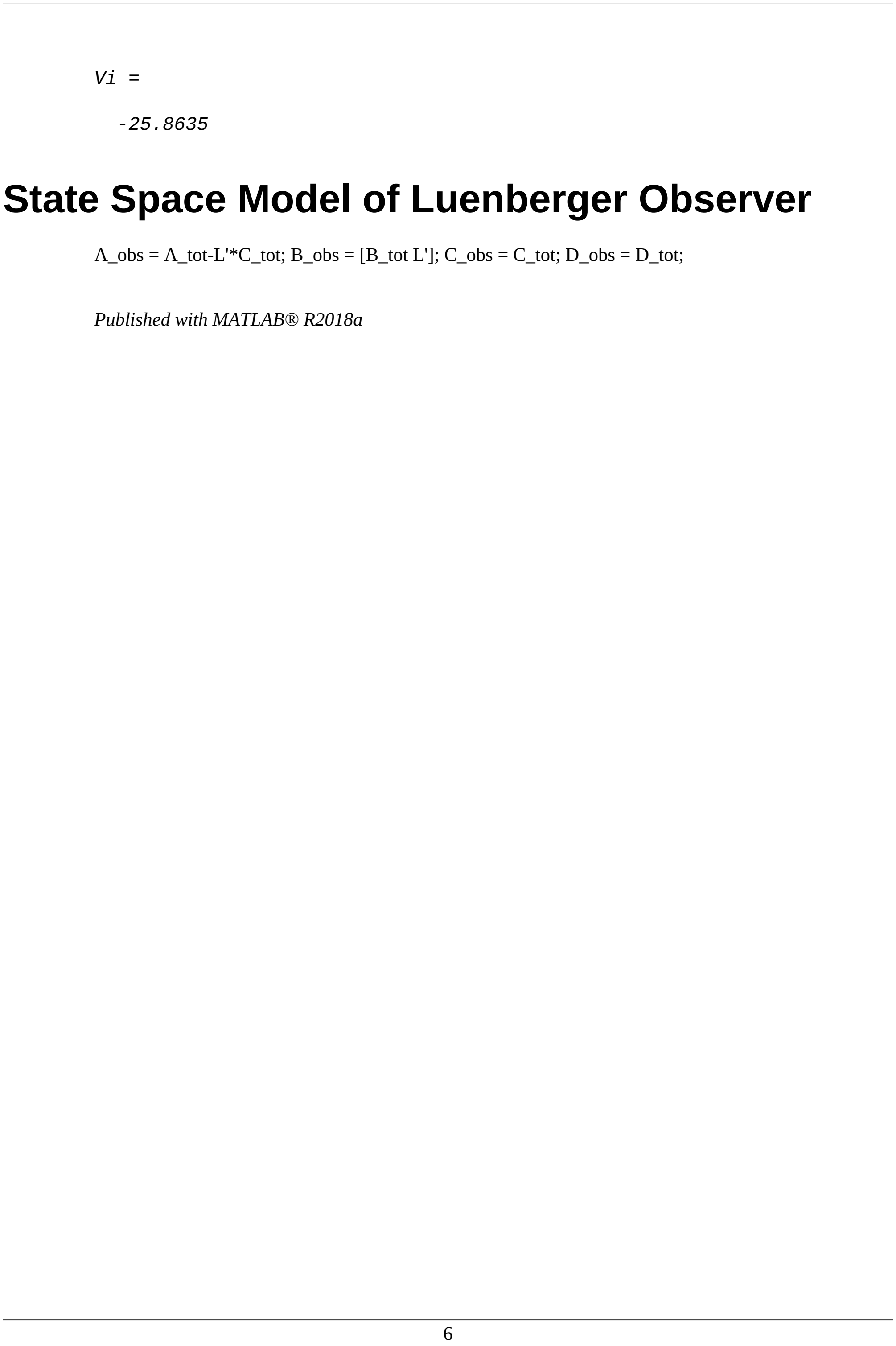
```
p_observer = p_plant*5; %5 times faster than the closed loop plant
 poles
L = place(A_tot',C_tot',p_observer);
```

Adding the integral state

```
Ai = [0 C_tot; [0 0 0]' A_tot];
Bi = [0; B_tot];

n = length(Ai);
ctrb(Ai,Bi);
unco = n - rank(ctrb(Ai,Bi))

%Pole placement of the additional integral state
pi = [-2, p_plant(1,1),p_plant(1,2),p_plant(1,3)];
Ki = place(Ai,Bi,pi) %The final gains for poleplacement

%closed loop poles:
Aci = Ai-Bi*Ki;
CLPoles_Poleplacement = eig(Aci)


unco =

     0


Ki =

  -26.2278  -63.0812   -9.9054    9.5000


CLPoles_Poleplacement =

   -9.0000
   -8.0000
   -4.0000
   -2.0000
```

Prefilter

```
Vi = (C_tot*(B_tot*Ki(1,2:4)-A_tot)^(-1)*B_tot)^(-1)
```



```
Vi =

  -25.8635
```

State Space Model of Luenberger Observer

```
A_obs = A_tot-L'*C_tot; B_obs = [B_tot L']; C_obs = C_tot; D_obs = D_tot;
```

Published with MATLAB® R2018a

C.2 Differential Drive Tuning

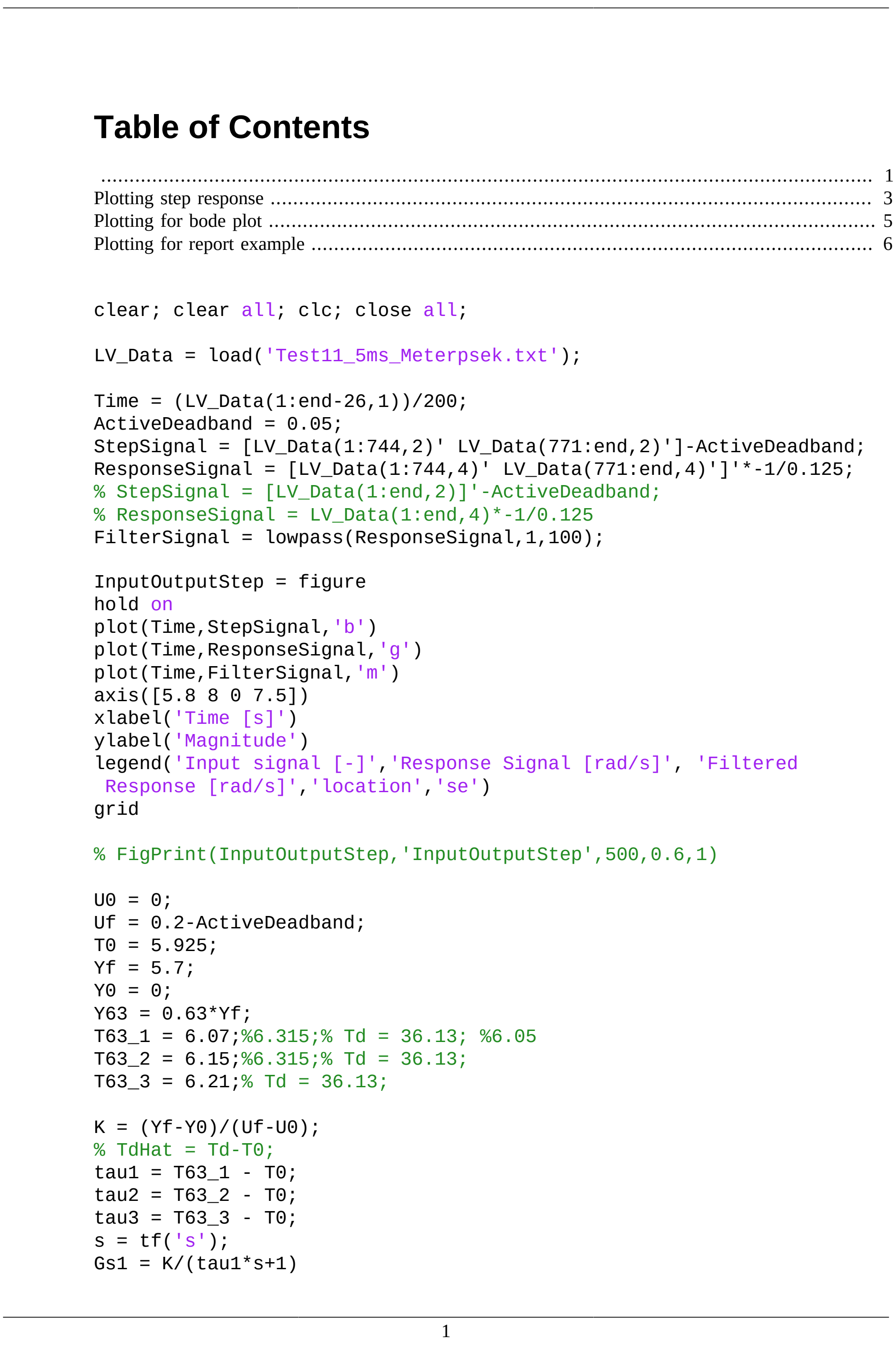
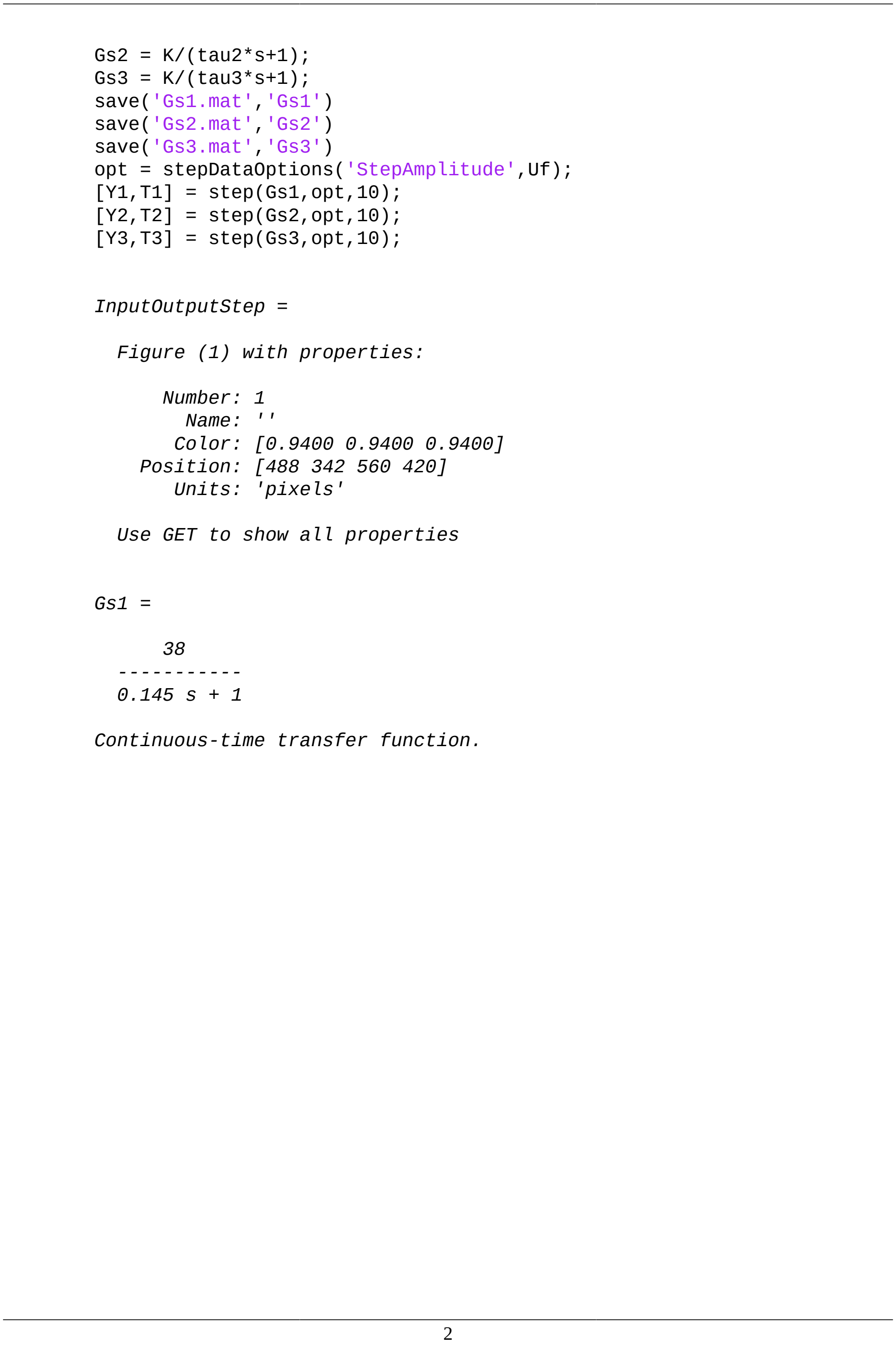


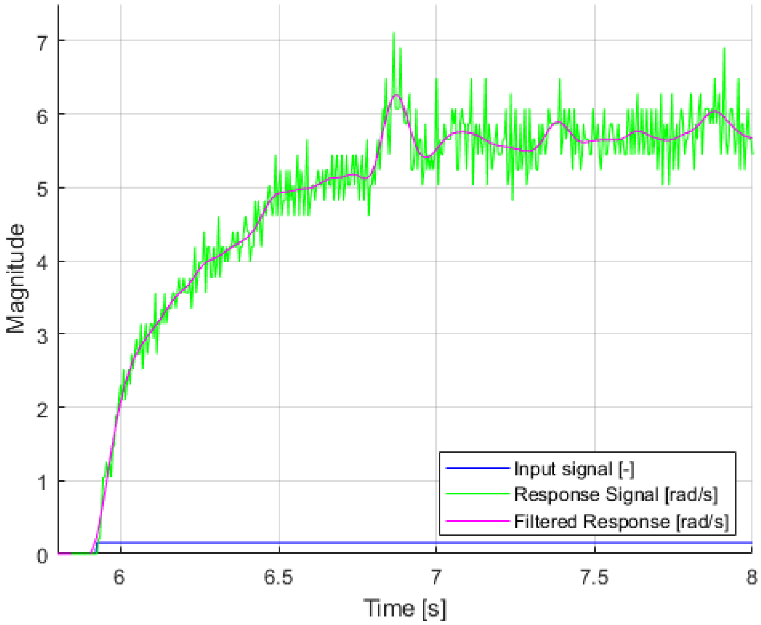
Table of Contents

```
clear; clear all; clc; close all;

LV_Data = load('Test11_5ms_Meterpsek.txt');

Time = (LV_Data(1:end-26,1))/200;
ActiveDeadband = 0.05;
StepSignal = [LV_Data(1:744,2)' LV_Data(771:end,2)']-ActiveDeadband;
ResponseSignal = [LV_Data(1:744,4)' LV_Data(771:end,4)']'*-1/0.125;
% StepSignal = [LV_Data(1:end,2)]'-ActiveDeadband;
% ResponseSignal = LV_Data(1:end,4)*-1/0.125
FilterSignal = lowpass(ResponseSignal,1,100);

InputOutputStep = figure
hold on
plot(Time,StepSignal,'b')
plot(Time,ResponseSignal,'g')
plot(Time,FilterSignal,'m')
axis([5.8 8 0 7.5])
xlabel('Time [s]')
ylabel('Magnitude')
legend('Input signal [-]','Response Signal [rad/s]', 'Filtered
 Response [rad/s]','location','se')
grid

% FigPrint(InputOutputStep,'InputOutputStep',500,0.6,1)

U0 = 0;
Uf = 0.2-ActiveDeadband;
T0 = 5.925;
Yf = 5.7;
Y0 = 0;
Y63 = 0.63*Yf;
T63_1 = 6.07;%6.315;% Td = 36.13; %6.05
T63_2 = 6.15;%6.315;% Td = 36.13;
T63_3 = 6.21;% Td = 36.13;

K = (Yf-Y0)/(Uf-U0);
% TdHat = Td-T0;
tau1 = T63_1 - T0;
tau2 = T63_2 - T0;
tau3 = T63_3 - T0;
s = tf('s');
Gs1 = K/(tau1*s+1)
```



```
Gs2 = K/(tau2*s+1);
Gs3 = K/(tau3*s+1);
save('Gs1.mat','Gs1')
save('Gs2.mat','Gs2')
save('Gs3.mat','Gs3')
opt = stepDataOptions('StepAmplitude',Uf);
[Y1,T1] = step(Gs1,opt,10);
[Y2,T2] = step(Gs2,opt,10);
[Y3,T3] = step(Gs3,opt,10);
```

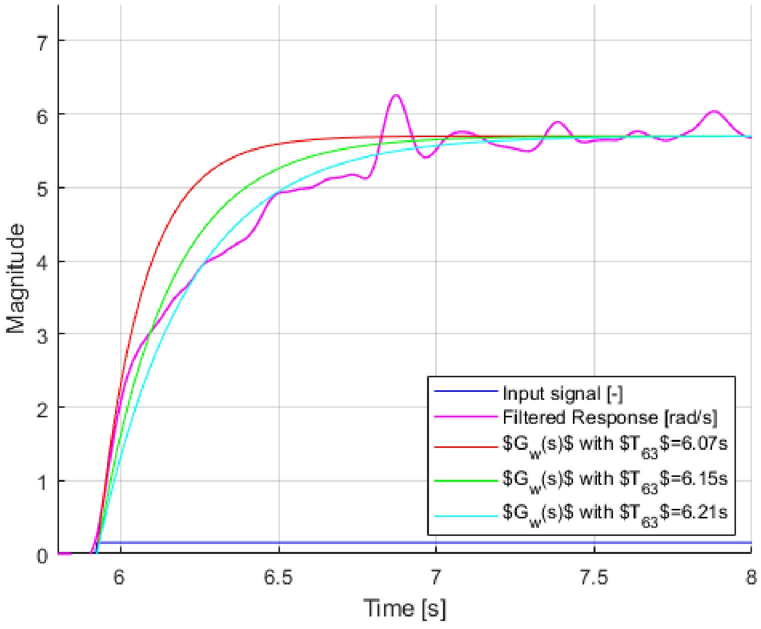
```
InputOutputStep =

  Figure (1) with properties:

      Number: 1
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [488 342 560 420]
       Units: 'pixels'

  Use GET to show all properties


Gs1 =

      38
  -----------
  0.145 s + 1

Continuous-time transfer function.
```

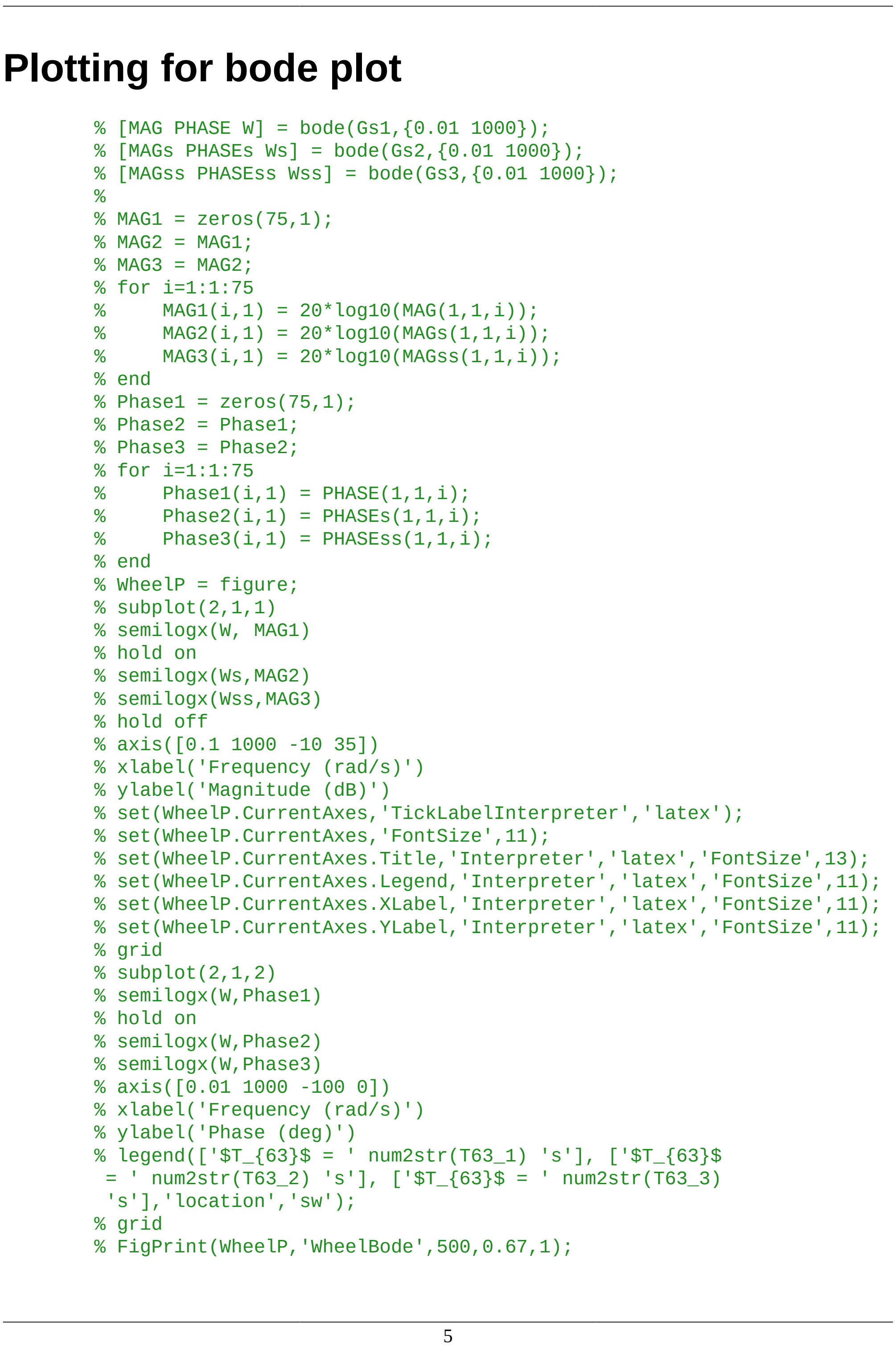


Plotting step response

```
Cake = figure
hold on
plot(Time,StepSignal,'b')
% plot(Time,ResponseSignal)
g =plot(Time,FilterSignal,'m')
h =plot(T1+T0,Y1,'r');
i =plot(T2+T0,Y2,'g');
j =plot(T3+T0,Y3,'c');
set([g],'linewidth',1)
axis([5.8 8 0 7.5])
xlabel('Time [s]')
ylabel('Magnitude')
legend('Input signal [-]', 'Filtered Response [rad/s]',
['$G_w(s)$ with $T_{63}$=' num2str(T63_1) 's'],['$G_w(s)$ with
 $T_{63}$=' num2str(T63_2) 's'],['$G_w(s)$ with $T_{63}$='
 num2str(T63_3) 's'],'location','se')
grid

% FigPrint(Cake,'StepFunction',500,0.65,1)
```

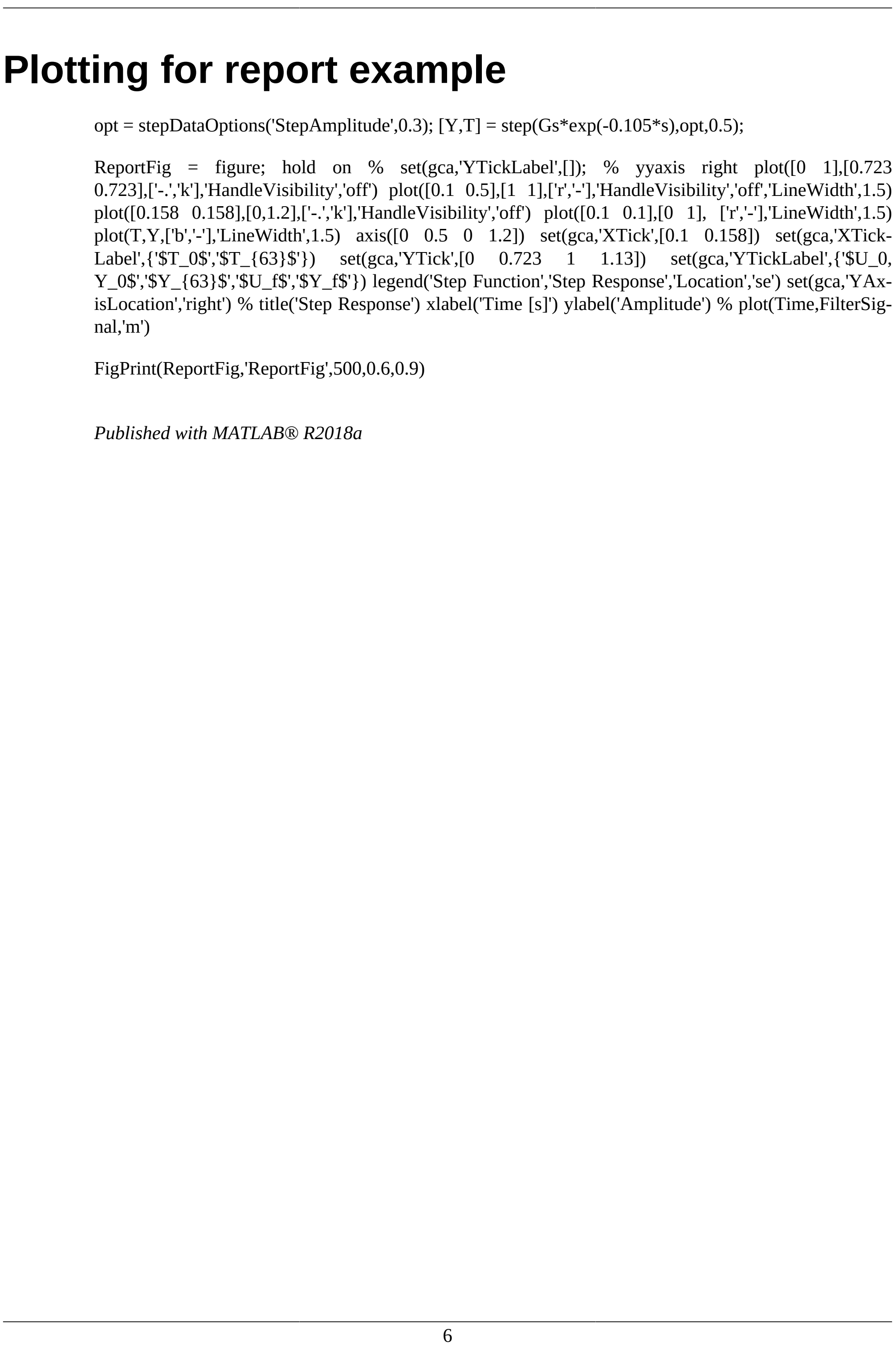
```
Cake =

  Figure (2) with properties:
        Number: 2
          Name: ''
         Color: [0.9400 0.9400 0.9400]
      Position: [488 342 560 420]
         Units: 'pixels'

  Use GET to show all properties


g = 

  Line with properties:

              Color: [1 0 1]
          LineStyle: '-'
          LineWidth: 0.5000
             Marker: 'none'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [1×1973 double]
              YData: [1×1973 double]
              ZData: [1×0 double]

  Use GET to show all properties
```





Plotting for bode plot

```
% [MAG PHASE W] = bode(Gs1,{0.01 1000});
% [MAGs PHASEs Ws] = bode(Gs2,{0.01 1000});
% [MAGss PHASEss Wss] = bode(Gs3,{0.01 1000});
%
% MAG1 = zeros(75,1);
% MAG2 = MAG1;
% MAG3 = MAG2;
% for i=1:1:75
%     MAG1(i,1) = 20*log10(MAG(1,1,i));
%     MAG2(i,1) = 20*log10(MAGs(1,1,i));
%     MAG3(i,1) = 20*log10(MAGss(1,1,i));
% end
% Phase1 = zeros(75,1);
% Phase2 = Phase1;
% Phase3 = Phase2;
% for i=1:1:75
%     Phase1(i,1) = PHASE(1,1,i);
%     Phase2(i,1) = PHASEs(1,1,i);
%     Phase3(i,1) = PHASEss(1,1,i);
% end
% WheelP = figure;
% subplot(2,1,1)
% semilogx(W, MAG1)
% hold on
% semilogx(Ws,MAG2)
% semilogx(Wss,MAG3)
% hold off
% axis([0.1 1000 -10 35])
% xlabel('Frequency (rad/s)')
% ylabel('Magnitude (dB)')
% set(WheelP.CurrentAxes,'TickLabelInterpreter','latex');
% set(WheelP.CurrentAxes,'FontSize',11);
% set(WheelP.CurrentAxes.Title,'Interpreter','latex','FontSize',13);
% set(WheelP.CurrentAxes.Legend,'Interpreter','latex','FontSize',11);
% set(WheelP.CurrentAxes.XLabel,'Interpreter','latex','FontSize',11);
% set(WheelP.CurrentAxes.YLabel,'Interpreter','latex','FontSize',11);
% grid
% subplot(2,1,2)
% semilogx(W,Phase1)
% hold on
% semilogx(W,Phase2)
% semilogx(W,Phase3)
% axis([0.01 1000 -100 0])
% xlabel('Frequency (rad/s)')
% ylabel('Phase (deg)')
% legend(['$T_{63}$ = ' num2str(T63_1) 's'], ['$T_{63}$
 = ' num2str(T63_2) 's'], ['$T_{63}$ = ' num2str(T63_3)
 's'],'location','sw');
% grid
% FigPrint(WheelP,'WheelBode',500,0.67,1);
```



Plotting for report example

```
opt = stepDataOptions('StepAmplitude',0.3); [Y,T] = step(Gs*exp(-0.105*s),opt,0.5);
```

```
ReportFig = figure; hold on % set(gca,'YTickLabel',[]); % yyaxis right plot([0 1],[0.723 0.723],['-.','k'],'HandleVisibility','off') plot([0.1 0.5],[1 1],['r','-'],'HandleVisibility','off','LineWidth',1.5) plot([0.158 0.158],[0,1.2],['-.','k'],'HandleVisibility','off') plot([0.1 0.1],[0 1], ['r','-'],'LineWidth',1.5) plot(T,Y,['b','-'],'LineWidth',1.5) axis([0 0.5 0 1.2]) set(gca,'XTick',[0.1 0.158]) set(gca,'XTickLabel',{'$T_0$','$T_{63}$'}) set(gca,'YTick',[0 0.723 1 1.13]) set(gca,'YTickLabel',{'$U_0, Y_0$','$Y_{63}$','$U_f$','$Y_f$'}) legend('Step Function','Step Response','Location','se') set(gca,'YAxisLocation','right') % title('Step Response') xlabel('Time [s]') ylabel('Amplitude') % plot(Time,FilterSignal,'m')
```

```
FigPrint(ReportFig,'ReportFig',500,0.6,0.9)
```

Published with MATLAB® R2018a

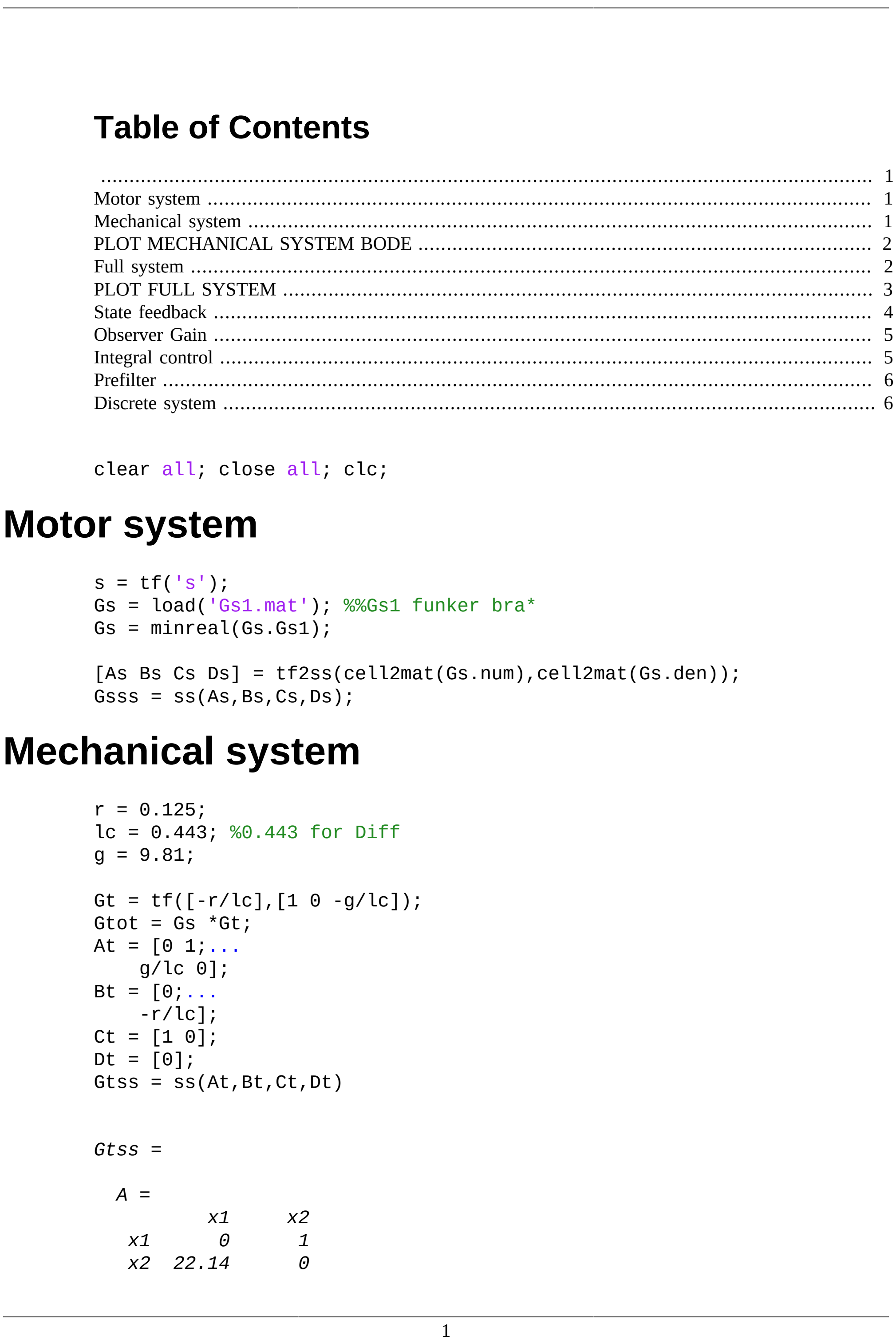


Table of Contents

```
clear all; close all; clc;
```

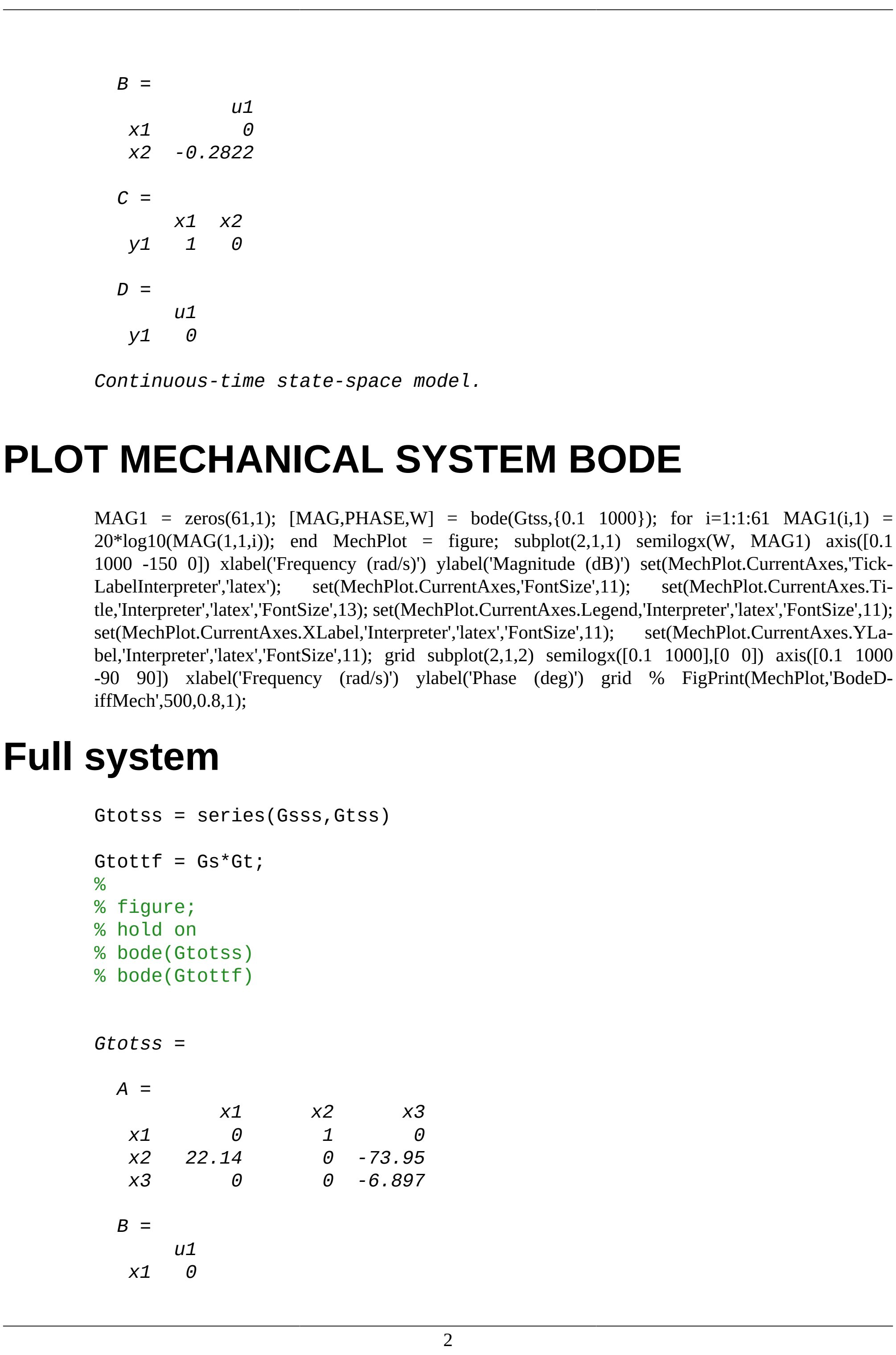
Motor system

```
s = tf('s');
Gs = load('Gs1.mat'); %%Gs1 funker bra*
Gs = minreal(Gs.Gs1);

[As Bs Cs Ds] = tf2ss(cell2mat(Gs.num),cell2mat(Gs.den));
Gsss = ss(As,Bs,Cs,Ds);
```

Mechanical system

```
r = 0.125;
lc = 0.443; %0.443 for Diff
g = 9.81;

Gt = tf([-r/lc],[1 0 -g/lc]);
Gtot = Gs *Gt;
At = [0 1;...
    g/lc 0];
Bt = [0;...
    -r/lc];
Ct = [1 0];
Dt = [0];
Gtss = ss(At,Bt,Ct,Dt)
```

```
Gtss =

  A =
          x1     x2
   x1      0      1
   x2  22.14      0
```



```
  B = 
          u1
   x1        0
   x2  -0.2822

  C = 
       x1  x2
   y1   1   0

  D = 
       u1
   y1   0

Continuous-time state-space model.
```

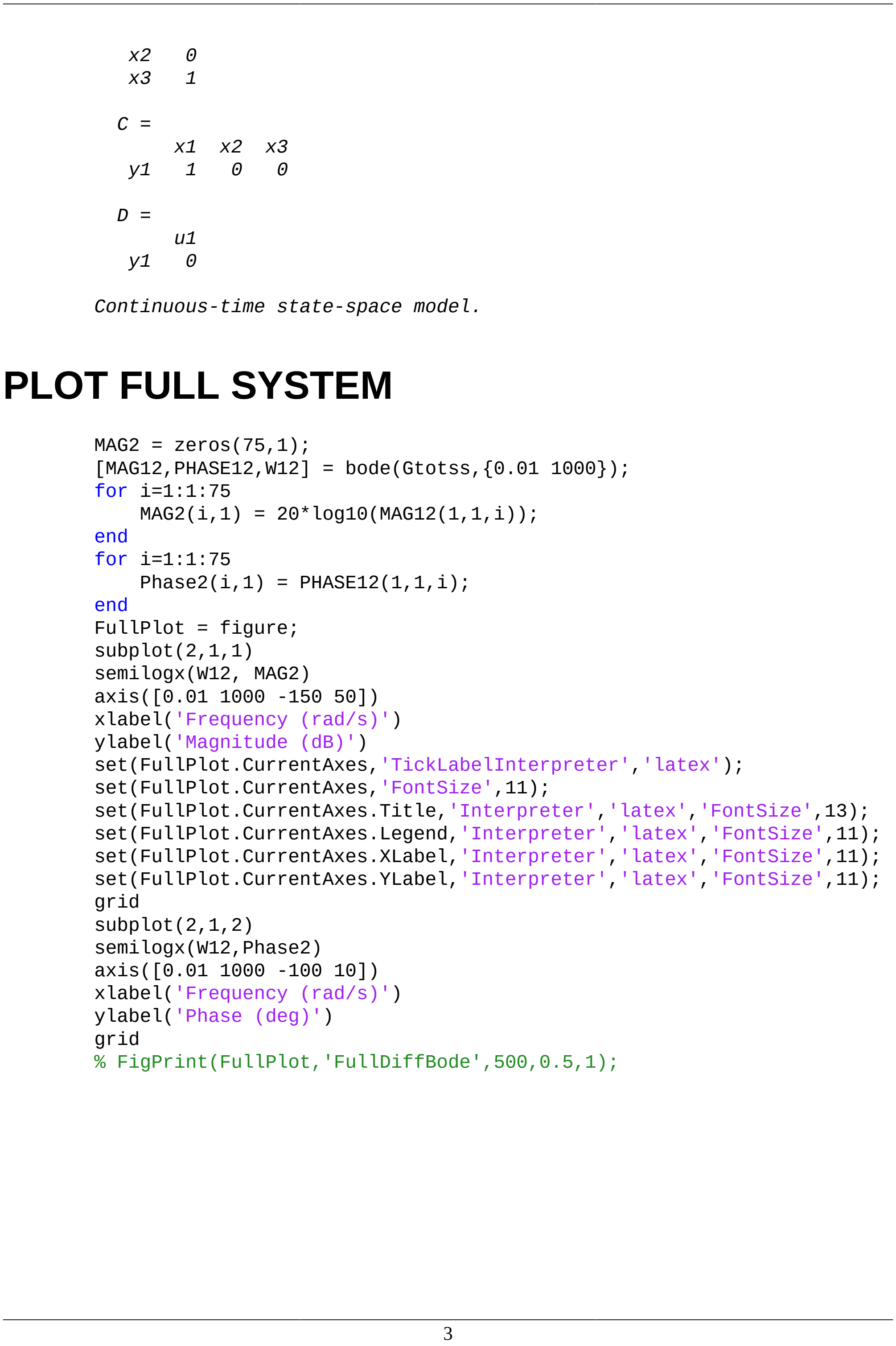
PLOT MECHANICAL SYSTEM BODE

MAG1 = zeros(61,1); [MAG,PHASE,W] = bode(Gtss,{0.1 1000}); for i=1:1:61 MAG1(i,1) = 20*log10(MAG(1,1,i)); end MechPlot = figure; subplot(2,1,1) semilogx(W, MAG1) axis([0.1 1000 -150 0]) xlabel('Frequency (rad/s)') ylabel('Magnitude (dB)') set(MechPlot.CurrentAxes,'TickLabelInterpreter','latex'); set(MechPlot.CurrentAxes,'FontSize',11); set(MechPlot.CurrentAxes.Title,'Interpreter','latex','FontSize',13); set(MechPlot.CurrentAxes.Legend,'Interpreter','latex','FontSize',11); set(MechPlot.CurrentAxes.XLabel,'Interpreter','latex','FontSize',11); set(MechPlot.CurrentAxes.YLabel,'Interpreter','latex','FontSize',11); grid subplot(2,1,2) semilogx([0.1 1000],[0 0]) axis([0.1 1000 -90 90]) xlabel('Frequency (rad/s)') ylabel('Phase (deg)') grid % FigPrint(MechPlot,'BodeDiffMech',500,0.8,1);

Full system

```
Gtotss = series(Gsss,Gtss)

Gtottf = Gs*Gt;
%
% figure;
% hold on
% bode(Gtotss)
% bode(Gtottf)
```

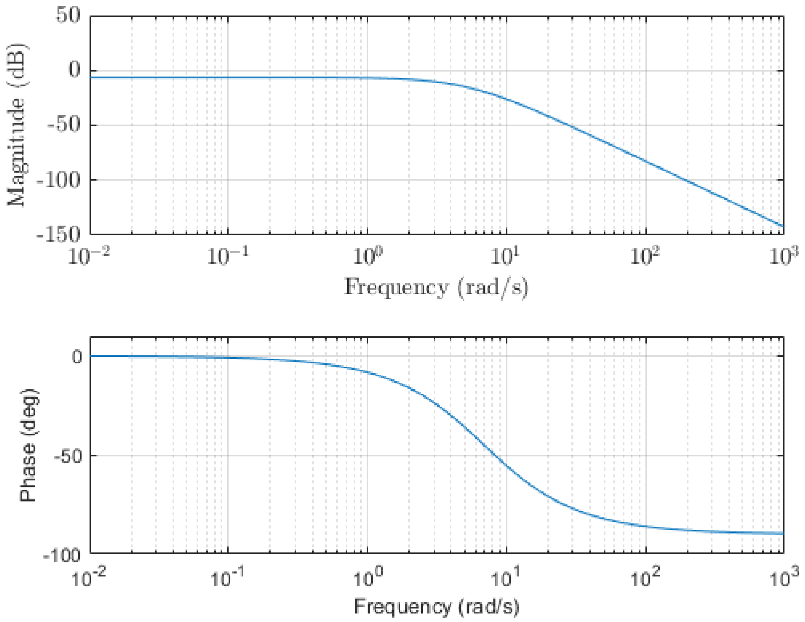
```
Gtotss =
 
  A = 
           x1      x2      x3
   x1       0       1       0
   x2   22.14       0  -73.95
   x3       0       0  -6.897
 
  B = 
       u1
   x1   0
```



```
   x2   0
   x3   1

  C =
       x1  x2  x3
   y1   1   0   0

  D =
       u1
   y1   0

Continuous-time state-space model.
```

PLOT FULL SYSTEM

```
MAG2 = zeros(75,1);
[MAG12,PHASE12,W12] = bode(Gtotss,{0.01 1000});
for i=1:1:75
    MAG2(i,1) = 20*log10(MAG12(1,1,i));
end
for i=1:1:75
    Phase2(i,1) = PHASE12(1,1,i);
end
FullPlot = figure;
subplot(2,1,1)
semilogx(W12, MAG2)
axis([0.01 1000 -150 50])
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
set(FullPlot.CurrentAxes,'TickLabelInterpreter','latex');
set(FullPlot.CurrentAxes,'FontSize',11);
set(FullPlot.CurrentAxes.Title,'Interpreter','latex','FontSize',13);
set(FullPlot.CurrentAxes.Legend,'Interpreter','latex','FontSize',11);
set(FullPlot.CurrentAxes.XLabel,'Interpreter','latex','FontSize',11);
set(FullPlot.CurrentAxes.YLabel,'Interpreter','latex','FontSize',11);
grid
subplot(2,1,2)
semilogx(W12,Phase2)
axis([0.01 1000 -100 10])
xlabel('Frequency (rad/s)')
ylabel('Phase (deg)')
grid
% FigPrint(FullPlot,'FullDiffBode',500,0.5,1);
```

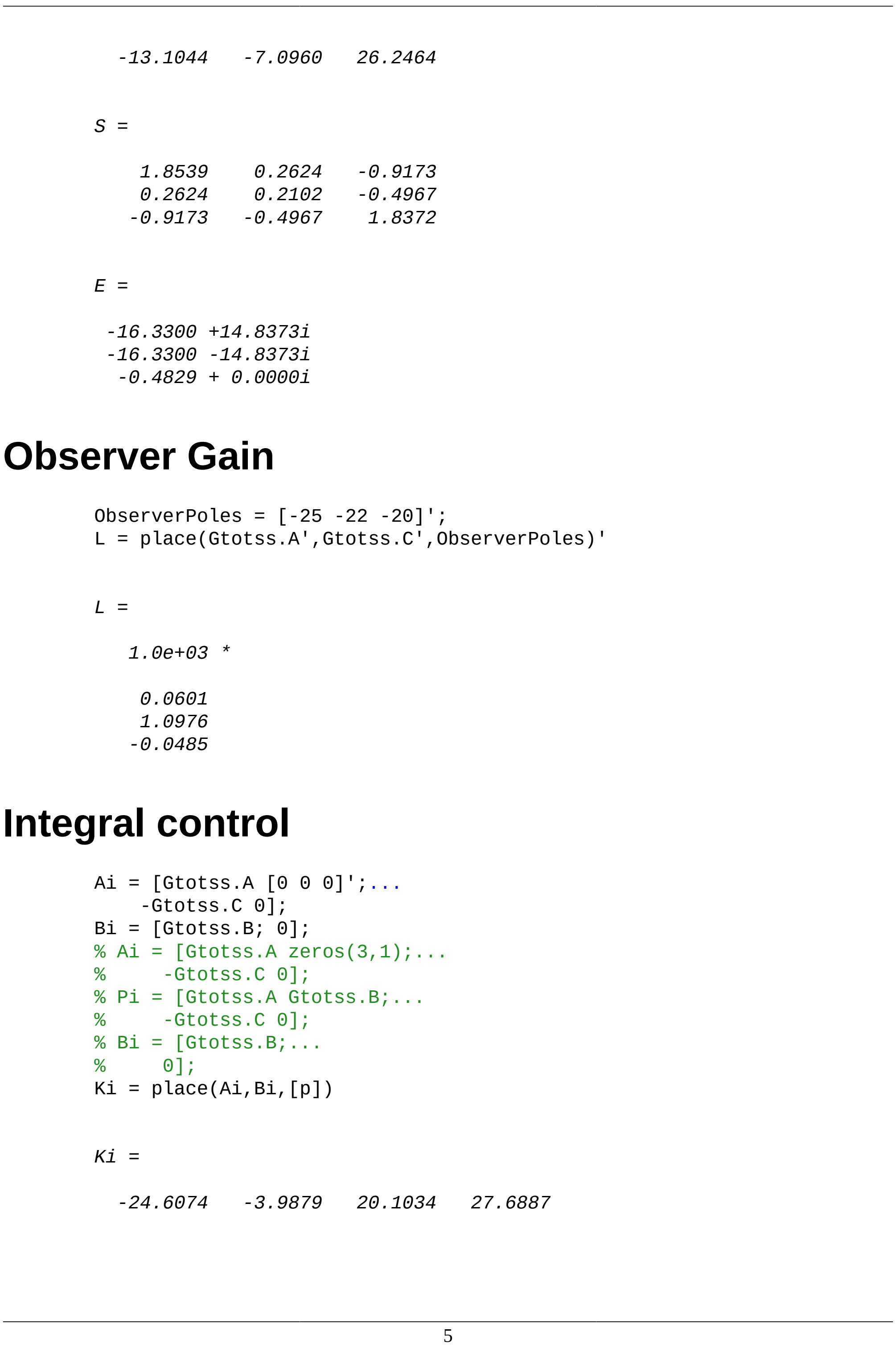


State feedback

p = [-0.1 -0.2 -0.3 -0.4];

```
p = [-6 -7 -7.5 -6.5] %for Gs1 fra 5.5--> %
% p = [-10 -11 -12 -13]
K = place(Gtotss.A,Gtotss.B,p(1:3))
% K = place([Gtotss.A Gtotss.B;[0 0 0 0]],[0;0;0;1],p)
Q = [0.4 0 0;...
    0 3 0;...
    0 0 0.1];
[Ks,S,E] = lqr(Gtotss,Q,0.07)
```

```
p =

   -6.0000   -7.0000   -7.5000   -6.5000


K =

  -10.3988   -2.1859   13.6034


Ks =
```



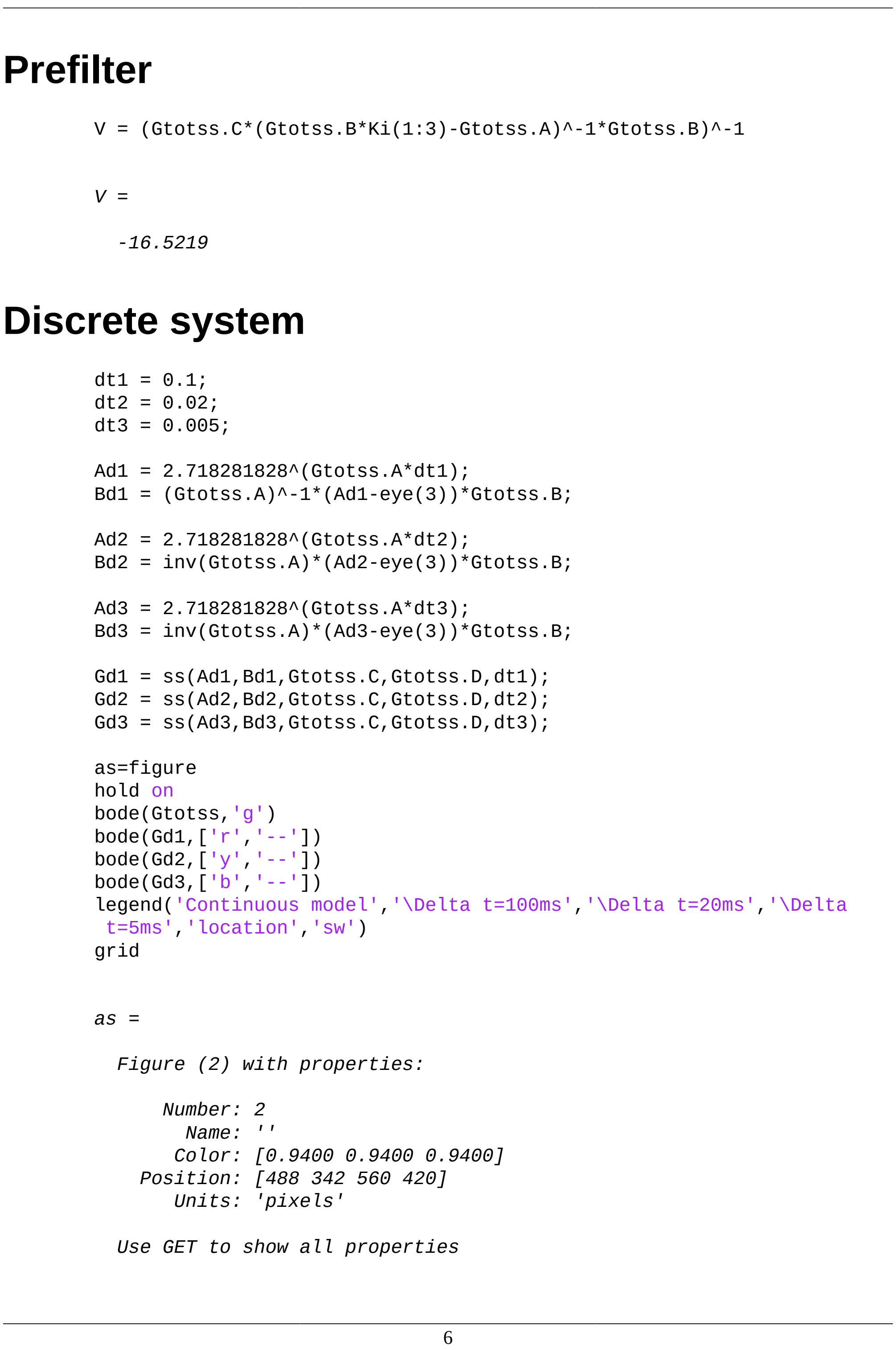
```
  -13.1044   -7.0960   26.2464


S =

    1.8539    0.2624   -0.9173
    0.2624    0.2102   -0.4967
   -0.9173   -0.4967    1.8372


E =

 -16.3300 +14.8373i
 -16.3300 -14.8373i
  -0.4829 + 0.0000i
```

Observer Gain

```
ObserverPoles = [-25 -22 -20]';
L = place(Gtotss.A',Gtotss.C',ObserverPoles)'


L =

   1.0e+03 *

    0.0601
    1.0976
   -0.0485
```

Integral control

```
Ai = [Gtotss.A [0 0 0]';...
    -Gtotss.C 0];
Bi = [Gtotss.B; 0];
% Ai = [Gtotss.A zeros(3,1);...
%     -Gtotss.C 0];
% Pi = [Gtotss.A Gtotss.B;...
%     -Gtotss.C 0];
% Bi = [Gtotss.B;...
%     0];
Ki = place(Ai,Bi,[p])


Ki =

  -24.6074   -3.9879   20.1034   27.6887
```

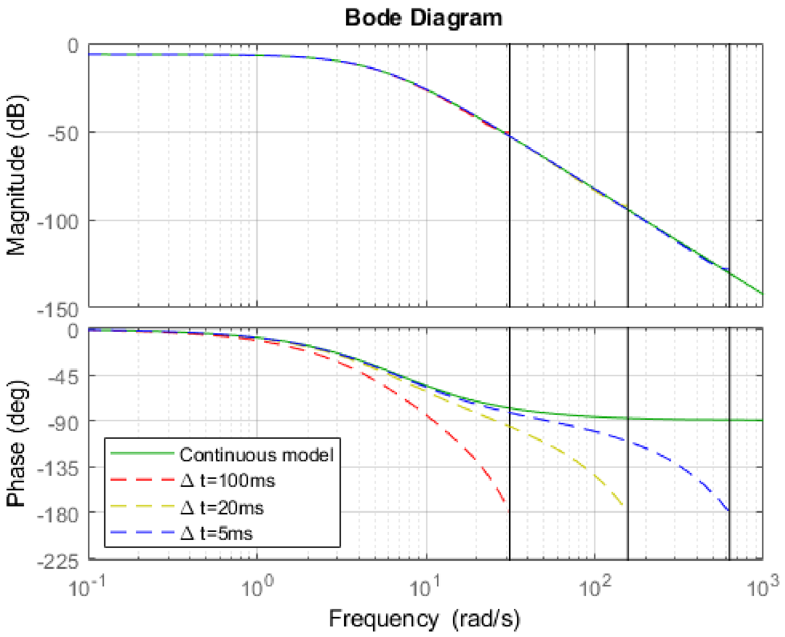


Prefilter

```
V = (Gtotss.C*(Gtotss.B*Ki(1:3)-Gtotss.A)^-1*Gtotss.B)^-1


V =

  -16.5219
```

Discrete system

```
dt1 = 0.1;
dt2 = 0.02;
dt3 = 0.005;

Ad1 = 2.718281828^(Gtotss.A*dt1);
Bd1 = (Gtotss.A)^-1*(Ad1-eye(3))*Gtotss.B;

Ad2 = 2.718281828^(Gtotss.A*dt2);
Bd2 = inv(Gtotss.A)*(Ad2-eye(3))*Gtotss.B;

Ad3 = 2.718281828^(Gtotss.A*dt3);
Bd3 = inv(Gtotss.A)*(Ad3-eye(3))*Gtotss.B;

Gd1 = ss(Ad1,Bd1,Gtotss.C,Gtotss.D,dt1);
Gd2 = ss(Ad2,Bd2,Gtotss.C,Gtotss.D,dt2);
Gd3 = ss(Ad3,Bd3,Gtotss.C,Gtotss.D,dt3);

as=figure
hold on
bode(Gtotss,'g')
bode(Gd1,['r','--'])
bode(Gd2,['y','--'])
bode(Gd3,['b','--'])
legend('Continuous model','\Delta t=100ms','\Delta t=20ms','\Delta
 t=5ms','location','sw')
grid


as =

  Figure (2) with properties:

      Number: 2
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [488 342 560 420]
       Units: 'pixels'

  Use GET to show all properties
```



Published with MATLAB® R2018a

D Simplified vehicle calculations

The following calculations are mainly done in order to find the average torque required based on an average translational acceleration. The calculations done in MATLAB on the next page is based on the Figures 156 and 157.

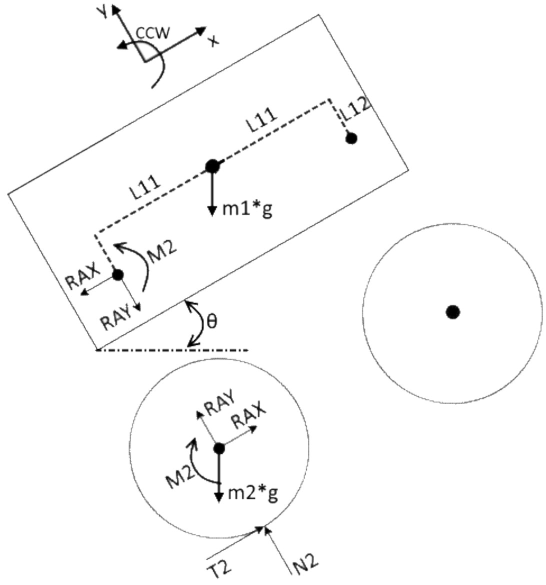


Figure 156: Free body diagram

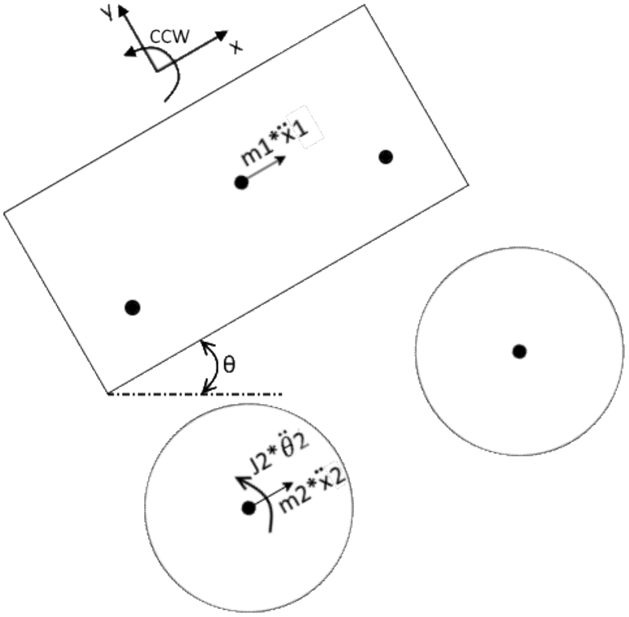
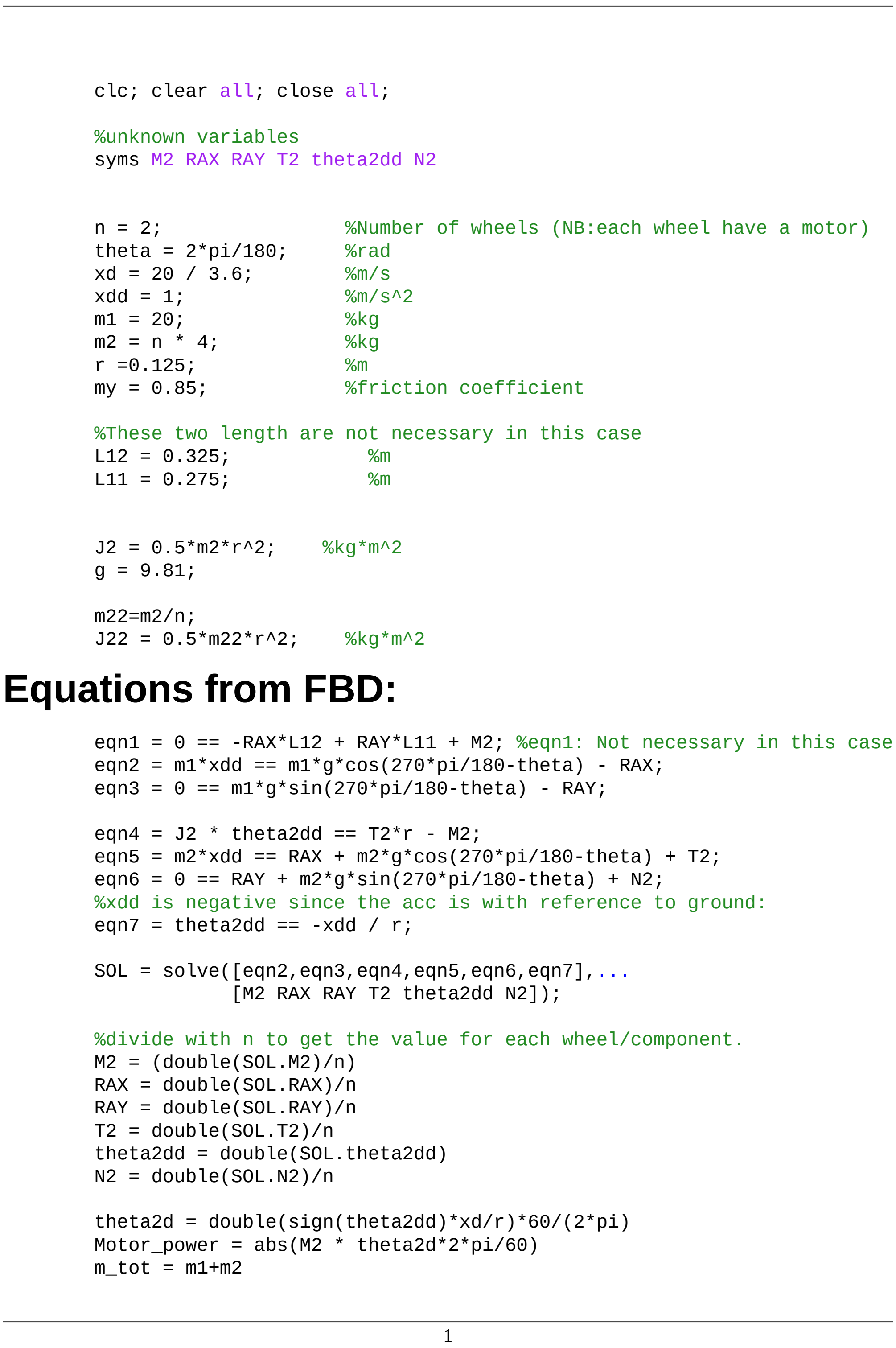


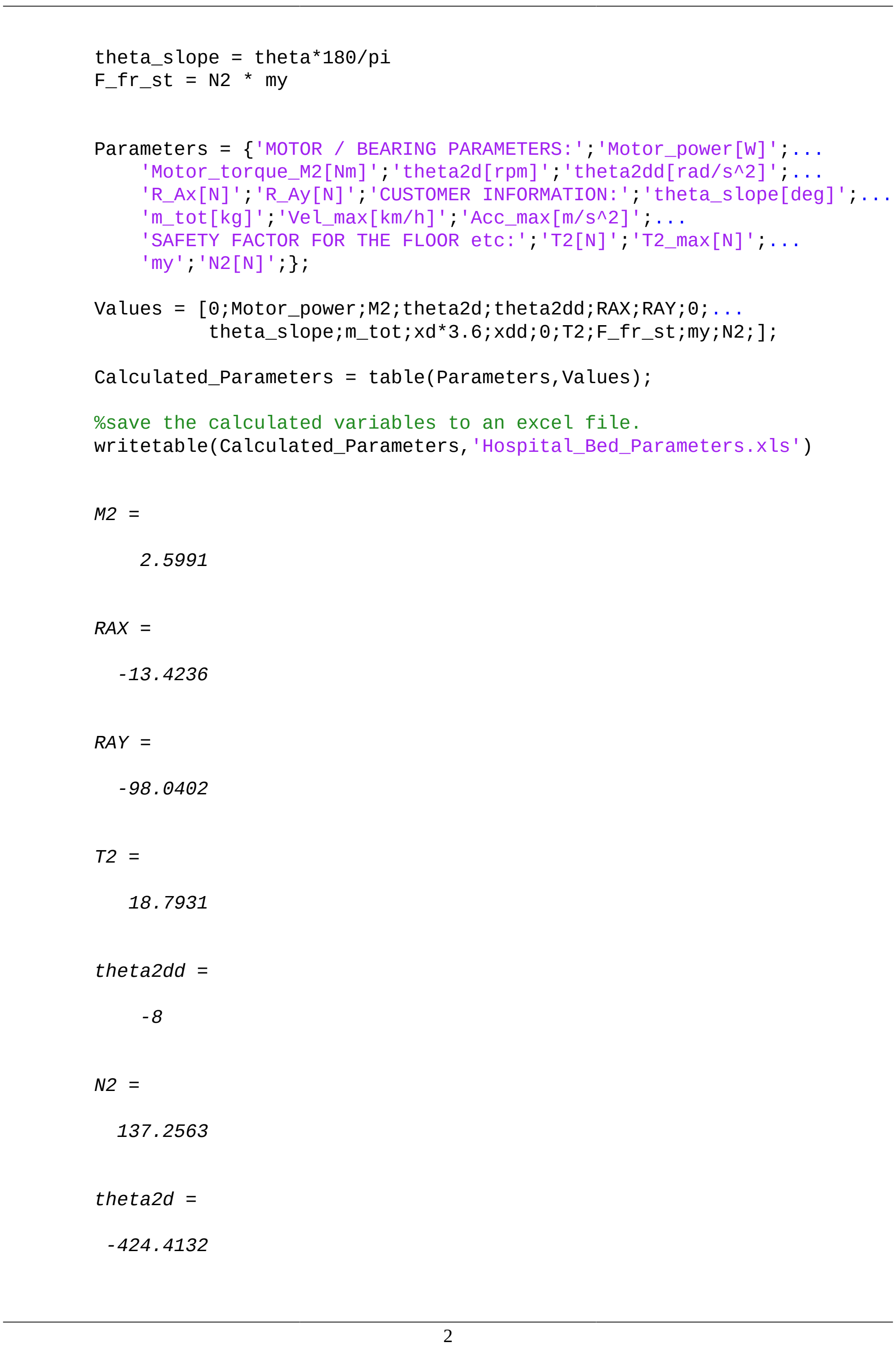
Figure 157: Kinetic diagram



```
clc; clear all; close all;

%unknown variables
syms M2 RAX RAY T2 theta2dd N2


n = 2;                %Number of wheels (NB:each wheel have a motor)
theta = 2*pi/180;     %rad
xd = 20 / 3.6;        %m/s
xdd = 1;              %m/s^2
m1 = 20;              %kg
m2 = n * 4;           %kg
r =0.125;             %m
my = 0.85;            %friction coefficient

%These two length are not necessary in this case
L12 = 0.325;            %m
L11 = 0.275;            %m


J2 = 0.5*m2*r^2;    %kg*m^2
g = 9.81;

m22=m2/n;
J22 = 0.5*m22*r^2;    %kg*m^2
```

Equations from FBD:

```
eqn1 = 0 == -RAX*L12 + RAY*L11 + M2; %eqn1: Not necessary in this case
eqn2 = m1*xdd == m1*g*cos(270*pi/180-theta) - RAX;
eqn3 = 0 == m1*g*sin(270*pi/180-theta) - RAY;

eqn4 = J2 * theta2dd == T2*r - M2;
eqn5 = m2*xdd == RAX + m2*g*cos(270*pi/180-theta) + T2;
eqn6 = 0 == RAY + m2*g*sin(270*pi/180-theta) + N2;
%xdd is negative since the acc is with reference to ground:
eqn7 = theta2dd == -xdd / r;

SOL = solve([eqn2,eqn3,eqn4,eqn5,eqn6,eqn7],...
            [M2 RAX RAY T2 theta2dd N2]);

%divide with n to get the value for each wheel/component.
M2 = (double(SOL.M2)/n)
RAX = double(SOL.RAX)/n
RAY = double(SOL.RAY)/n
T2 = double(SOL.T2)/n
theta2dd = double(SOL.theta2dd)
N2 = double(SOL.N2)/n

theta2d = double(sign(theta2dd)*xd/r)*60/(2*pi)
Motor_power = abs(M2 * theta2d*2*pi/60)
m_tot = m1+m2
```



```
theta_slope = theta*180/pi
F_fr_st = N2 * my


Parameters = {'MOTOR / BEARING PARAMETERS:';'Motor_power[W]';...
    'Motor_torque_M2[Nm]';'theta2d[rpm]';'theta2dd[rad/s^2]';...
    'R_Ax[N]';'R_Ay[N]';'CUSTOMER INFORMATION:';'theta_slope[deg]';...
    'm_tot[kg]';'Vel_max[km/h]';'Acc_max[m/s^2]';...
    'SAFETY FACTOR FOR THE FLOOR etc:';'T2[N]';'T2_max[N]';...
    'my';'N2[N]';};

Values = [0;Motor_power;M2;theta2d;theta2dd;RAX;RAY;0;...
          theta_slope;m_tot;xd*3.6;xdd;0;T2;F_fr_st;my;N2;];

Calculated_Parameters = table(Parameters,Values);

%save the calculated variables to an excel file.
writetable(Calculated_Parameters,'Hospital_Bed_Parameters.xls')


M2 =

    2.5991


RAX =

  -13.4236


RAY =

  -98.0402


T2 =

   18.7931


theta2dd =

    -8


N2 =

  137.2563


theta2d =

 -424.4132
```

```
Motor_power =

  115.5172


m_tot =

    28


theta_slope =

     2


F_fr_st =

  116.6679
```

Published with MATLAB® R2018a

E Various

E.1 BLDC Motor Characteristics

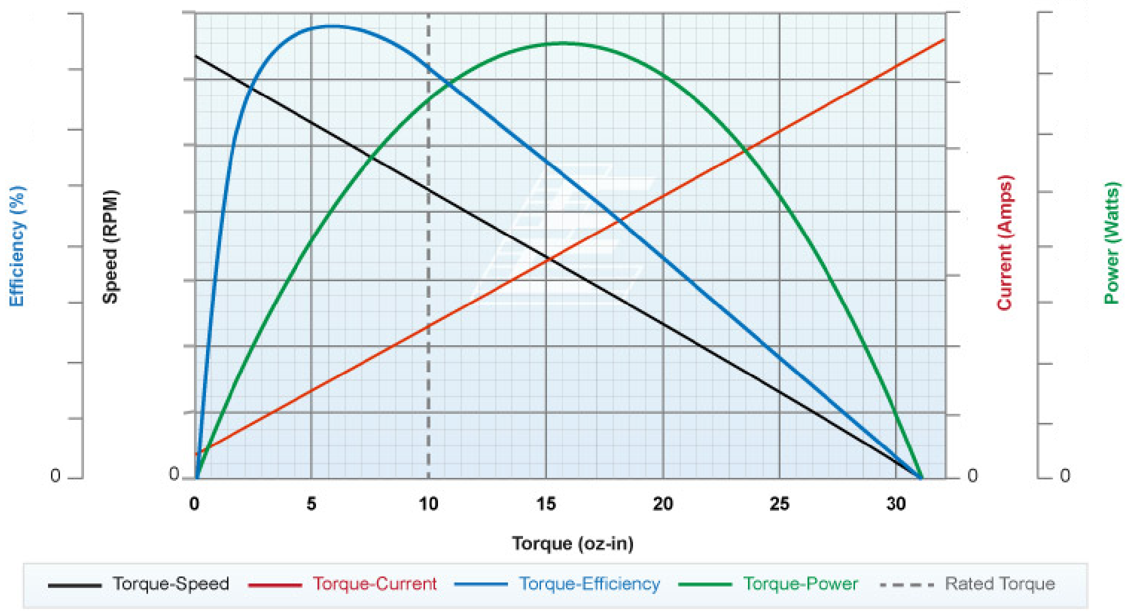
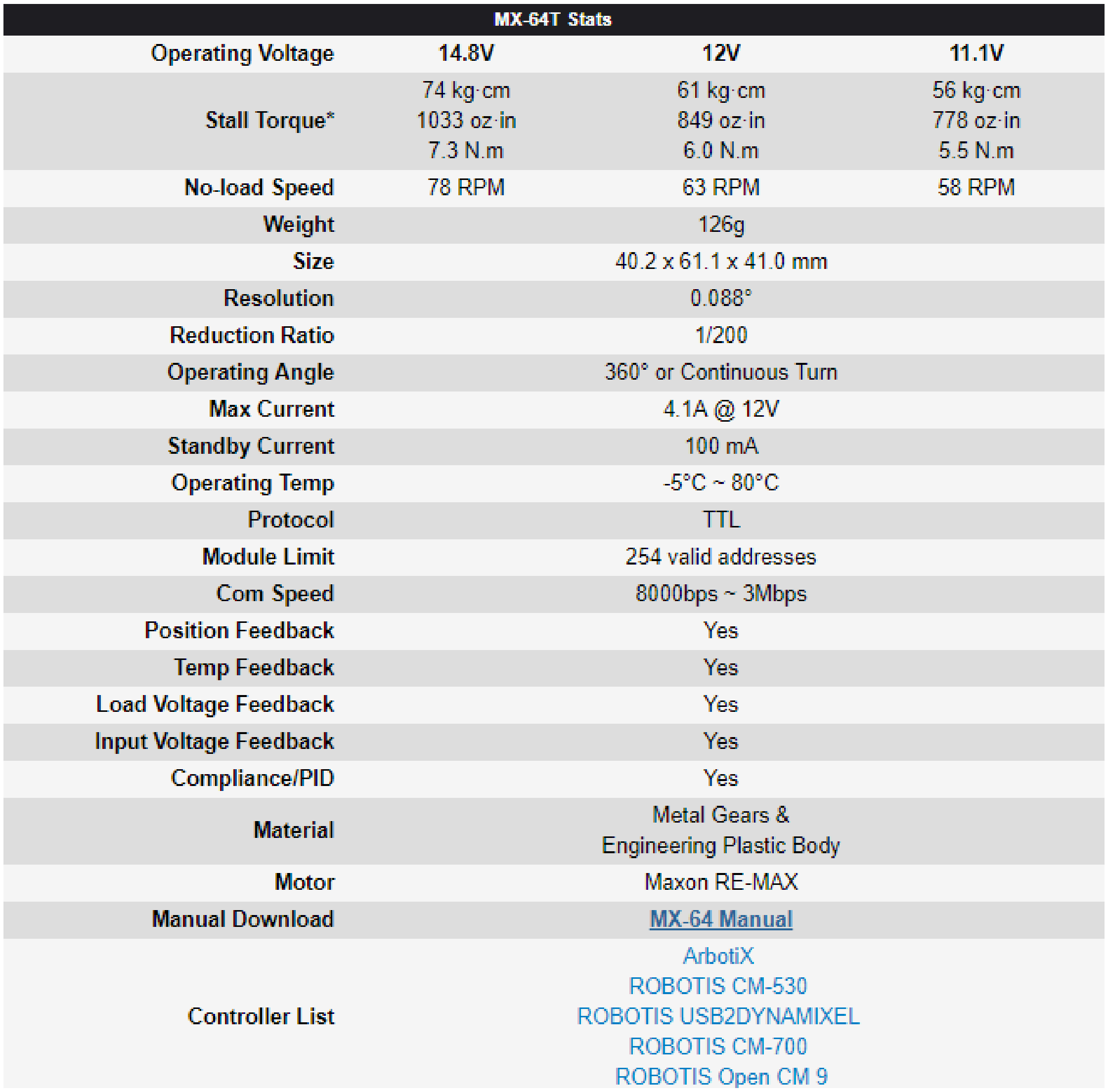


Figure 158: Typical curves for a BLDC motor [8]

F Datasheets

F.1 Dynamixel MX-64T



MX-64T Stats

Operating Voltage	14.8V	12V	11.1V
Stall Torque*	74 kg·cm 1033 oz·in 7.3 N.m	61 kg·cm 849 oz·in 6.0 N.m	56 kg·cm 778 oz·in 5.5 N.m
No-load Speed	78 RPM	63 RPM	58 RPM
Weight	126g		
Size	40.2 x 61.1 x 41.0 mm		
Resolution	0.088°		
Reduction Ratio	1/200		
Operating Angle	360° or Continuous Turn		
Max Current	4.1A @ 12V		
Standby Current	100 mA		
Operating Temp	-5°C ~ 80°C		
Protocol	TTL		
Module Limit	254 valid addresses		
Com Speed	8000bps ~ 3Mbps		
Position Feedback	Yes		
Temp Feedback	Yes		
Load Voltage Feedback	Yes		
Input Voltage Feedback	Yes		
Compliance/PID	Yes		
Material	Metal Gears & Engineering Plastic Body		
Motor	Maxon RE-MAX		
Manual Download	MX-64 Manual		
Controller List	ArbotiX ROBOTIS CM-530 ROBOTIS USB2DYNAMIXEL ROBOTIS CM-700 ROBOTIS Open CM 9		

G User Guide

G.1 LabVIEW Packages

LabVIEW:

- LabVIEW 2017 (32-bit)
- VI Package Manager

LabVIEW Add-Ons: Design:

- LabVIEW 2017 Control Design and Simulation Module
- LabVIEW 2017 MathScript RT Module
- Robotics Module

LabVIEW Add-Ons: Deploy:

- LabVIEW 2017 Real-Time Module
- LabVIEW 2017 myRIO Toolkit

DIAdem:

- DIAdem 2017 (32-bit)

Device Drivers:

- NI-DAQmx
- NI ELVISmx

G.2 Procedure

1. Connect the power line to the battery package
2. Connect to the myRIO through the wifi ”hagrid” with the password ”vidda123”.
3. Open the ”MAS500” project
4. Run the ”Main” VI under BalancingRio WiFi.
5. Boot the ”Main” program, if the angular velocity of the IMU is offset, re-boot the program.
6. Run the ”JoystickLoop” VI to connect with the Xbox controller.
7. Disable ”Manual?” button, and then disable the ”RESET INTEGRATOR” button to initiate balancing.
8. The ”pitch offset” might need to be redefined if the robot leans toward one side.