1 Oktober 2018

MASTER OF SCIENCE THESIS

REDESIGN & IMPLEMENTATION OF A MOMENT EXCHANGE UNICYCLE ROBOT

J.F. (Franck) De Vries

Faculty of Engineering Technology (ET)
Structural Dynamics, Acoustics & Control (SDAC)

Exam committee:
Dr.ir. R.G.K.M. Aarts
Dr.ir. J. van Dijk
Dr.ir. T.J.A. de Vries

ET.18/TM-5837
Mechanics of Solids, Surfaces and Systems (MS^3)
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE.

Summary

Unicycles have a very small footprint, giving them a lot of potential for certain applications, one of which is delivery drones. In this project the feasibility of a specific concept known as the Moment Exchange Unicycle Robot (MEUR) is investigated. More specifically, it is explored whether the MEUR can be used to carry loads and to drive over sloped surfaces.

To do this, first the existing prototype has been evaluated. A new sensor algorithm has been created in order to obtain more accurate angle measurements and side wheels have been added to the prototype in order to allow for easy testing.

Next various control systems, which were found to be working in literature, have been created and the performance of these controllers has been compared using simulations. It was found that for both the lateral and longitudinal direction the sliding mode controller gives the best performance.

The sliding mode controllers have been implemented and tuned, showing that they are able to stabilize the actual unicycle robot. Next the performance of the unicycle for tracking a reference has been tested and it has been found that the unicycle gives better tracking performance compared to the results found in literature.

Finally, experiments have been performed to investigate the ability of the unicycle to carry additional loads and drive along sloped surfaces. They showed that additional loads of up to 0.8 kg can be carried and sloped surfaces of up to 12° in either direction can be driven over.

Preface

This report and the work it describes conclude my master thesis project at the University of Twente as part of my Master of Science in Systems & Control. It was a great opportunity to apply the knowledge gained during my studies on a practical problem and to experience which kinds of problems are involved in such a project. This project has definitely taught me a lot about the process involved in designing a control system and applying it in an actual robot.

I would like to thank the Structural Dynamics, Acoustics & Control (SDAC) group for this opportunity. In particular, I would like to thank my supervisors Ronald Aarts and Johannes van Dijk for the guidance they provided during the project. I would also like to thank Leo Tiemersma for his part in creating and obtaining the parts required for the unicycle to work successfully. Next to that, I would also like to thank my friends and family for supporting me during this time, in particular Pim Witkop and Dennis Niehoff for the various discussions we have had and the help they have provided. Finally, I would like to thank Theo de Vries for being a part of the exam committee and playing a role in grading my work.

I hope you enjoy reading this report. If anything needs further clarification, please do not hesitate to reach out.

Kind regards,

Franck de Vries
`www.linkedin.com/in/franckdevries`

Contents

List of Abbreviations and Symbols

Abbreviations

CoM	Center of Mass
DW	Drive Wheel
EMF	Electromotive Force
IMU	Inertial Measurement Unit
LiPo	Lithium Polymer
LQR	Linear Quadratic Regulator
MEUR	Moment Exchange Unicycle Robot
PD	Proportional-Derivative
PID	Proportional–Integral–Derivative
RPM	Revolutions Per Minute
RW	Reaction Wheel
SISO	Single-Input and Single-Output
SMC	Sliding Mode Control

Symbols

a_{lon}	Horizontal acceleration of the unicycle in the longitudinal direction (m/s^2)
a_y	Acceleration measured by the y-axis of the accelerometer (m/s^2)
a_z	Acceleration measured by the z-axis of the accelerometer (old orientation) (m/s^2)
$\bar{a}_z$	Acceleration measured by the z-axis of the accelerometer (new orientation) (m/s^2)
A	System matrix in state space equations
B	Input matrix in state space equations
C, c	General constant
$C(s)$	Controller transfer function
CL	Closed-loop transfer function
CL_{eig}	Matrix containing the closed-loop eigenvalues of a system
D	Bound of the unknown system dynamics (including disturbances)
e	Error, difference between desired and actual value
f_c	Cut-off frequency of filter (Hz)
g	Gravitational acceleration: $g \approx 9.81\, m/s^2$
G	General system plant
J	Criterion used for LQR control design
J_i	Rotational inertia of body i $(kg\, m^2)$
J_{eq}	Equivalent inertia term $(kg\, m^2)$
k	Constant used for the sliding surface design of the SMC
K	Feedback control matrix
K_d, k_d	Derivative action constant used during controller design
K_i, k_i	Integral action constant used during controller design
K_p, k_p	Proportional action constant used during controller design
L_1	Distance between center of mass of body 1 and point of rotation of the roll angle (m)
L_2	Distance between center of mass of body 2 and point of rotation of the roll angle (m)
L_4	Distance between center of mass of body 4 and point of rotation of the pitch angle (m)
L_p	Distance between the point of rotation of the pitch angle and the accelerometer (m)
L_r	Distance between the point of rotation of the roll angle and the accelerometer (m)
m_{eq}	Equivalent mass
m_i	Mass of body i (kg)

m_n	Moving mass
p	System poles
q_d	Dependent coordinates
q_i	Independent coordinates
Q	Weighting matrix on the states of a system used for LQR control design
Q_i	Generalized force term used in Lagrange's equation
r	Radius of drive wheel (m)
r_{sys}	Relative degree of a system
R	Weighting matrix on the inputs of a system used for LQR control design
s	(Complex) frequency variable of the Laplace transform
S	Matrix used to derive an LQR controller
T	Kinetic energy term used in Lagrange's equation
t	Time (s)
u	General system input
u'	Discontinuous/switching system input signal of the SMC
v	Velocity (m/s)
V	Potential energy term used in Lagrange's equation
$V(\sigma)$	Lyapunov function for the sliding surface σ
W_1, W_2	Weighting matrices for the H_∞ loop-shaping controller
x	Matrix with system states in a state space equation
x'	General state variable used in the derivation for the SMC
x_i	Position in the x-direction of the center of mass of body i (m)
x_p	Setpoint of the system states for an LQR controller
y	General system output
y_i	Position in the y-direction of the center of mass of body i (m)
y_r	Desired system output (reference signal)
z_i	Position in the z-direction of the center of mass of body i (m)
α	Constant determining the amount of phase-lead of a P(I)D controller
α_p	Acceleration measured by the accelerometer due to pitch angle rotation (m/s^2)
α_r	Acceleration measured by the accelerometer due to roll angle rotation (m/s^2)
β	Constant determining the start of the integral action of a P(I)D controller
ϵ	Constant determining the smoothness of the switching input of the SMC
$\theta_{acc,p}$	Pitch angle calculated using only the accelerometer data (rad)
$\theta_{acc,r}$	Roll angle calculated using only the accelerometer data (rad)
θ_p	Pitch angle of the unicycle (rad)
θ_r	Roll angle of the unicycle (rad)
ξ_i	Damping factor of the complementary filter
π	Pi, mathematical constant: $\pi \approx 3.14159$
ρ	Switching gain of the SMC
σ	Sliding surface of the sliding mode controller
τ	Time constant of the low-pass filter of a P(I)D controller
τ_i	Constant used to tune the integral action of a PID controller
τ_{lat}	Torque applied in the lateral direction (Nm)
τ_{lon}	Torque applied in the longitudinal direction (Nm)
τ_p	Constant used to tune the pole location of a P(I)D controller
τ_z	Constant used to tune the zero location of a P(I)D controller
ϕ_{dw}	Drive wheel angle (rad)
ϕ_{rw}	Reaction wheel angle (rad)
ω	Constant determining the bandwidth of the error dynamics of a SMC (rad/s)
$\omega_{0,i}$	Cut-off frequency of the complementary filter (rad/s)
ω_c	Cross-over frequency (rad/s)
ω_p	Pitch angular velocity measured using the gyroscope (rad/s)
ω_r	Roll angular velocity measured using the gyroscope (rad/s)

1. Introduction

Technology is advancing every day and more and more tasks are being automated. Autonomous delivery vehicles have already become a reality [2] and the market for autonomous delivery will undoubtedly grow the coming years. Different kinds of delivery drone concepts are being used for this. Flying drones are not bound to the streets, giving them the option to deliver packages much faster than wheeled drones, but they do offer more safety risks and cannot operate everywhere due to airspace restrictions. Wheeled drones do not have this restriction since they can simply use our road network, but in general they do take up quite a lot of space on our already busy roads since they use three or more wheels in combination with a large chassis and suspension system for stability. To greatly reduce the footprint of wheeled drones, single wheeled or unicycle robots can be used instead. An example of what such a drone could look like is the Transwheel delivery drone concept [1, 3] which can be seen in figure 1.1. Due to the unicycle design these drones have a very small footprint and can even navigate through small alleys to deliver packages at the door.

Figure 1.1.: Transwheel delivery drone concept. [1]

The design and realisation of a unicycle robot is however quite challenging from both the mechanical and control perspectives. Compared to the commonly known two-wheeled robots like the Segway [6], also sideways stability needs to be implemented, which involves additional actuation and control. This makes unicycles more complex, but does allow them to drive over sloped and uneven surfaces where two-wheeled robots would topple over.

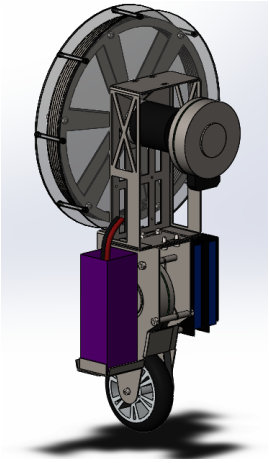


Figure 1.2.: Moment Exchange Unicycle Robot (MEUR). [4, 5]

The goal of this assignment is to investigate the feasibility of a specific concept known as the Moment Exchange Unicycle Robot (MEUR). This concept uses a wheel with a large inertia for stabilisation in the lateral direction. During a previous student project in the group Structural Dynamics, Acoustics and Control (SDAC) at the University of Twente a prototype MEUR has been build [4, 5], which can be seen in figure 1.2. However, controlled motion has not been demonstrated yet due to time constraints and issues with the sensors, so this prototype is taken as the starting point. In this project a new sensor algorithm has been created and experiments have been performed carrying additional loads and driving over sloped surfaces in order to answer the following research question:

"Is it feasible to use the Moment Exchange Unicycle Robot as a delivery drone?"

In this thesis the background information and findings from previous studies are described and used to further clarify the research question in chapter 2. Chapter 3 describes the evaluation and modification of the prototype. In chapter 4 the design of various control systems for the unicycle are described and evaluated using simulations. In chapter 5 the implementation of the controllers is described and experimental results are shown. A discussion of these results is given in chapter 6, after which conclusions are drawn in chapter 7 and chapter 8 will give further recommendations on how the work presented in this thesis can be improved in future work.

2. Background

2.1. Wheeled Systems

In order for wheeled systems stay upright intrinsically, a minimum of three wheels is required. Two wheeled systems, like bicycles (two wheels in line) and dicycles (two wheels with a common wheel axis), do not posses this static stability. Bicycles are dynamically stable above a certain velocity [8], but at lower velocities stabilization is required in the lateral direction, which can for example be done by using an inertia wheel [7], like the robot shown in figure 2.1. Dicycles, like the Segway [6], have an unstable equilibrium located directly above the center of their wheels and need to be stabilized in the longitudinal direction, which can be done by driving forwards and backwards.

Figure 2.1.: Murata Boy, a bicycle-riding robot. [7]

Systems which only use one wheel are either a monowheel or a unicycle depending on where their center of gravity is positioned [10]. Monowheels have their center of gravity located below the center of the wheel, which is usually achieved by using a relatively large wheel, making them only unstable in the lateral direction. An example of a self-stabilizing monowheel robot can be seen in figure 2.2, which uses a balancing lever to change its center of gravity and produce a moment in order to stabilize and steer [9]. Unicycles have their center of gravity located above the center of the wheel, making them also unstable in the longitudinal direction, but allowing their wheel to be much smaller.

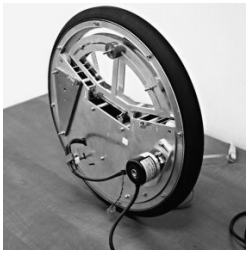


Figure 2.2.: Self-stabilizing monowheel robot prototype. [9]

2.2. Unicycle Robots

A general schematic of the orientations of a unicycle can be seen in figure 2.3. Unicycles need to be stabilized in both the lateral and longitudinal direction. Similarly as for dicycles, the longitudinal direction is generally stabilized by using the drive wheel. Hence a single actuator controls both the position and pitch angle, thus making the system underactuated.

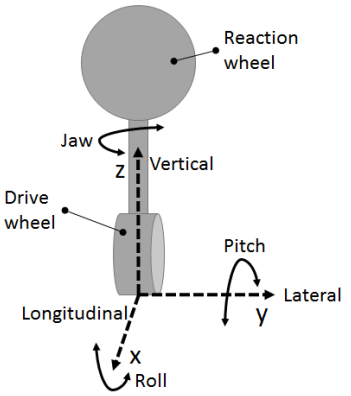


Figure 2.3.: Axis definitions of a (moment exchange) unicycle.

To stabilize the robot in the lateral direction, various techniques can be used. Unicycles and monowheels tend to fall over when their center of gravity is not located directly above the point of contact with the floor, so one way to stabilize them is by using a changing center of gravity [10, 11], similar to the monowheel shown in figure 2.2. Using this technique the robot can also be stabilized in a leaning position, making it possible to drive around corners. The drawback of this approach is that the reaction force exerted onto the unicycle when changing the center of gravity tends to push the unicycle over during standstill.

Another technique which has been used for stabilizing unicycles is by using fast rotating

flywheels which utilize the gyroscopic precession effect [12, 13, 14]. Due to precession a flywheel affects both the roll and jaw angle, but these effects can be decoupled by using two flywheels rotating in opposite direction. This makes it possible to stabilize the lateral direction while also being able to change the jaw angle to steer the unicycle. These designs do however require a lot of energy, since the gyroscopes need to be constantly rotating at high speeds.

A reaction wheel can also be used for lateral stabilization. This technique uses an exchange of moment between this wheel and the unicycle for stabilization [10, 15, 5]. Applying a torque on the reaction wheel makes it spin in a certain direction, which in turn exerts a reaction moment in the opposite direction on the unicycle. If aligned properly, this moment will mostly be in the lateral direction, so the yaw angle will not be influenced much by the lateral stabilization.

The technique using an exchange of moment does not suffer from reaction forces like the technique using a changing center of gravity, but rather utilizes them. Next to that, it does not require a lot of energy like the technique using the gyroscopic effect, since it only needs to rotate the reaction wheel when necessary. This method was therefore chosen to be used for the unicycle and this project will therefore focus on a so called Moment Exchange Unicycle Robot (MEUR).

2.3. Moment Exchange Unicycle Robots

In recent years various research groups have been working on moment exchange unicycle robots and a couple of different control techniques have been applied. Some have only shown the performance of their controllers using simulations [4, 5, 16, 17, 18], but also a few have been able to successfully implement them and test their performance experimentally. One of these is the Murata Girl [19], which not only shows lateral and longitudinal stability, but also yaw control by using an additional reaction wheel. Unfortunately the company has not published much documentation about this robot and it is therefore unknown what kinds of control techniques have been used. The other robots which have shown to be working luckily do have proper documentation and their techniques and results will be discussed in this section.

Next to researching the stability of moment exchange unicycle robots, also papers have been published investigating means of steering for the moment exchange unicycle robot. One technique which can be used for steering is by adding an actuator and making use of an additional reaction wheel, as shown by [19, 20]. A technique which does not require an additional actuator is by leaning while driving forwards or backwards, similarly as the monowheel shown in figure 2.2, but this has only been demonstrated to work in simulations [21].

2.3.1. Unibot

One of the first unicycle robots using a reaction wheel was the Unibot [15], a collaboration between the University of Twente and the University of California San Diego, which can be seen in figure 2.4. To control the longitudinal direction a Linear Quadratic Regulator (LQR) was used and for the lateral direction H_∞ loop-shaping was applied in combination with a proportional motor controller used to overcome dry friction. Experiments showed good stability robustness for the longitudinal control, but a much less robustly stable lateral system, probably caused by saturation of the motor voltage. Combined control showed that both controllers can work together to stabilize the robot and did not show much coupling. When adding a driving speed to the motor the designed controller followed the trajectory, but did not show very accurate tracking, as can be seen in figure 2.5.

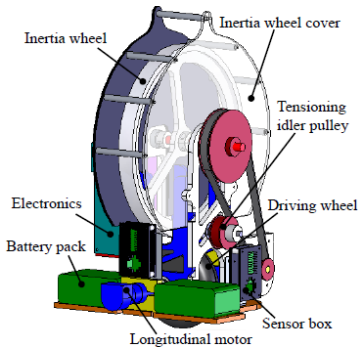


Figure 2.4.: Unibot. [15]

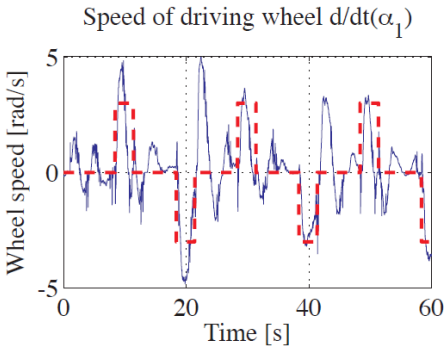


Figure 2.5.: Speed tracking performance of the Unibot. [15]

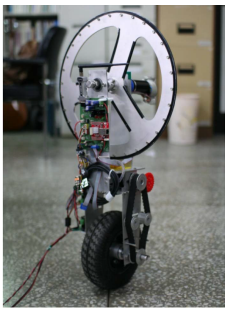


Figure 2.6.: Unicycle robot from [24].

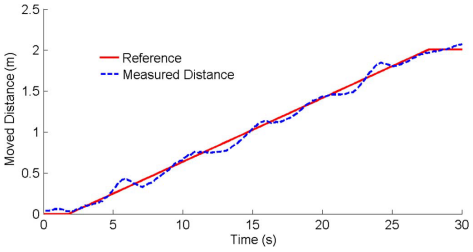


Figure 2.7.: Straight-line distance tracking of the unicycle from [25].

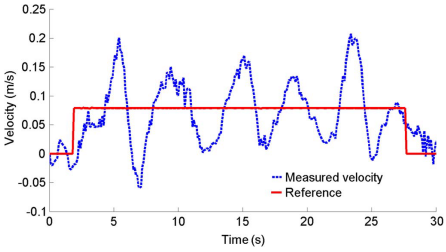


Figure 2.8.: Straight-line driving velocity of the unicycle from [25].

2.3.2. Pusan National University

A team of researchers from the Pusan National University has also developed a moment exchange unicycle robot [22, 23, 24, 25], which can be seen in figure 2.6. They initially used a fuzzy-sliding mode controller for both the lateral and longitudinal directions, but later replaced the longitudinal controller by an LQR control scheme. Experiments showed that the robot can follow a ramp (see figures 2.7 and 2.8), ladder or parabolic trajectory while keeping the pitch and roll angles within about 2° and 4° of the equilibrium position respectively. The experiments do show a steady state error of about 0.1 meters after stabilization, while the maximum tracking error is about 0.25 meters. Rather high velocity errors were obtained during the experiments, showing oscillations and reaching as high as 0.22 m/s for a reference value of 0.05 m/s, which is probably because the controller prioritizes stabilization over tracking.

2.3.3. National Cheng Kung University

At the National Cheng Kung University another moment exchange unicycle robot has been developed, which can be seen in figure 2.9. Initially they just used a reaction wheel for stabilization [26], but later a turntable was added to steer the robot as well [20]. Sliding mode control was used for the longitudinal direction, which uses a saturation function to eliminate the chattering problem. For the lateral direction initially LQR control was used, which was later replaced by a sliding mode controller. The turntable used for yaw-control (steering) is also controlled using sliding mode control.

To let the robot move from point to point, the strategy is to first balance the robot at the initial point, then set the yaw angle reference and use steering control to rotate the robot, after

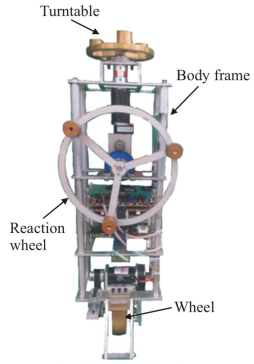


Figure 2.9.: Unicycle robot from [20].

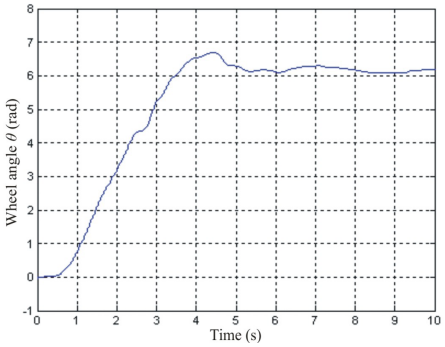


Figure 2.10.: Step tracking performance of the unicycle from [20].

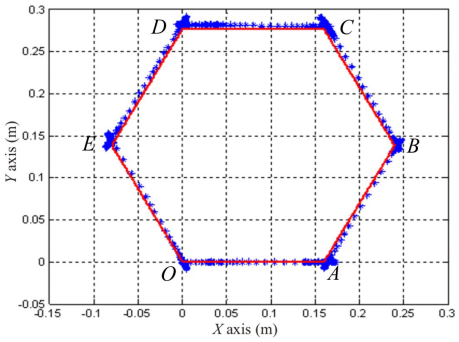


Figure 2.11.: Point to point tracking of the unicycle from [20].

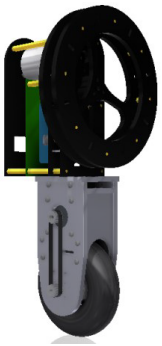


Figure 2.12.: Unicycle robot from [27].

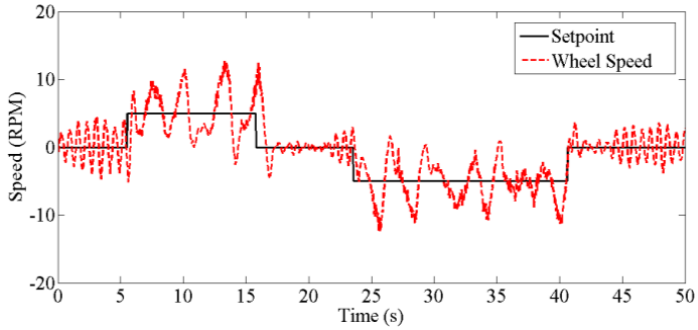


Figure 2.13.: Speed tracking performance of the unicycle from [27].

which the reference angle of the wheel is set and the robot drives in a straight line to the next point. The experimental results without yaw control show that the robot can make a step of about 0.3 meters in about three seconds with an overshoot of about 0.05 meters, see figure 2.10, while keeping the robot stable and the yaw angle about constant. After adding yaw control, the robot could move between points about 0.16 meters apart in about 10 seconds (including steering), as can be seen figure 2.11. The robot shows quite accurate tracking, but even though it uses a driving wheel with a flat profile rather than a round wheel, it still requires the reaction wheel to rotate at high speeds in order to keep the robot stable.

2.3.4. Electronic Engineering Polytechnic Institute of Surabaya

Another moment exchange unicycle robot was created by the Electronic Engineering Polytechnic Institute of Surabaya [27], which can be seen in figure 2.12. Cascaded PID control was used in the longitudinal direction, with the outer loop for speed control and the inner loop for pitch control. The roll angle was also controlled using a PID controller, which uses angular velocity feedback to limit the motion of the reaction wheel.

The experimental results showed that the pitch and roll angles could be controlled with settling times of 1.13 and 2.6 seconds, having maximum recoverable angles of 23° and 3.5° respectively, and disturbance rejection was also demonstrated. The tracking performance of the speed controller does however show rather large oscillations, as can be seen in figure 2.13. Finally, they showed that the robot can stabilize itself on a floor with a varying slope in the

longitudinal direction, with a maximum slope of 30°, with the controller adjusting the pitch angle accordingly.

2.4. Summary & Problem Statement

Unicycle robots need to be stabilized in both the lateral and longitudinal direction. For the longitudinal stabilization the drive wheel is generally used, while for the lateral stabilization a few different techniques can be used. An energy efficient way of doing this is by using a reaction wheel. Various studies have been performed on these moment exchange unicycle robots and different types of controllers have been investigated for the stabilization. Some studies have only shown the performance of their controllers in simulations, but a few have also been able to successfully implement them in an actual robot. For the lateral direction sliding mode control (SMC) is mostly used, but also PID control and H_∞ loop shaping have been successfully implemented. For the longitudinal direction state feedback control using a linear–quadratic regulator (LQR) was mostly used and shown to be effective, but also SMC and PID control have been successfully implemented.

Experiments performed using the robots showed that in general they do not suffer a lot from the coupling between the lateral and the longitudinal directions. The robots do have quite some difficulty tracking a certain path, often overshooting and sometimes suffering from a steady state error, while the obtained velocities oscillate quite a bit, even when using velocity control with a constant setpoint. These are attributed to the controllers prioritizing stabilization of the unicycle over tracking of a reference. The robustness of the controllers regarding changes in the parameters of the robots, like having them carry additional weight, has not been demonstrated yet and only limited testing has been performed for driving over sloped surfaces.

Looking at the control of the yaw angle and the steering of the robots, only experiments using a turntable have been performed so far, which require an additional actuator to control this degree of freedom. An option for steering the unicycle without using an additional actuator is by leaning, but for moment exchange unicycle robots this has only been shown to work in simulations.

For the moment exchange unicycle robot to be feasible as a delivery drone, it should not only be able to follow a trajectory while keeping the unicycle stable, as is mostly shown by current literature, but it should also be able to carry additional loads and drive over uncertain terrains. The weight of loads being carried might not be constant, so the unicycle should be able to handle this without knowing what the actual weight of the load is. Regarding uncertain terrains, the streets over which the unicycle has to travel might not be flat, but could contain bumps or have certain slopes, either in the lateral or longitudinal direction, so the unicycle should be able to handle this. These factors have not been experimentally tested yet, so the main problems which should still be investigated in order to answer the research question are:

- Can the MEUR be used to carry various loads of unknown weights?
- Can the MEUR be used to drive over sloped surfaces of unknown slope and orientation?

3. Evaluation of the Prototype

This project continues on the Moment Exchange Unicycle Robot created by Giesen et al. [4] and adapted by Dannenberg [5], which can be seen in figure 3.1. The robot uses its drive wheel for the longitudinal and a large reaction wheel for the lateral stabilization, using the axis definitions as shown in figure 2.3 on page 3. To measure the rotations of the unicycle an inertial measurement unit (IMU) was used, but it was observed that it did not give very accurate readings. Hence a new sensor set-up and algorithm to calculate the angles should be created. Next to that, in order to be able to tune the controllers and perform experiments without the unicycle falling over and getting damaged, an adjustable testing rig should be created which offers the opportunity to test the robot under various circumstances.

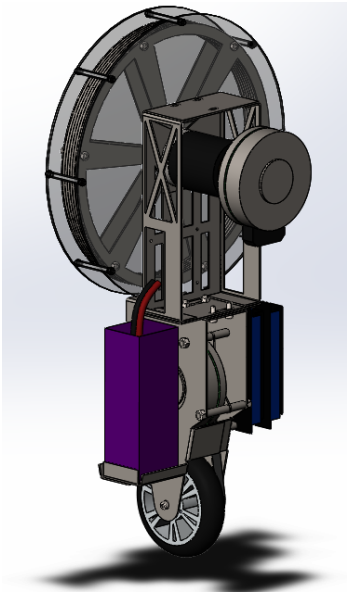


Figure 3.1.: Design of the Moment Exchange Unicycle Robot as created by Giesen et al. [4] and adapted by Dannenberg [5].

3.1. Sensor Set-up

In order to be able to control the unicycle, a few things need to be measured. The rotations of the reaction wheel and drive wheel can simply be measured using the encoders from the motors. To measure the pitch, roll and yaw angles the SparkFun 9DoF Razor IMU [28] was used, but it was observed that it did not give very accurate readings. A new sensor block has therefore been created using three analog gyroscopes (EVAL-ADXRS623 [29]) and a triaxial accelerometer (EVAL-ADXL327Z [30]). To obtain the angles from these sensors, an algorithm had to be written which processes the analog signals.

3.1.1. Initial algorithm

Since gyroscopes measure rotational velocity, their measurement data can be integrated to obtain the rotation angles, but due to noise and a bias being present in the measured signal, the integration causes drift in the obtained angles. Next to that, the initial condition for the integration is unknown when not starting exactly in the upright position, so there will be an offset in the angle. A high pass filter can be used to remove the bias and drift, making gyroscopes mostly useful for measuring high frequency changes in angle.

The accelerometer can be used to measure the angles by looking at gravity [31]. Due to its orientation, the accelerometer has its x-axis oriented in the vertical, its y-axis in the lateral and its z-axis in the longitudinal direction when the unicycle is oriented upwards. In that case the horizontal accelerometer axes (y and z) are perpendicular to gravity, so they will not measure any accelerations because of it. When the unicycle is tilted, these accelerometer axes will no longer be horizontal and will measure a certain part of the gravitational acceleration depending on the angle of rotation. When assuming the angles stay small, they can simply be calculated using the inverse sine: $\theta_{acc,p} = \arcsin\left(\frac{a_z}{g}\right)$, $\theta_{acc,r} = \arcsin\left(\frac{a_y}{g}\right)$, in which g is the gravitational acceleration, $\theta_{acc,p}$ and $\theta_{acc,r}$ are the pitch and roll angles and a_z and a_y are the accelerations

measured by the accelerometer using its z- and y-axis. Accelerometers do however usually suffer from high frequency noise, so a low pass filter should be used to remove this and only obtain the accelerations due to gravity. Because of the filter a delay is obtained when measuring an angle using accelerometers, making them only useful for measuring the low frequency changes.

Since the gyroscope is mostly useful for the high and the accelerometer for the low frequency content, both signals can be combined using a complementary filter [32], which acts as a low-pass filter for the accelerometer data and as an integrator and high-pass filter for the gyroscope data. This approach cannot be used for the yaw angle since its axis of rotation is aligned with gravity, so only the gyroscopic data can be used in that case. It was decided to use the second order complementary filter in order to increase the cut-off and get rid of as much parasitic data as possible, giving the following equation to calculate the angles:

$$\theta_i = \frac{2\xi_i\omega_{0,i}\, s + \omega_{0,i}^2}{s^2 + 2\xi_i\omega_{0,i}\, s + \omega_{0,i}^2}\theta_{acc,i} + \frac{s^2}{s^2 + 2\xi_i\omega_{i,0}\, s + \omega_{i,0}^2}\,\frac{1}{s}\omega_i, \qquad i = p, r \tag{3.1}$$

In which the subscripts p and r represent the pitch and roll angles, $\theta_{acc,i}$ the angles calculated using the accelerometer data, ω_i the angular velocities measured using the gyroscopes, $\omega_{0,i}$ the cut-off frequencies and ξ_i the damping factors. A flat frequency response will be obtained for $\xi_i \geq 1$ and the value can be increased to obtain a smoother response.

3.1.2. New algorithm

The initial algorithm showed good performance when tilting the unicycle on its place, but when driving forwards and backwards without rotating it, the algorithm also showed a change in angle. This was because the accelerometer also picked up the horizontal acceleration due to the motion of the unicycle and the algorithm was unable to distinguish between this and the acceleration due to gravity. Since the accelerometer is located in the center of the unicycle, it will not only measure gravity, but also the horizontal driving acceleration and the rotational accelerations, so these will affect the calculated angle during motion.

Since the horizontal driving acceleration can be obtained using the gyroscope and the encoder from the motor of the drive wheel ($a_{lon} = r(\ddot{\phi}_{dw} + \dot{\omega}_p)$ (see figure E.2)) and the rotational accelerations can be obtained from the gyroscopes ($\alpha_p = L_p\dot{\omega}_p$, $\alpha_r = L_r\dot{\omega}_r$, in which L_p and

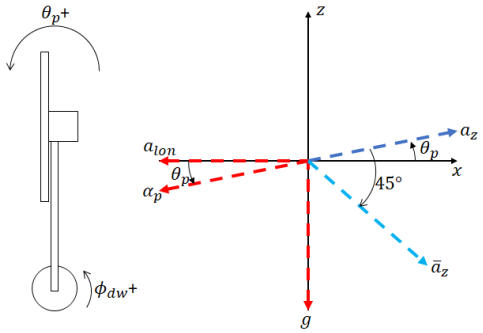


Figure 3.2.: Orientation of the measurement axis a_z of the accelerometer and the accelerations g, a_{lon} and α_p being measured in the longitudinal direction. After changing the orientation of the accelerometer by 45°, the measurement axis $\bar{a}_z$ will be obtained.

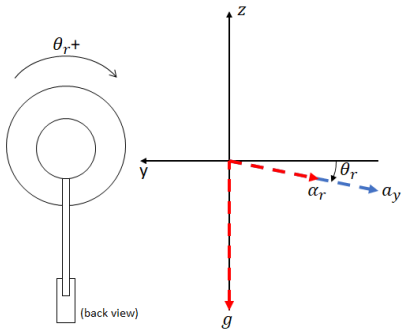


Figure 3.3.: Orientation of the measurement axis a_y of the accelerometer and the accelerations g and α_r being measured in the lateral direction.

L_r are the distance between the points of rotation and the sensor), a new algorithm can be created using figures 3.2 and 3.3. The driving acceleration and gravity have a fixed direction, while the rotational acceleration and the measurement axes of the accelerometer depend on the angle of rotation, so for the longitudinal direction the accelerometer (z-axis) will measure the following values: $a_z = -g\sin(\theta_p) - a_{lon}\cos(\theta_p) - \alpha_p$. Since the only unknown is the angle (θ_p), the equation can be solved, giving:

$$\theta_p = 2\arctan\left(\frac{g \pm \sqrt{|a_{lon}^2 + g^2 - (\alpha_p + a_z)^2|}}{a_{lon} - \alpha_p - a_z}\right), \qquad a_{lon} \neq \alpha_p + a_z \tag{3.2}$$

$$\theta_p = 0, \qquad a_{lon} = \alpha_p + a_z \tag{3.3}$$

In here $\pm\sqrt{|...|}$ represents the signed square root. Since the first equation causes the denominator to approach zero when the angle approaches zero, a switch is required and the algorithm requires quite a lot of computational power. To improve this, the accelerometer can be oriented differently, placing the z-axes at a 45° angle from the horizontal axes ($\bar{a}_z$ in figure 3.2). The measured accelerations will then change and the formula to calculate the angle will become:

$$\bar{a}_z = -g\sin\left(\theta_p - \frac{\pi}{4}\right) - a_{lon}\cos\left(\theta_p - \frac{\pi}{4}\right) - \frac{1}{2}\sqrt{2}\alpha_p \tag{3.4}$$

$$\theta_p = -2\arctan\left(\frac{-a_{lon} \pm \sqrt{|a_{lon}^2 + g^2 - \frac{1}{2}\alpha_p^2 - \bar{a}_z^2 - \sqrt{2}\alpha_p\bar{a}_z|}}{\frac{1}{2}\sqrt{2}\alpha_p + g + \bar{a}_z}\right) + \frac{3\pi}{4}, \qquad \frac{1}{2}\sqrt{2}\alpha_p + g + \bar{a}_z \neq 0 \tag{3.5}$$

In this equation the denominator will no longer become zero in the range of motion of the unicycle and the performance of the algorithm is much better.

Since the lateral direction is not affected by the driving acceleration, only the effect of α_r should be removed. Since α_r will always be aligned with the y-axis of the accelerometer, the rotational acceleration can simply be subtracted from the measured acceleration, after which the initial algorithm can be used (assuming small angles): $\theta_r = \arcsin\left(\frac{a_y - \alpha_r}{g}\right)$.

The performance of the old and new algorithm have been compared experimentally by comparing their outputs under various circumstances, giving the results shown in figures 3.4 and 3.5.

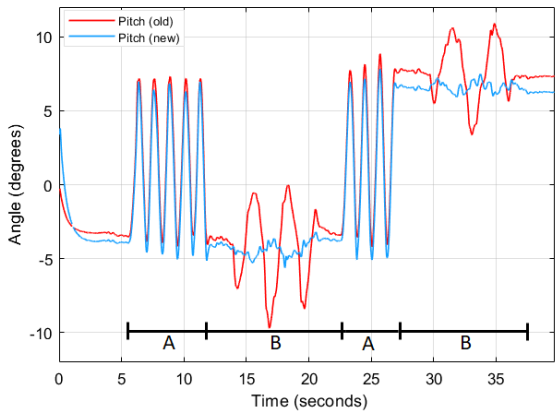


Figure 3.4.: Performance of the algorithms used to calculate the pitch angle, showing the calculated angle when changing the angle during standstill (A) and while driving forwards and backwards with a constant angle (B).

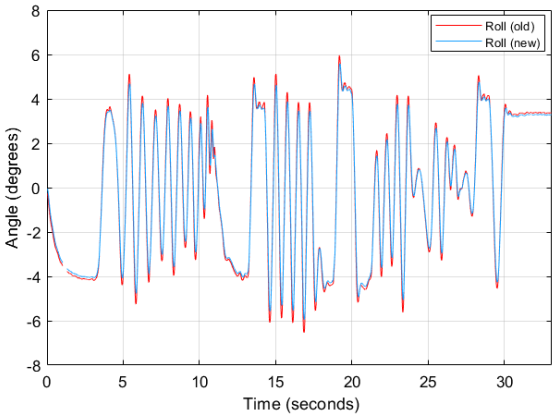


Figure 3.5.: Performance of the algorithms used to calculate the roll angle, showing the calculated angle for quick and slow changes.

For the pitch angle the new algorithm no longer shows a large influence from the driving motion and gives a much better reading of the actual angle. For the roll angle the difference between the algorithms is not very large, but does show that the amplitude is a bit larger when using the old algorithm, probably because the angular motion gives the accelerometer some additional acceleration, which caused the old algorithm to overestimate the angle.

3.2. Testing Rig

In order to be able to test and tune the designed controllers without the unicycle falling over, a testing rig should be used. To allow for individual testing and tuning of the controllers as well as combined testing, the rig should be adjustable. The motions in the lateral and longitudinal directions of the unicycle should be isolated such that they can be tuned individually. The unicycle should still not fall over during these isolated tests though.

To achieve this, a set of additional wheels has been used and two metal frames were designed, which can be seen in figure 3.6. The frames are fixed to the unicycle using slots such that their height is adjustable. To isolate the motions of both directions, also the height of the wheels used in the lateral direction can be adjusted relative to the frames.

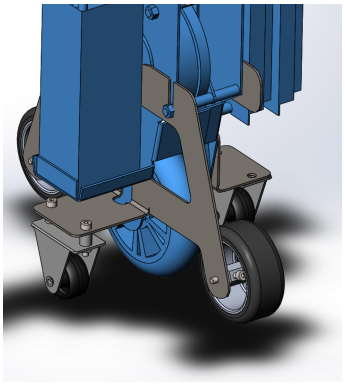


Figure 3.6.: Side wheels fixed to the unicycle.

3.3. Updated Design

The updated design of the Moment Exchange Unicycle Robot with the new sensor block and the side wheels can be seen in figure 3.7 and the actual robot is shown in figure 3.8. A list of the design parameters can be found in appendix A, the wiring schematics showing how all electronics are connected can be found in appendix B and the Simulink real-time files showing how the measured signals are processed and the unicycle is controlled can be found in appendix C.

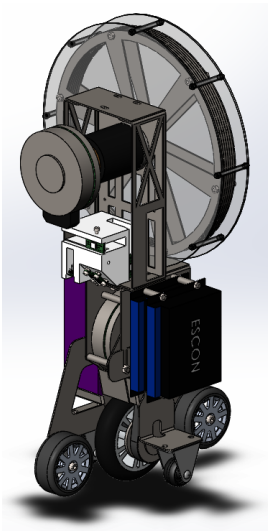


Figure 3.7.: Updated unicycle design, with the new sensor block shown in white.

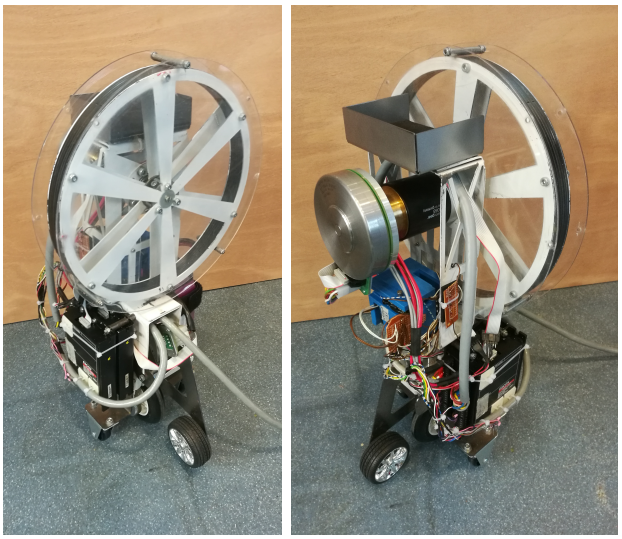


Figure 3.8.: Realized design, including a tray attached on the top for carrying loads.

4. Controller Design

As discussed in chapter 2, previous studies have shown that a couple of different methods can be used to control moment exchange unicycle robots successfully. Using the axis definitions as shown in figure 2.3 on page 3, for the lateral direction the control methods are SMC, PID control and H_∞ loop-shaping, while for the longitudinal direction these are LQR, SMC and PID control. A more detailed explanation of how these controllers work and can be tuned is given in appendix D. To investigate which of these controllers give the best performance, for each type a controller is tuned using the parameters of the robot as given in appendix A and simulations are performed using the (nonlinear) mathematical models derived in appendix E.

4.1. PD & PID

The PD and PID controllers have been tuned using the approach explained in appendix D.2 using the formulas derived in [33]:

$$C_{\text{PID}}(s) = k_p \frac{(s\tau_z + 1)(s\tau_i + 1)}{s\tau_i(s\tau_p + 1)}, \qquad C_{\text{PD}}(s) = k_p \frac{(s\tau_z + 1)}{(s\tau_p + 1)} \tag{4.1}$$

$$\tau_z = \frac{\sqrt{\frac{1}{\alpha}}}{\omega_c}, \qquad \tau_i = \beta\tau_z, \qquad \tau_p = \frac{1}{\omega_c\sqrt{\frac{1}{\alpha}}} \tag{4.2}$$

$$k_p = \frac{m_{eq}\omega_c^2}{\sqrt{\frac{1}{\alpha}}}, \qquad \beta = 2, \qquad \alpha = 0.1 \tag{4.3}$$

Using these equations, the equivalent mass m_{eq} can be obtained from the model parameters and the cross-over frequency ω_c is used to tune the controller values. To find the cross-over frequencies, pole placement is used, placing the zeros of the controllers ($1/\tau_z$) at the stable poles of the system. Values of $\alpha = 0.1$ and $\beta = 2$ were chosen, so the obtained values for τ_z can then be used to find ω_c and determine the other parameters of the controllers.

To find the poles of the systems, the parameters and mathematical models of the systems are used, giving the pole-zero maps shown in figure 4.1. The figure shows that both models have a single real pole located in the right half plane, making the systems unstable. The obtained cross-over frequencies using this approach are: $\omega_{c,lat} = 17.5$ rad/s and $\omega_{c,lon} = 32.3$ rad/s.

To test the stability of the controllers, the Nyquist criterion is used, as explained in appendix D.1. The pole-zero maps of the open loop controlled systems can be seen in figure 4.2. As expected, the plot shows an additional pole and zero for the PID controller while the other poles and zero overlap. For both the lateral and longitudinal direction the open-loop systems still have one unstable pole. The Nyquist plot of the systems, which can be seen in figure 4.3, shows that each of the open-loop systems encircle the -1 point once in the counter-clockwise direction, so all controllers will stabilize their closed-loop systems.

4.2. SMC

For the sliding mode controllers, a switching gain ρ and the tracking error e are used and the controller can be tuned using the equations for the the sliding surface σ and the input signal

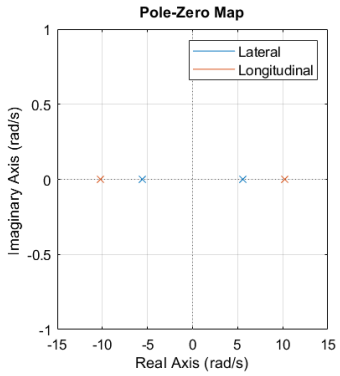


Figure 4.1.: Pole-zero map of the lateral and longitudinal directions of the unicycle.

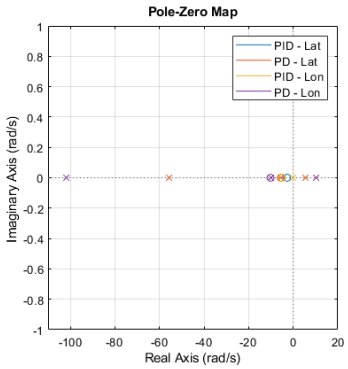


Figure 4.2.: Pole-zero map of the open-loop controlled systems.

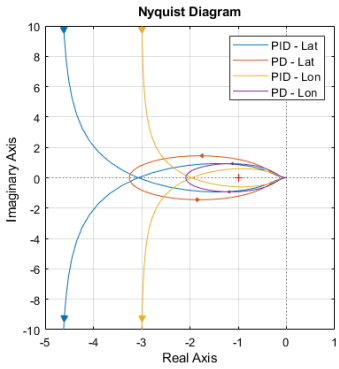


Figure 4.3.: Nyquist plot of the open-loop controlled systems.

$u(t)$ as derived in appendix D.3:

$$\sigma = \dot{e} + \omega e \tag{4.4}$$

$$u(t) = \rho \frac{\sigma}{|\sigma| + \epsilon} + k_d \dot{e} + k_p e \tag{4.5}$$

$$k_d = 2\omega, \qquad k_p = \omega^2, \qquad \omega = \omega_c \tag{4.6}$$

In which ω determines the bandwidth of the system and ϵ determines the smoothness of the switching function. To tune the parameters of the controller, the cross-over frequencies derived for the P(I)D controllers can be used here for ω as well. The value of ϵ will have to be tuned heuristically and will determine the smoothness of the control input at the cost of response time. The value of the parameter ρ is rather important, since asymptotic stability is ensured as long as ρ is larger than the amplitude of the disturbances.

The focus of this thesis is to investigate whether the unicycle can not only stabilize and track a reference, for which it will need to handle general disturbances like model uncertainties and outside influences, but can also handle additional disturbances like extra loads and sloped surfaces. The value of ρ should therefore be relatively large in order to be able to compensate for these unknowns. A value of $\rho = 1$ seems more than sufficient, since disturbances as large as 1 radian probably will not occur. To have reasonably smooth control action but still a fast response, it was decided to choose a value of $\epsilon = 0.1$. For the lateral direction the control parameters will therefore be: $\omega_{lat} = 17.5$ rad/s, $\rho_{lat} = 1$ and $\epsilon_{lat} = 0.1$, and for the longitudinal directions these will be: $\omega_{lon} = 32.3$ rad/s, $\rho_{lon} = 1$ and $\epsilon_{lon} = 0.1$.

4.3. LQR

The LQR controller for the longitudinal direction was tuned using the approach explained in appendix D.4, using the state space matrices (A and B) derived in appendix E:

$$K(t) = -R^{-1}B^T(t)S(t) \tag{4.7}$$

$$[S, CL_{eig}] = care(A, B, Q, R) \tag{4.8}$$

In which the Matlab function *care* solves the algebraic Ricatti equation in order to obtain the state feedback controller K, using the weighting matrices Q and R for the states of the system

and the control input respectively. The states of the system are equal to: $[\theta_p \quad \dot{\theta}_p \quad \phi_{dw} \quad \dot{\phi}_{dw}]^T$. The most important states are the angle of the unicycle θ_p and the angle of its drive wheel ϕ_{dw}. These will determine whether the unicycle will remain upright and on its place. Since the stability is more important than the position of the robot, the pitch angle will get the highest weighting factor. The angular velocity states are less important and therefore get lower weighting factors. The height of the control input is also not too important and therefore also gets a lower weighting factor. After heuristically tuning the matrices, the following were found to give good performance in simulations (see section 4.5):

$$Q = \begin{bmatrix} 0.8 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.05 \end{bmatrix}, \qquad R = 0.01 \tag{4.9}$$

Using these weighting factors, the obtained control matrix K and the closed-loop eigenvalues are approximately:

$$K = \begin{bmatrix} 207.83 & 42.12 & 4.47 & 4.07 \end{bmatrix}, \qquad CL_{eig} = \begin{bmatrix} -588.24 \\ -5.16 \\ -4.59 \\ -2.00 \end{bmatrix} \tag{4.10}$$

As can be seen, all of the closed-loop eigenvalues are negative, so the system will be stable.

4.4. H_∞ Loop-shaping

The H_∞ loop-shaping controller for the lateral direction was tuned using the approach explained in appendix D.5. First the weighted plant is found using: $G_s = augw(G, W_1, W_2, 1)$, in which W_1 and W_2 are weighting factors applied to the control and error signals respectively. Then the H_∞ norm is used to obtain the controller (K_∞) and closed-loop transfer function (CL): $[K_\infty, CL] = hinfsyn(G_s)$. Finally, the weights are shifted from the plant to the controller in order to obtain the feedback controller: $K = W_1 K_\infty W_2$.

The lateral state-space model derived in appendix E was converted to a transfer function to obtain the plant G and the weighting factors W_1 and W_2 were tuned heuristically, obtaining good performance for $W_1 = W_2 = 0.1$. Computing the poles of the obtained closed-loop transfer function gave only negative poles, so the system is stable. Strangely, good performance was only obtained when applying different weighting factors to the system in the simulation, using $W_1 = W_2 = 10$ instead of the values $W_1 = W_2 = 0.1$ used to synthesise the controller. Using this controller did give the performance as expected, stabilizing the system similarly as the SMC, PD and PID controllers.

4.5. Offset Compensation

Since the sensors of the unicycle cannot be perfectly calibrated, there will be a small offset present between the measured and actual angle, so a compensation for this is necessary. Without compensation, the controller will not stabilize the unicycle at its equilibrium position, so a constant torque is required to keep it at this angle. This will make the unicycle drive away and causes the drive and reaction wheels to be accelerated constantly, until they reach their maximum speed, after which they will no longer be able to supply the required torque and the unicycle will fall over.

Of the designed controllers, only the LQR uses all states of the system and therefore already has compensation built in. The other controllers only look at the pitch and roll angles, so a second control loop is required. The cascaded control structure as introduced in appendix D.6 can be used for this. The controllers of the different directions will have similar responses, so the same outer loop control can be used for the different controllers of the same direction.

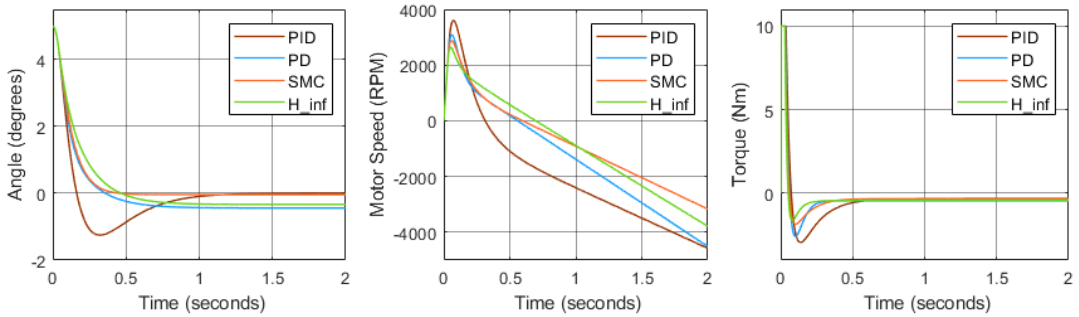
4.5.1. Lateral direction

The response of the controlled systems without offset compensation for an initial angle of 4° and a sensor offset of +1° can be seen in figure 4.4, showing that the reaction wheel will keep on accelerating and will reach the maximum speed of the motor (3190 RPM) rather quickly, after which the actual robot would fall over.

The offset compensation can be done by using a feedback controller on the angular velocity of the reaction wheel, but to obtain this velocity the derivative of the encoder signal needs to be taken, which will cause amplification of noise. A method for which this will not be the case is by using the control output (torque) instead [34]. If the torque is used to compensate for the offset, the controller will be made to stop feeding a torque into the system, stopping the wheel from being accelerated, after which damping in the system will decelerate it.

Since the additional controller needs to compensate for an offset, an integral controller with low-pass filter can be used similarly as done by [34]. The low-pass filter will make sure that the controller will not react and limit the torques used to stabilize the unicycle and the integral action will only compensate for the offset in angle. A small proportional action can be added to speed up the response of the controller. To tune the controller a heuristic method is used. Increasing the integral action makes the unicycle settle faster at the correct angle, but does also respond more aggressive on the stabilization action, which cannot be filtered out completely by the low-pass filter, causing the unicycle to overshoot more during the stabilization.

The response using the additional controller for an initial angle of 4° and a sensor offset of +1°, with some damping added to the simulation, can be seen in figure 4.5. Compared to the simulations without offset compensation in which none of the controllers settle at the correct angle, they now all do, but they do take more time to settle. Initially the settling time was less 0.5 seconds for the SMC and about 1 second for the PID, PD and H_∞ controllers, while they now all take about 3 seconds to settle. The torques into the system return back to zero in about 2 seconds, after which the damping slowly decelerates the motor and the reaction wheel.



(a) Roll angle. (b) Motor speed. (c) Torque.

Figure 4.4.: Performance of the lateral controllers without offset compensation for an initial angle of 4° and a sensor offset of +1°.

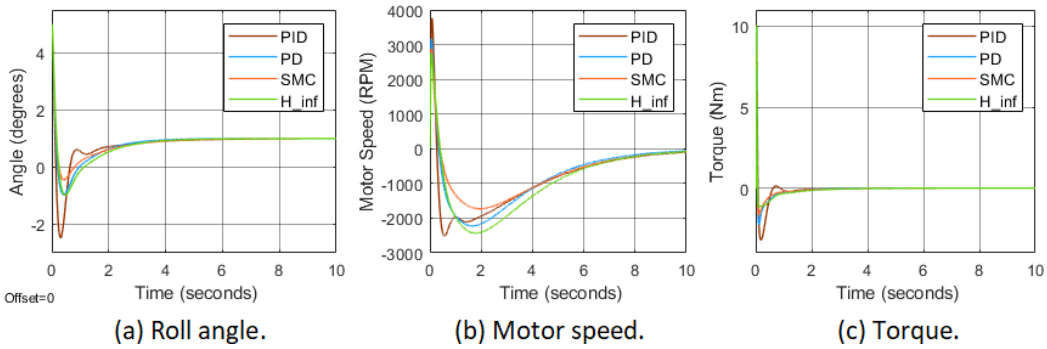


Figure 4.5.: Performance of the lateral controllers with offset compensation for an initial angle of 4° and a sensor offset of +1°.

4.5.2. Longitudinal direction

The response of the controlled system without offset compensation for an initial angle of 4° and a sensor offset of +1° can be seen in figure 4.6. The simulation shows that the LQR controller is the only controller which already has offset compensation build in and therefore already settles at the correct angle. Because of this the LQR controller is also the only one for which the distance settles, although here still an offset is present. The other controllers will keep on accelerating the drive wheel, making it drive away rather fast. Since the position of the unicycle should be controlled in order to make it follow a certain reference, the additional outer loop controller can be used for the offset compensation as well as reference tracking.

The LQR controller already has position control, but produces a steady state error. To remove this error, an additional loop can be added to the drive wheel angle state of the system. For the additional control a PID controller can be used, which compensates for the distance offset. To tune this controller, the same approach is uses as for the stabilization PID controller, but a lower cross-over frequency is used to prevent interference with the stabilization. Using a heuristic method, good performance was obtained by using a frequency equal to half of the cross-over frequency of the stabilization controller.

For the offset compensation and position control of the SMC, PD and PID controllers a cascaded control structure can be used. The outer loop can use drive wheel angle to control the position and update the pitch angle reference such that the inner loop will make the unicycle

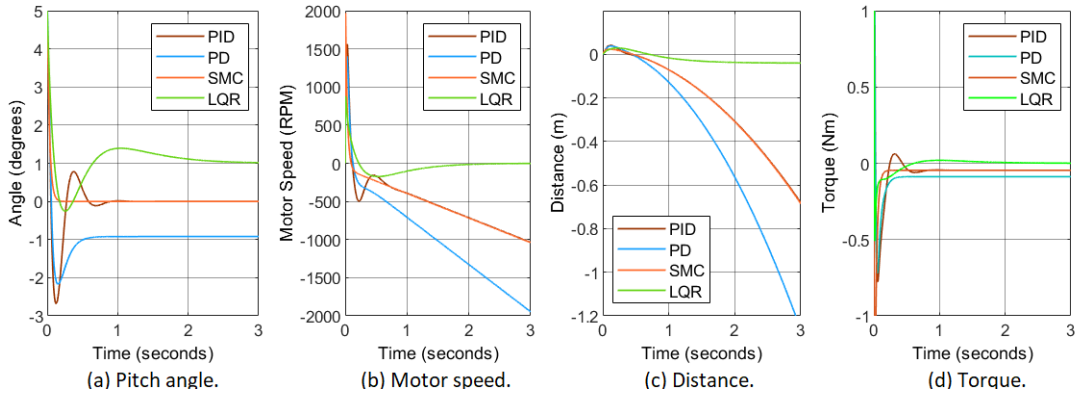


Figure 4.6.: Performance of the longitudinal controllers without offset compensation for an initial angle of 4° and a sensor offset of +1°.

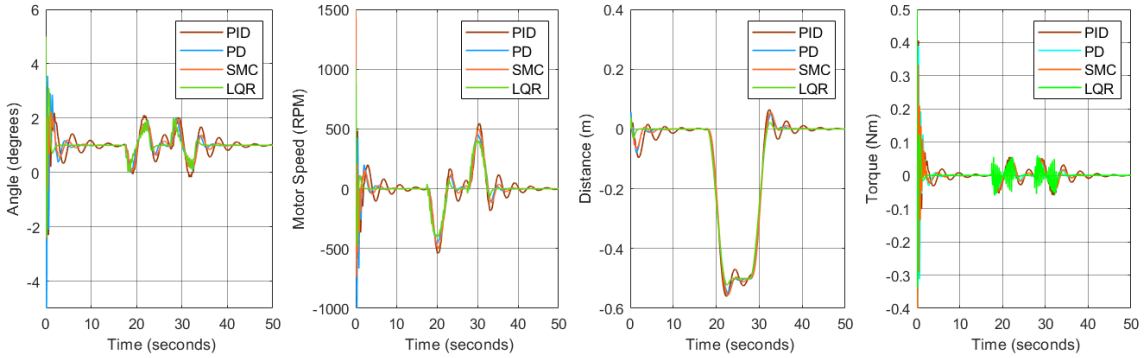


Figure 4.7.: Performance of the longitudinal controllers with offset compensation for an initial angle of 4° and a sensor offset of +1°. After 17.5 and 27.5 seconds skew sine reference profiles are applied for the drive wheel angle.

move accordingly. A PID controller with low-pass filter can be used, similarly as the PI controller with low-pass used for the lateral outer loop, with the derivative action added in order to obtain smoother motion of the unicycle. The controller can also be tuned using a heuristic method and it was found that for each of the controller types optimal performance was found for different parameters, while the same parameters could be used in the lateral direction for the different types of controller.

The performance of the controllers including offset compensation and distance control for an initial angle of 4°, a sensor offset of +1° and a reference signal consisting of two skew sines can be seen in figure 4.7. The simulation shows that the PD and PID controller have a rather high overshoot of more than 4°, while the overshoot of the LQR and SMC is only half that. The LQR controller settles fastest with about 3 seconds, while the SMC and PD controllers take about 8 seconds to settle and the PID controller takes about 15 seconds. When the skew sine references are applied for the drive wheel angle, the controllers show similar performance, following the reference with a slight overshoot. The LQR controller does however give a very spiky torque input signal when tracking the reference.

4.6. Evaluation of the Controllers

As can be seen in figures 4.5 and 4.7, the designed controllers are able to stabilize the unicycle, even for an initial angle of 4° and an offset of +1° in the measured angle. The longitudinal controllers also show good tracking of a reference for the position of the unicycle. In this section the controllers will be compared into detail and a decision will be made on which controllers will be implemented in the actual robot. In order to do this, additional simulations are performed in order to test the robustness of the controllers for parameter uncertainties.

4.6.1. Lateral

To test the robustness of the lateral controllers, a load of 3 kg is added to the unicycle at the height of the reaction wheel, giving the simulation shown in figure 4.8. It can be seen that the PD, the H_∞ and especially the PID controller are starting to show oscillations and will probably become unstable when the load is increased much more. When increasing the load in steps of 1 kg, the PID controller becomes unstable at 4 kg, the PD controller at 6 kg, the H_∞ controller at 7 kg and the SMC at 10 kg. In reality they will become unstable much faster since the maximum

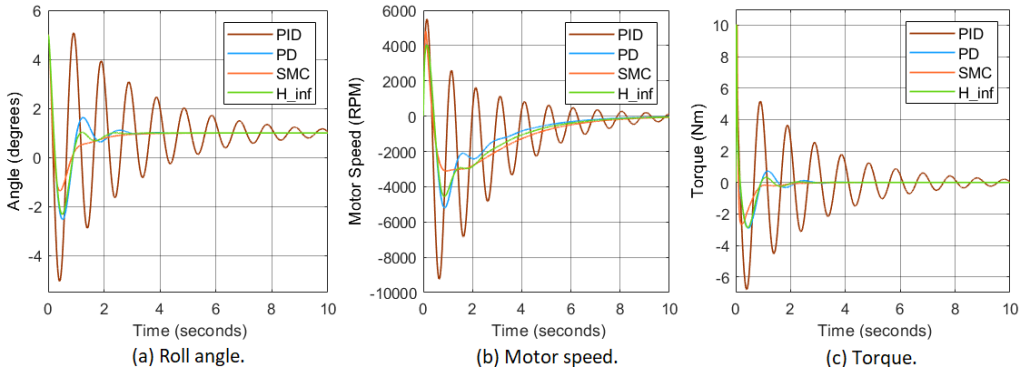


Figure 4.8.: Performance of the lateral controllers with offset compensation for an initial angle of 4 degrees, a sensor offset of +1 degree and an additional load of 3 kg.

motor speed has already been reached at a load of 3 kg, but the test does confirm that the SMC is indeed much more robustly stable regarding the parameters of the unicycle and should be able to handle additional loads much better than the other controllers. Since it also showed the best performance when no load was added, with a fast settling time without a large overshoot, the SMC will be implemented for the lateral direction.

4.6.2. Longitudinal

To test the robustness of the longitudinal controllers, a load of 4 kg is added to the unicycle, giving the simulation shown in figure 4.9. The PD controller already shows large oscillations for this load and will probably become unstable first. When increasing the load, the PD controller becomes unstable for a load of 7 kg and the PID controller for a load of 11 kg. The LQR and SMC could both handle additional weights of more than 100 kg, so both show a large robustness for model uncertainties. Comparing the SMC and LQR, the LQR has a faster settling time but does create very spiky torque signals and is much harder to tune. Because the SMC has comparable performance, but is much easier to tune and has a smoother torque signal, it has been chosen to be implemented for the longitudinal control of the unicycle robot.

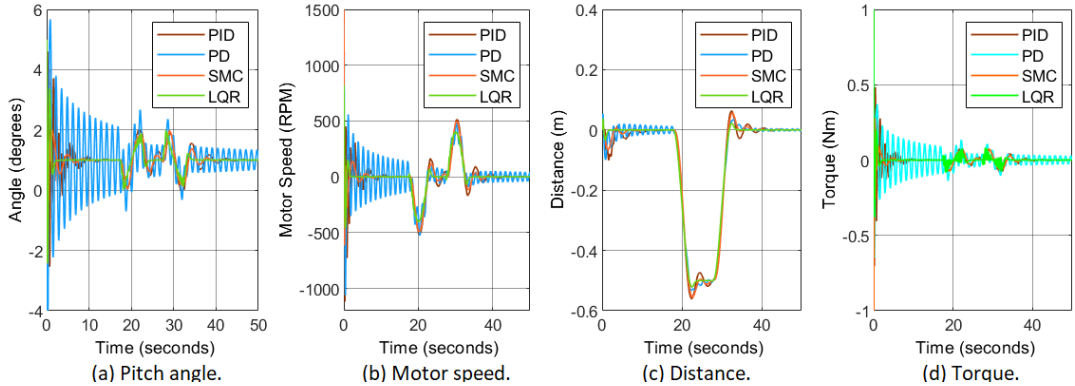


Figure 4.9.: Performance of the longitudinal controllers with offset compensation for an initial angle of 4 degrees, a sensor offset of +1 degree and an additional load of 4 kg. After 17.5 and 27.5 seconds skew sine reference profiles are applied for the drive wheel angle.

5. Implementation & Experimental Results

The controllers designed in chapter 4 can now be implemented on the actual robot. To test and optimize the controllers, the side wheels can be attached such that the motions in the lateral and longitudinal directions can be tested separately, using the axis definitions as shown in figure 2.3 on page 3. Then for both the lateral and longitudinal directions first the sliding mode controllers without offset compensation are implemented and optimized. Since there will be a small offset between the measured angles and the actual values, the controllers will continue accelerating the reaction and drive wheel, but that does not influence the tuning of the stabilization controllers. Next the offset compensation is added and tuned such that the unicycle is actually stabilized in its equilibrium position and will stay on its place.

After the controllers for the lateral and longitudinal directions have been tuned individually, they are combined to test the influence of the coupling between the two directions. Next the tracking performance is tested to see how well the robot follows a reference. Finally, to answer the research question of this thesis, the ability of the robot to carry additional loads and to drive over sloped surfaces is tested.

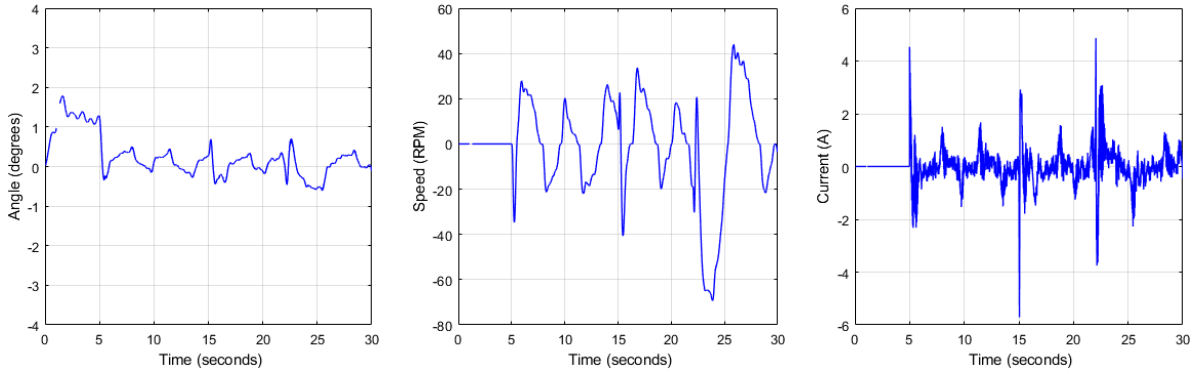
5.1. Lateral Stability

The tests for the lateral controller (without offset compensation) showed that for a too high bandwidth the robot responds too aggressively, causing it to overshoot too much, while for a too low bandwidth it does not respond aggressively enough, making it unable to move towards the equilibrium at all. Good performance was obtained for $\omega_{lat} = 12$ rad/s, $\rho_{lat} = 1$, $\epsilon_{lat} = 0.1$, $k_{d,lat} = 2\omega_{lat}$ and $k_{p,lat} = \omega_{lat}^2$.

To compensate for any offset present in the measured angle and to limit the speed of the reaction wheel, the PI controller with low-pass filter was implemented as well and tests were performed to tune this controller. The tests showed that making this controller too aggressive limits the performance of the stabilizing controller, while if not aggressive enough the reaction wheel will still reach its maximum speed. Good performance was obtained for $K_{p,RW} = 0.25$, $K_{i,RW} = 1$ and $f_{c,RW} = 1$ Hz.

The performance of the tuned lateral controller can be seen in figure 5.1, showing that it can stabilize the robot when released close to its equilibrium position without having to accelerate the reaction wheel to its maximum speed (211 RPM). When triangular disturbances were applied as inputs to the system, giving a small pulse with a magnitude of 6 Nm to the reaction wheel and the unicycle, the controller can easily keep the roll angle close to its equilibrium. Tests showed that larger disturbances could also be compensated for, but the timing of those disturbances did have an influence, giving difficulties when the unicycle was already somewhat away from its equilibrium. Smaller disturbances were sometimes not even noticed. When pushing the unicycle by hand, the controller also reacts directly and instantly tries to push the hand away in order to stay at its equilibrium.

To find the limits of the lateral controller, the maximum initial angle at which it can be released was investigated as well as the effect this has on the settling time. It was found that when tilting the robot to the left the maximum release angle was about 2.4°, while when tilting it to the right the unicycle could be released even when resting on its side wheel at about 5.5°. The settling time also depends on whether the unicycle is tilted to the right or left, but also



(a) Roll angle. (b) Reaction wheel speed. (c) Motor current.

Figure 5.1.: Lateral stability test, where after five seconds the motors are enabled and the unicycle is released. A triangular pulse disturbance with a magnitude of 6 Nm is applied to the input torque of the system negatively at 15 seconds and positively at 22 seconds.

depends on the release angle. The settling time is about one second when released from an angle of 2°, while taking less than half a second when released within 1° of equilibrium. Performing a simulation for an initial angle of 2° with an offset of 0.1° also gives a settling time of about 1 second, so the controller appears to behave as expected.

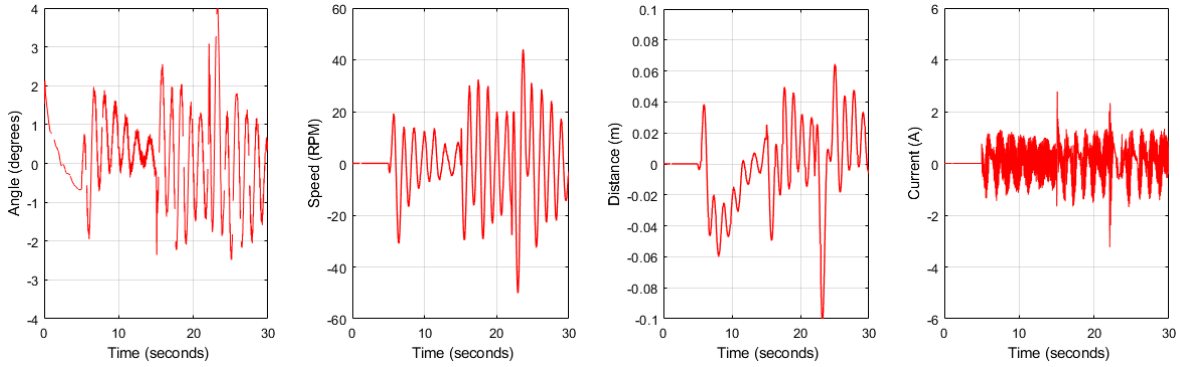
5.2. Longitudinal Stability

The longitudinal tests showed quite some vibrations in the robot, so making the controller less aggressive was necessary in order to prevent amplification of these vibrations. The main source turned out to be noise in the measured angular velocity, which could not be filtered out without adding too much delay in the signal. By lowering the derivative action and the switching amplitude of the sliding mode controller the effect of this noise could be minimized and the best performance was obtained for $\omega_{lon} = 5$ rad/s, $\rho_{lon} = 0.3$, $\epsilon_{lon} = 0.1$, $k_{d,lon} = \omega_{lon}/5$ and $k_{p,lon} = \omega_{lon}^2$.

To compensate for any offset present in the measured angle and to control the position of the unicycle, the PID controller with low-pass filter was implemented as well and tests were performed to tune this controller. Making it too aggressive again showed interfere with the stabilization of the unicycle. Good performance was obtained for $K_{p,DW} = 0.015$, $K_{d,DW} = 0.005$, $K_{i,DW} = 0.002$ and $f_{c,DW} = 5$ Hz.

The performance of the tuned longitudinal controller can be seen in figure 5.2, showing that it can stabilize the robot when released close to its equilibrium position. The controller does require some time to bring the robot back to its initial position, but it can keep the robot stable and on its place despite a small offset in the angle. When applying triangular input pulses as inputs to the system, giving a small pulse with a magnitude of 4 Nm to the drive wheel and the unicycle, the controller can easily keep the pitch angle close to its equilibrium and the position of the unicycle also recovered from the pulse quite fast. Larger disturbances cause the side wheels of the unicycle to hit the ground, but without the side wheels they could probably be compensated for. Also when pushing the unicycle by hand, quite large excitations could be applied, pushing the robot forwards up to ten centimetres did not give much problems.

To find the limits of the longitudinal controller, the maximum initial angle at which it can be released was investigated as well as the effect this has on the settling time. For initial angles



(a) Pitch angle. (b) Drive wheel speed. (c) Distance. (d) Motor current.

Figure 5.2.: Longitudinal stability test, where after five seconds the motors are enabled and the unicycle is released. A triangular pulse disturbance with a magnitude of 4 Nm is applied to the input torque of the system negatively at 15 seconds and positively at 22 seconds.

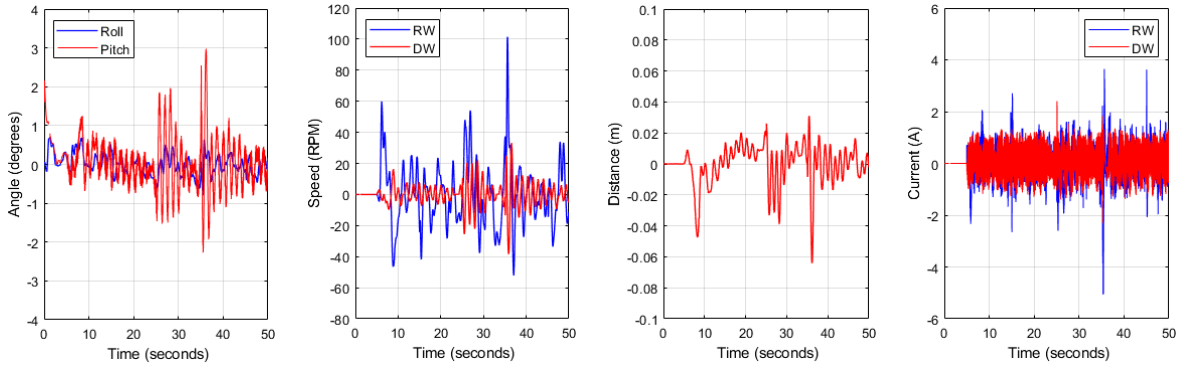
of about 3° in either direction, the settling time for the pitch angle is about two seconds, while the position takes about three to four seconds to get back to its initial value. The response of the unicycle did not depend strongly on the release angle and it showed to be able to handle angles up to 5° without problems and without needing much additional settling time. Larger angles caused the side wheels to hit the ground, but could probably also be compensated for. Performing a simulation with an initial angle of 3° and an offset of 0.1° shows a settling time of about 4 seconds for the pitch angle and about 2 seconds for the position, so probably due to different tuning slightly different results are obtained, but the performance is quite similar and the controller appears to behave as expected.

5.3. Combined Stability

After tuning the controllers individually, they were combined and tested together, giving the performance shown in figure 5.3. The tests show that as long as the unicycle is released close to its equilibrium position, the controllers will not affect each other much and both directions can be stabilized. When the unicycle is released further from its equilibrium, coupling effects started to show more and could cause instability. When disturbances were applied, the coupling between the two directions also showed up, with the unicycle rotating a little when compensating, but overall the orientation of the unicycle did not change much. The magnitude of the disturbances could not be as large as for the isolated directions. When larger amplitudes were applied, the controllers started to influence each other and could grow unstable.

The tests also showed that the cable connecting the robot to the computer had quite a large influence on the yaw angle, especially when having to pull or push the cable while driving. To minimize this effect the cable is laid as freely as possible and has been attached to the center of the robot, limiting the moment it exerts as much as possible.

To find the limits of the controllers working together, the maximum initial angles at which the unicycle can be released and the effect this has on the settling time was investigated. It was found that the maximum release angles depended on each other. When the pitch angle was released close to its equilibrium, the maximum roll angle was about 2.5°, while for larger pitch angles the maximum reduced to about 2°. The pitch angle did not show much influence from the roll angle and the maximum pitch angle was found to be about 4.5°. To consistently



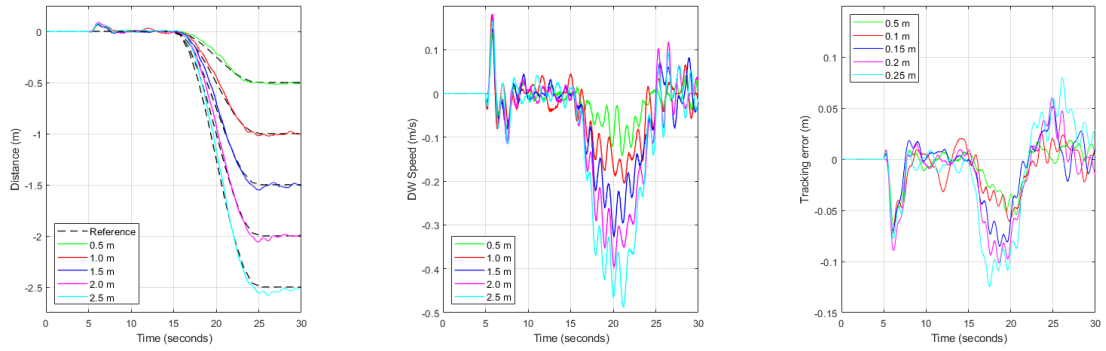
(a) Angles. (b) Wheel speeds. (c) Distance. (d) Motor currents.

Figure 5.3.: Combined stability test, where after five seconds the motors are enabled and the unicycle is released. A triangular pulse disturbance with a magnitude of 1.5 Nm is applied to the longitudinal (DW) input torque of the system negatively at 25 seconds and positively at 35 seconds and a triangular pulse disturbance with a magnitude of 4 Nm is applied to the lateral (RW) input torque negatively at 15 seconds and positively at 45 seconds.

obtain good performance, it was found that the unicycle should be released within about 2° of the equilibrium position for both angles. The settling times for the roll angle depended on the release angle, but were mostly within one second. The pitch angle usually took about three seconds to stabilize while the position took about four to six seconds depending on the pitch angle at which it was released.

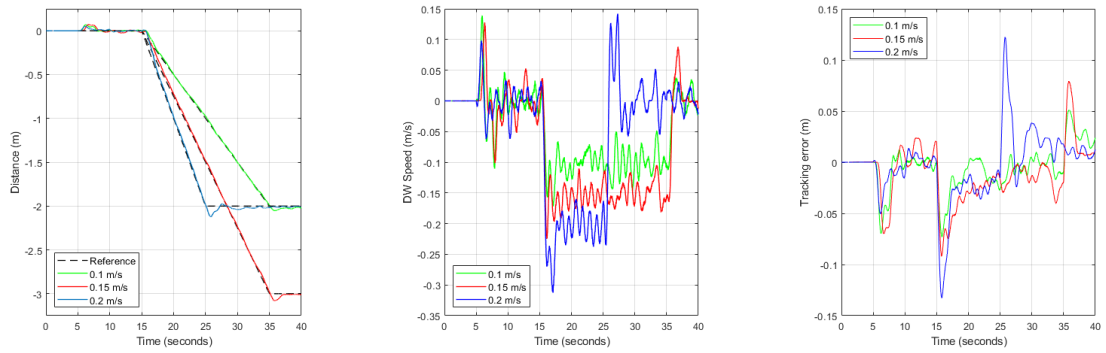
5.4. Following a Reference

To test whether the controllers can also keep the robot stabilized while following a reference, test were performed using skew sine and ramp reference signals of various magnitudes. The skew sine references, which can be seen in figure 5.4, show that the unicycle can perform steps of up to 2.5 meters within 10 seconds. As expected, the velocity shows to follow a sine shape when performing the step. Oscillations are present in the velocity though, which is probably caused by the unicycle prioritizing stabilization over tracking. The position of the unicycle shows quite



(a) Distances. (b) Speeds. (c) Tracking errors.

Figure 5.4.: Response of the unicycle for skew sine inputs of various amplitudes.



(a) Distances. (b) Speeds. (c) Tracking errors.

Figure 5.5.: Response of the unicycle for ramp inputs of various slopes.

accurate tracking, with a small lag behind the reference and some overshoot at the end. The tracking error shows that the lag increases from about 5 cm for the smaller steps to about 12 cm for the largest step, while the overshoot increases from 1 cm to about 8 cm. During the experiments it was observed that the unicycle also became a little less stable when increasing the step size, keeping the deviation of the roll angles within 0.8° for the steps up to two meters, but for the step of 2.5m a deviation of 1.4° was observed. The pitch angles showed a slight increase in deviation for increased step sizes, increasing from 1.5° to 2°, which was required in order to obtain the acceleration necessary to follow the reference.

When following a ramp input with a certain slope the unicycle also showed good tracking, as can be seen in figure 5.5. It initially lags behind a bit, but then overlaps with the reference very well, keeping the tracking error within 4 cm. Again some overshoot was observed when stabilizing after following the ramp, which increases when the unicycle is driving at faster speeds. The velocities show to be stabilizing at the slope of the ramp quite well, but here also some oscillations are obtained. For the different slopes not much difference could be seen for the lateral stabilization, all keeping the deviation of the roll angle within 0.6°. The pitch angle showed relatively large deviations when speeding up at the beginning of the ramp and slowing down at the end of the ramp, requiring angles of 3° for the ramp of 0.2 m/s, but next to these pulses not much deviation could be observed. While following the ramp references, the unicycle drove in a straight line, but did show a little steering in the counter-clockwise direction.

5.5. Load

A delivery drone would have to carry packages of various weights, so the unicycle should be able to compensate for this additional weight without having to change its control parameters, effectively making the additional load an unknown for the system. To test whether it can handle additional loads, a tray was mounted on top of the unicycle and various weights were placed in there, as can be seen in figure 5.6. To prevent the weights from shaking and starting to move around in the tray, they were fixed in place using tape.

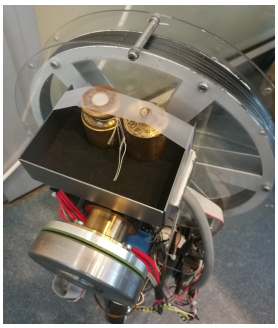
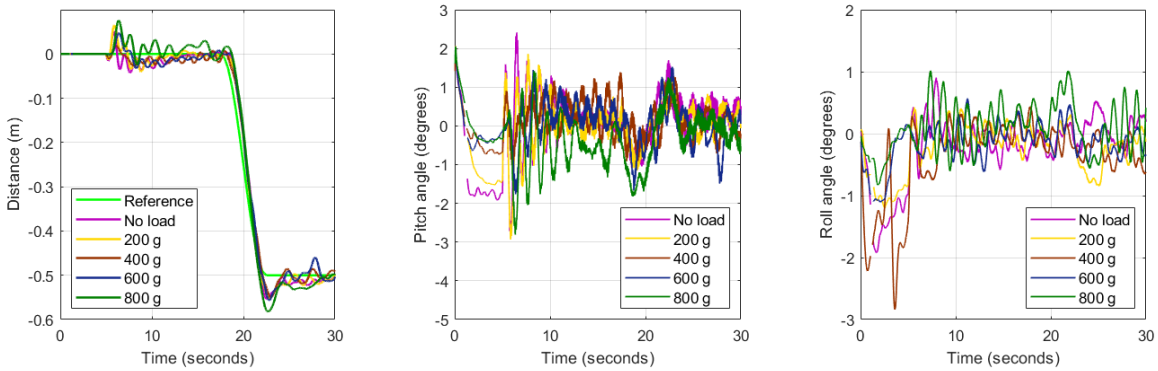


Figure 5.6.: Tray fixed to the unicycle, with weights.

During the experiments the load was increased incrementally and the results can be seen in figure 5.7. The experiments show that for smaller loads the unicycle can easily be released and follow a



(a) Distances. (b) Pitch angles. (c) Roll angles.

Figure 5.7.: Response of the unicycle following a skew sine for various loads.

skew sine. For heavier loads it needs to be released very close to its equilibrium position for it to stabilize. The maximum weight for which the unicycle could stabilize was about 800 grams. With the heavier loads the unicycle also overshoots more and takes longer to settle.

When following the reference without any loads, the unicycle was observed to drive in a straight line, not influencing the yaw angle much. When placing the weights not exactly in the center of the tray (laterally), the unicycle would no longer drive in such a straight line and would start to steer in a certain direction. This effect was however a bit inconsistent and the influence of the cable seemed to be much larger than the effect of the offset.

5.6. Slope

To be able to use the unicycle as a delivery drone, it should be able to drive over uncertain surfaces which could have various slopes. To test its ability to do this, the unicycle was made to drive over a surface of which the slope was increased in steps in order to find the maximum slope that it can stabilize on and drive over. This was done for a surface with a slope in the longitudinal direction and one with a slope in the lateral direction, as can be seen in figure 5.8.

The results for a surface with various slopes in the longitudinal direction can be seen in figure 5.9. To perform the test the side wheels in the longitudinal direction had to be removed and for a slope of 12° the frame to which the side wheels are normally attached also almost

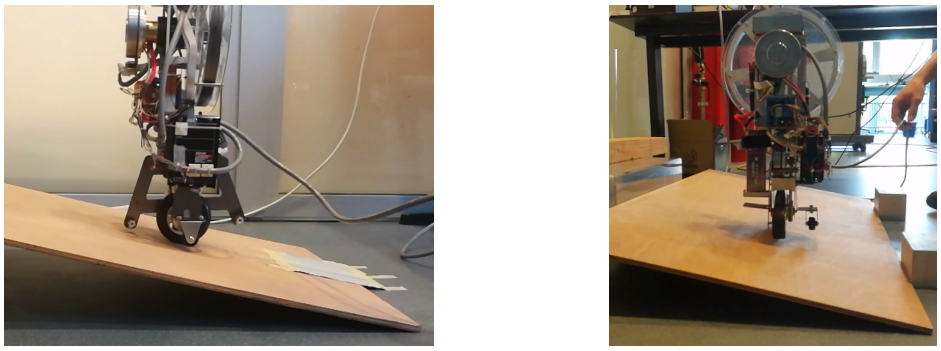
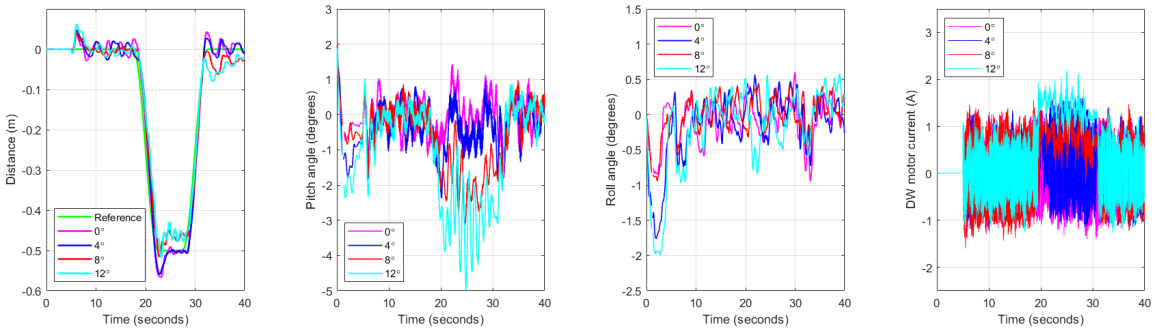


Figure 5.8.: Testing the unicycle a surface with a slope in the longitudinal direction (left) and one with a slope in the lateral direction (right).

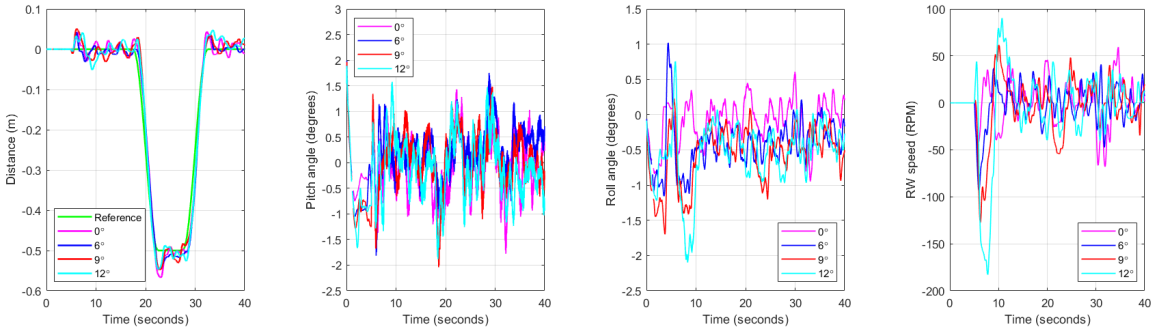


(a) Distances. (b) Pitch angles. (c) Roll angles. (d) DW motor currents.

Figure 5.9.: Response of the unicycle for various longitudinal slopes. After 17.5 and 27.5 seconds skew sine reference profiles are applied for the drive wheel angle.

started to hit the surface. The results show that for relatively low slopes (up to 4°) the unicycle barely notices the slope and has no problems following skew sine reference signals while keeping the unicycle stabilized. For larger slopes a small steady state error is obtained when following the reference. Next to that, the pitch angle shows that the unicycle needs to lean forward in order to be able to stabilize and drive over the surface, while the currents also show a certain bias in order to prevent the unicycle from driving back down due to the pull of gravity. As expected, the lateral stabilization is not influenced by the slope and the roll angle shows similar behaviour regardless of the slope. When following the reference the robot drove in a straight line, so the sloped surface did not seem to have much effect on the yaw angle either.

The results for a surface with various slopes in the lateral direction can be seen in figure 5.10. The figures show that a slope in the lateral direction does not have much influence on the tracking of a reference and that the distance and pitch angles do not change much. The roll angle does show to be stabilized at a different value depending on the slope of the surface and the unicycle shows to have more problems stabilizing when released on a surface with a higher slope. For a slope of 12° the reaction wheel almost reaches its maximum speed and during the experiment the unicycle had a lot of trouble stabilizing, wanting to rotate and drive down the surface. Once stabilized, the unicycle showed not to be bothered much by the surface though and could still drive in a straight line.



(a) Distances. (b) Pitch angles. (c) Roll angles. (d) RW speeds.

Figure 5.10.: Response of the unicycle for various lateral slopes. After 17.5 and 27.5 seconds skew sine reference profiles are applied for the drive wheel angle.

6. Discussion

6.1. Stabilization

The experimental results have shown that the designed sliding mode controllers for the lateral and longitudinal direction are able to stabilize the unicycle. When the unicycle was released within a few degrees of its equilibrium position, not much influence of the coupling between the two direction could be seen. When released further away, often the longitudinal direction did not seem to be bothered much, but the lateral direction could grow unstable. Since this was not the case when testing the lateral direction individually, the oscillatory motion in the longitudinal direction probably somewhat amplifies the oscillations in the lateral direction and at a certain point the controller can no longer compensate for these.

For the lateral direction it was observed that the maximum release angle depended on the direction, so probably the center of mass of the unicycle is not located directly above the drive wheel. Due to this it could be that the profile of the wheel, which has a very small flat region at the bottom, could cause the overshoot of one direction to reach this flat region, while the other direction reaches a round region of the wheel. The flat region probably helps stabilizing the unicycle, making this direction more forgiving for overshoots.

The actual robot has some additional dynamics which were neglected in the model and a small difference also exists between the parameters used for the model and the actual parameters. Because of this the controllers had to be tuned again for the actual robot, but the experiments still showed similar results compared to the simulations. The pitch angle has some additional noise which could not be filtered out, but this is the only large deviation from the simulations. Overall the used mathematical model therefore seems to represent the behaviour of the unicycle good enough to use it as a basis for tuning the controllers and comparing their performance in the simulations.

6.2. Tracking

The tracking performance of the unicycle while following a reference showed quite good performance. The unicycle could track the references with a little overshoot and with quite small tracking errors. Not much oscillations were observed and the overall motion of the unicycle was quite smooth. The velocity of the unicycle while following the references did show some more oscillations, but in general the velocity showed the expected shape. Comparing the tracking performance to the unicycles found in literature, as described in chapter 2, the robot shows much smaller oscillations in the velocity for the same set-points, while the tracking error is also smaller when following a similar reference.

The overall stability of the unicycle tended to decrease for higher speeds, with the pitch angle and the driving motion probably influencing the roll angle due to a coupling present in the system, causing this to become unstable. When driving at the highest velocities, the side wheels could also sometimes hit the floor since the unicycle had to lean forward in order to obtain the desired acceleration. Finally, the cable to which the unicycle is attached to the computer running the controllers also showed a large influence on the stability. The cable is quite stiff and has to be pulled or pushed by the robot, so the instabilities obtained for higher speeds might also have been partially caused by the robot having to pull or push the cable.

6.3. Load, Slope and Steering

The experiments for the unicycle with various loads showed that at a certain point the unicycle had problems stabilizing, while smaller loads did not seem to be much of an issue. This is probably because the unicycle is heavier, so more torque is required in order to move it to its equilibrium position. If the controller parameters are adapted accordingly, the unicycle will probably have less problems carrying the loads. Next to that, it would probably be better if the loads could be placed in line with the center of mass, since they now also cause an additional offset, which the controller has to compensate for, making it take longer to stabilize and causing more coupling. If the loads could be placed lower on the unicycle, it would probably have less problems carrying them, but there is no space for them there and that would make them less accessible.

For the sloped surfaces, the unicycle showed to have not much problems driving over the slopes. For the longitudinal directions it can probably drive over even larger slopes in case the frame of the side wheels is removed, since it did not show any sign of becoming unstable. For the lateral direction this might also be the case, but releasing it on a high slope was rather difficult, since due to a slight coupling it would start to rotate a bit and then drive down the surface. If the unicycle would be released on a flat surface and the slope of this surface would then be increased, it might be able to handle slopes more easily. The main limitation of driving over sloped surfaces is probably the slip that might occur rather than the performance of the controllers.

Unfortunately, the experiments looking into the ability of the unicycle to steer did not give consistent results. The main reason for this was that the effect of the cable was much larger than the effect of leaning due to the high cable stiffness. The experiments suggest that it is possible, but further experiments need to be performed in which the effect of the cable is removed in order to confirm this. Another limitation is the measurement of the yaw angle, which is done by just a gyroscope. The drift in the angle due to the integration of the signal can be minimized by removing the bias of the measured signal, but it was observed that this bias has a rather high dependence on temperature. During a day of testing, the drift would sometimes change quite a bit when the room would heat up or cool down. The gyroscope does however have a temperature sensor on board, so this effect can possibly be removed using this sensor in order to obtain much more accurate measurements. The bias shift is not much of an issue for the pitch and roll measurements, since this is filtered out by using the accelerometer data and the complementary filter.

7. Conclusion

The focus of this project was to answer the following research question: *"Is it feasible to use the Moment Exchange Unicycle Robot as a delivery drone?"* To investigate this, first a literature research was performed, in which it was found that two main questions which should still be answered in order to answer the research question are: *"Can the MEUR be used to carry various loads of unknown weights?"* and *"Can the MEUR be used to drive over sloped surfaces of unknown slope and orientation?"*

To answer these questions, first the existing prototype was evaluated, a new sensor set-up was created and side wheels were added in order to allow for easy testing. The controller types which were found to be working in literature were then compared using simulations and it was found that the sliding mode control structure gave the best performance for both the lateral and longitudinal direction. These controllers were therefore implemented and tuned, showing that they are able to stabilize the actual unicycle robot. Next the performance of the unicycle for tracking a reference was tested and it was found that the unicycle gave better tracking performance compared to the results found in literature.

To answer the research question, experiments were then performed by carrying additional loads and driving over sloped surfaces. The unicycle showed to be able to carry additional loads of up to 0.8 kg and could drive over surfaces with a slope in the longitudinal direction of up to 12°. For the lateral direction also a maximum slope of 12° could be handled.

It can thus be concluded that the unicycle is able to stabilize itself, track a reference, carry additional loads and can drive over sloped surfaces. To actually be able to implement it as a delivery drone, the robot should still be made wireless and steering should be added, but it has already been shown in literature that by adding an additional actuator and reaction wheel, active steering can be obtained, so it can be concluded that it is definitely possible to use the Moment Exchange Unicycle Robot as a deliver drone!

8. Recommendations

The performed experiments have shown that the Moment Exchange Unicycle Robot is able to stabilize itself, track a reference, carry additional loads and drive over uncertain surfaces with various slopes. It is therefore definitely feasible to use it as a delivery drone, but to actually be able to implement it as such, a few things should still be improved.

First of all, the yaw angle measurement can be improved. This angle is currently measured by just a gyroscope, of which the bias needs to be removed in order to prevent drift. Since this bias depends on the temperature, it is difficult to remove it. For the pith and roll angles the accelerometer data and a complementary filter are used for this, but for the yaw angle this is not possible. Usually a magnetometer is used instead, but this is not possible inside, since buildings distort the magnetic field of the earth. One way in which the measurement can probably be improved is by compensating for the bias shift by using the temperature sensor present on the gyroscope.

To improve the stability of the unicycle, the sensor noise should be reduced. Especially in the longitudinal direction the sensor noise causes vibrations in the response of the system. This noise can possibly be reduced by tuning the on-board filter better by using a different capacitor. Otherwise, sensors with less sensitivity to noise should be used. Next to that, the stability can be further improved by decoupling the system, for example by using the mass matrix, removing the coupling effects which cause instability for larger deviations from the equilibrium position.

The side wheels currently used for preventing the robot from falling over during the experiments should probably also be replaced by a better way of preventing damage when falling over. The side wheels were designed for tuning the controllers of the lateral and longitudinal direction individually, but when tracking a reference or driving over sloped surfaces, they are not ideal. They limit the maximum angle of the surfaces over which can be driven and when having to accelerate quickly for a certain reference, they could sometimes hit the ground. Since stability is already obtained and the controllers have been tuned properly, they can probably be replaced by a different protection mechanism which only prevents damage in case the unicycle falls over, but does not limit its motion. This could for example be done by creating some kind of (circular) bumper around the unicycle which would be the only part hitting the ground in case the unicycle might fall over.

Next, the unicycle should also be made wireless. The cable which is currently used to connect it to the computer running the control software is rather stiff and showed quite some influence on the stability and yaw angle of the robot. The cable can be removed by either using an on-board controller or by using a wireless connection between the robot and a computer running the control software. In order to ensure performance, the wireless system should be able to send sufficient data in a short enough time in order to make sure that the controllers can perform their task accurate and fast enough for the unicycle to not loose performance. The wireless solution will probably add some additional weight to the unicycle, but as the experiments have shown, this should not give an issue as long as this weight is not too high. Heavier solutions are probably also not a big problem, but in that case the controllers need to be tuned again and it should be investigated if the reaction wheel can still deliver sufficient torque, since otherwise a heavier reaction wheel might be desirable.

Finally, a means of steering the unicycle robot should still be implemented. Some experiments were already performed looking in the possibility of steering by leaning the unicycle, but due to

the cable the results were inconsistent. When the robot is made wireless, additional experiments need to be performed in order to test the feasibility of steering the unicycle this way. In case this is not possible, an additional actuator and reaction wheel (or turntable) can be added in order to perform the steering instead.

Bibliography

[1] K. Shikar, Transwheel delivery drone concept, http://kobishikar.wixsite.com/jaykob-shikar/transwheel, Accessed on 20-02-2018.

[2] The Kroger Co., Kroger and Nuro Partner to Pilot Autonomous Delivery, 28-06-2018, https://www.prnewswire.com/news-releases/kroger-and-nuro-partner-to-pilot-autonomous-delivery-300673917.html, Accessed on 30-08-2018.

[3] Reuters, Israeli engineer designs grounded drone delivery service, 14-10-2015, https://uk.reuters.com/article/us-israel-deliveryrobert/israeli-engineer-designs-grounded-drone-delivery-service-idUKKCN0S815E20151014, Accessed on 20-02-2018.

[4] R. Giesen, T.H. Hoksbergen, A.A. Pleijsier, R.C. Swiersema and K. Ywema, PD control of a self-balancing Moment Exchange Unicycle Robot based on multiple theoretical models, 2016, Bachelor of Science Thesis, University of Twente.

[5] E. Dannenberg, Redesign & Control of a Moment Exchange Unicycle Robot 'MEUR', 2017, Bachelor of Applied Sciences Thesis, University of Twente.

[6] Segway Inc., Segway PT, www.segway.com, Accessed on 22-02-2018.

[7] Murata Manufactoring Co., Murata Boy, 29/09/2005, https://www.murata.com/en-eu/about/mboymgirl/mboy, Accessed on 22-02-2018.

[8] A. L. Schwab, J. P. Meijaard, and J. M. Papadopoulos. Benchmark results on the linearized equations of motion of an uncontrolled bicycle. *Journal of Mechanical Science and Technology*, 19(1):292–304, Jan 2005. https://doi.org/10.1007/bf02916147.

[9] Patryk Cieslak, Tomasz Buratowski, Tadeusz Uhl, and Mariusz Giergiel. The mono-wheel robot with dynamic stabilisation. *Robotics and Autonomous Systems*, 59(9):611 – 619, 2011. https://doi.org/10.1016/j.robot.2011.05.002.

[10] S. Langius, 3D Modeling and simulation of a moment exchange unicycle robot (WA1333), 16-09-2011, Master of Science Thesis, University of Twente.

[11] Yohanes Daud, Abdullah Al Mamun, and Jian-Xin Xu. Dynamic modeling and characteristics analysis of lateral-pendulum unicycle robot. *Robotica*, 35(3):537–568, 2017. https://doi.org/10.1017/S0263574715000703.

[12] Minh-Quan DAO and Kang-Zhi LIU. Gain-scheduled stabilization control of a unicycle robot. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, 48(4):649–656, 2005. https://doi.org/10.1299/jsmec.48.649.

[13] Surachat Chantarachit and Manukid Parnichkun. Development and control of a unicycle robot with double flywheels. *Mechatronics*, 40:28 – 40, 2016. https://doi.org/10.1016/j.mechatronics.2016.10.011.

[14] H. Jin, T. Wang, F. Yu, Y. Zhu, J. Zhao, and J. Lee. Unicycle robot stabilized by the effect of gyroscopic precession and its control realization based on centrifugal force compensation. *IEEE/ASME Transactions on Mechatronics*, 21(6):2737–2745, Dec 2016. https://doi.org/10.1109/TMECH.2016.2590020.

[15] J. Van Pommeren, The unibot, design and control of a self-balancing unicycle robot (WA1125), 18-12-2007, Master of Science Thesis, University of Twente.

[16] D. Gong, Q. Pan, G. Zuo, and W. Deng. Lqr control for a self-balancing unicycle robot. In *Proceedings of the 10th World Congress on Intelligent Control and Automation*, pages 1424–1429, July 2012. https://doi.org/10.1109/WCICA.2012.6358103.

[17] Shweda Mohan, J.L. Nandagopal, and S. Amritha. Decoupled dynamic control of unicycle robot using integral linear quadratic regulator and sliding mode controller. *Procedia Technology*, 25:84 – 91, 2016. https://doi.org/10.1016/j.protcy.2016.08.084.

[18] Shweda Mohan, JL. Nandagopal, and S. Amritha. Coupled dynamic control of unicycle robot using integral linear quadratic regulator and sliding mode controller. *Materials Today: Proceedings*, 5(1, Part 1):1447 – 1454, 2018. https://doi.org/10.1016/j.matpr.2017.11.232.

[19] Murata Manufactoring Co., Murata Girl, 23/09/2008, https://www.murata.com/en-eu/about/mboymgirl/mgirl, Accessed on 22-02-2018.

[20] Y. Rizal, C. T. Ke, and M. T. Ho. Point-to-point motion control of a unicycle robot: Design, implementation, and validation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4379–4384, May 2015. http://dx.doi.org/10.1109/ICRA.2015.7139804.

[21] Y. Isomi and S. Majima. Tracking control method for an underactuated unicycle robot using an equilibrium state. In *2009 IEEE International Conference on Control and Automation*, pages 1844–1849, Dec 2009. https://doi.org/10.1109/ICCA.2009.5410172.

[22] Lee Jae-Oh, Han In-Woo, and Lee Jang-Myung. Fuzzy sliding mode control of unicycle robot. In *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 521–524, Nov 2011. https://doi.org/10.1109/URAI.2011.6145875.

[23] Yan Li, Jae-Oh Lee, and Jangmyung Lee. Attitude control of the unicycle robot using fuzzy-sliding mode control. In Chun-Yi Su, Subhash Rakheja, and Honghai Liu, editors, *Intelligent Robotics and Applications*, pages 62–72, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-33503-7_7.

[24] J. Lee, S. Han, and J. Lee. Decoupled dynamic control for pitch and roll axes of the unicycle robot. *IEEE Transactions on Industrial Electronics*, 60(9):3814–3822, Sept 2013. https://doi.org/10.1109/TIE.2012.2208431.

[25] S. I. Han and J. M. Lee. Balancing and velocity control of a unicycle robot based on the dynamic model. *IEEE Transactions on Industrial Electronics*, 62(1):405–413, Jan 2015. https://doi.org/10.1109/TIE.2014.2327562.

[26] Ming-Tzu Ho, Y. Rizal, and Yi-Lung Chen. Balance control of a unicycle robot. In *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, pages 1186–1191, June 2014. https://doi.org/10.1109/ISIE.2014.6864782.

[27] M. A. Rosyidi, E. H. Binugroho, S. E. Radin Charel, R. S. Dewanto, and D. Pramadihanto. Speed and balancing control for unicycle robot. In *2016 International Electronics Symposium (IES)*, pages 19–24, Sept 2016. https://doi.org/10.1109/ELECSYM.2016.7860969.

[28] SparkFun 9DoF Razor IMU M0, SparkFun Electronics, https://www.sparkfun.com/products/14001, Accessed on 19-07-2018.

[29] EVAL-ADXRS623, ADXRS623 Evaluation Board, Analog Devices, Inc, http://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/EVAL-ADXRS623.html, Accessed on 18-07-2018.

[30] EVAL-ADXL327, ADXL327 Breakout Board, Analog Devices, Inc, http://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/EVAL-ADXL327.html, Accessed on 18-07-2018.

[31] J. Juan Rincón Pasaye, J. Alberto Bonales Valencia, and F. Jiménez Pérez. Tilt measurement based on an accelerometer, a gyro and a kalman filter to control a self-balancing vehicle. In *2013 IEEE International Autumn Meeting on Power Electronics and Computing (ROPEC)*, pages 1–5, Nov 2013. https://doi.org/10.1109/ROPEC.2013.6702711.

[32] IMU Data Fusing: Complementary, Kalman, and Mahony Filter, OlliW, 17-09-2013, http://www.olliw.eu/2013/imu-data-fusing/#chapter31, Accessed on 18-07-2018.

[33] Johannes van Dijk and Ronald Aarts. Analytical one parameter method for pid motion controller settings. *IFAC Proceedings Volumes*, 45(3):223 – 228, 2012. 2nd IFAC Conference on Advances in PID Control, https://doi.org/10.3182/20120328-3-IT-3014.00038.

[34] R.J. van Dasselaar, Longitudinal stabilization of a monocycle by a reactionwheel on top, 2016, University of Twente.

[35] EC 90 flat Ø90 mm, brushless, 90 Watt, with Hall sensors, Part number 323772, maxon motor ag, https://www.maxonmotor.com/maxon/view/product/323772, Accessed on 23-07-2018.

[36] Planetary Gearhead GP 52 C, maxon motor ag, https://www.maxonmotor.com/maxon/view/product/gear/planetary/gp52/223080, Accessed on 23-07-2018.

[37] EC 90 flat Ø90 mm, brushless, 90 Watt, with Hall sensors, Part number 244879, maxon motor ag, https://www.maxonmotor.com/maxon/view/product/244879, Accessed on 23-07-2018.

[38] ESCON 50/5, 4-Q Servocontroller for DC/EC motors, maxon motor ag, https://www.maxonmotor.com/maxon/view/product/control/4-Q-Servokontroller/409510, Accessed on 19-07-2018.

[39] Matlab & Simulink R2015B, The MathWorks, Inc.

[40] BNC-2110 Shielded Connector Block, National Instruments, http://sine.ni.com/nips/cds/view/p/lang/nl/nid/1865, Accessed on 11-09-2018.

[41] PCI-6221 Multifunction I/O Device, National Instruments, http://www.ni.com/nl-nl/support/model.pci-6221.html, Accessed on 11-09-2018.

[42] TopFuel LiPo 20C-ECO-X 3800mAh 6S, Hacker Motor GmbH, https://www.hacker-motor-shop.com/Akkus-und-Akkuzubehoer/TopFuel-EcoX/TopFuel-LiPo-20C-ECO-X-3800mAh-6S.htm?SessionId=&a=article&ProdNr=23800631&p=5649, Accessed on 19-07-2018.

[43] Feedback loop with descriptions, Orzetto, 15 October 2008, https://commons.wikimedia.org/wiki/File:Feedback_loop_with_descriptions.svg, Accessed on 12-09-2018.

[44] Understanding Poles and Zeros, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, DEPARTMENT OF MECHANICAL ENGINEERING, http://web.mit.edu/2.14/www/Handouts/PoleZero.pdfg, Accessed on 12-09-2018.

[45] Yuri Shtessel, Christopher Edwards, Leonid Fridman, and Arie Levant. *Sliding Mode Control and Observation*. Birkhäuser, 2014. https://doi.org/10.1007/978-0-8176-4893-0.

[46] Christopher Edwards and Sarah K. Spurgeon. *Sliding Mode Control Theory and Applications*. Taylor & Francis, 1998.

[47] Katsuhiko Ogata. *Discrete-Time Control Systems (2nd Edition)*. Pearson, 1995.

[48] Optimization-Based Control, Richard M. Murray, Control and Dynamical Systems, California Institute of Technology , 04-01-2010, http://www.cds.caltech.edu/~murray/books/AM08/pdf/obc-optimal_04Jan10.pdf, Accessed on 21-09-2018.

[49] D. McFarlane and K. Glover. A loop-shaping design procedure using h/sub infinity / synthesis. *IEEE Transactions on Automatic Control*, 37(6):759–769, June 1992. https://doi.org/10.1109/9.256330.

[50] Jorge A Romero-Bustamante, Jazael G Moguel-Castañeda, Hector Puebla, and Eliseo Hernandez-Martinez. Robust cascade control for chemical reactors: An approach based on modelling error compensation. *International Journal of Chemical Reactor Engineering*, 15(6), November 2017. https://doi.org/10.1515/ijcre-2017-0082.

A. Design Parameters

In the parameters below the back EMF max speed is the speed after which the back EMF will start to affect the current input into the motor, as calculated by Dannenberg [5]. Due to this the output torque of the motors will be lower than specified and at the maximum continuous speed it will actually completely counteract the input current and the motor can no longer deliver additional torque to the system (in that direction). The specifications of the driving of the reaction wheel, taking into account the limits of the motor [35] and the transmission [36] are:

Transmission ratio:	49/4
Torque constant motor:	70.5 mNm/A
Max continuous torque motor:	368 mNm
Stall torque motor:	1837 mNm
Max continuous current:	5.22 A
Stall current:	25.5 A
No load motor speed:	3190 rpm
Max cont. motor speed:	2590 rpm
Max cont. reaction wheel speed:	211 rpm
Max reaction wheel speed (Back EMF):	158 rpm

And the same specification for the motor [37] and transmission of the drive wheel are:

Transmission ratio:	60/25
Torque constant motor:	217 mNm/A
Max continuous torque motor:	533 mNm
Stall torque motor:	4460 mNm
Max continuous current:	2.27 A
Stall current:	21.1 A
No load motor speed:	2080 rpm
Max cont. motor speed:	1610 rpm
Max cont. drive wheel speed:	670 rpm
Max drive wheel speed (Back EMF):	122 rpm

The mechanical parameters of the unicycle are approximately:

Mass drive wheel (m_3):	0.115 kg
Mass reaction wheel (m_2):	1.5 kg
Mass frame (incl. electronics):	5.5 kg
Inertia drive wheel (J_3):	0.0001 kgm^2
Inertia reaction wheel (J_2):	0.017 kgm^2
Inertia frame (lateral, J_1):	0.086 kgm^2
Inertia frame (longitudinal, J_4):	0.122 kgm^2
Radius drive wheel (r):	0.05 m
Height reaction wheel (L_2):	0.316 m (+0.05 m (r))
Height CoM (without reaction wheel, L_1):	0.18 m (+0.05 m (r))
Height CoM (with reaction wheel, L_4):	0.21 m (+0.05 m (r))

B. Wiring Schematic

The unicycle is driven by two *Maxon EC 90 flat* motors [35, 37], which in turn are powered by two *ESCON 50/5* motor controllers [38]. The unicycle is controlled using *Simulink Real-Time R2015B* [39] using a sampling time of 0.001 seconds. To interface with the computer the *National Instruments BNC-2110* connector block [40] is used in combination with the *National Instruments PCI-6221* interface board [41]. To power the motors and motor controllers a 22.2V LiPo battery [42] is used. To measure the orientation of the robot, an analog triaxial accelerometer (EVAL-ADXL327Z [30]) and three analog gyroscopes (EVAL-ADXRS623 [29]) are used.

The wiring schematic of the moment exchange unicycle robot can be seen in figure B.1. The control signals are send from the analog output ports of the connector block to the analog input ports of the motor controllers. To enable the motors a digital signal is send from the connector block to the motor controllers. The motor controllers use the hall sensors and the encoders to control the motor. The encoder signals are connected to the connector block according to the specifications of the *National Instruments PCI-6221 Incremental Encoder* Simulink block in order to measure the wheel rotations. The digital grounds are also connected to the connector block to make sure that the connector block and the motor controllers have a common ground. The 5V output ports of the motor controllers are used to power the gyroscopes and the accelerometer. Since the accelerometer has a maximum input voltage of 3.6V, the 5V from the motor controller is led through three diodes to step the voltage down to about 3.2V. Finally, the measurement signals from the sensors are connected to the analog input ports of the connector block.

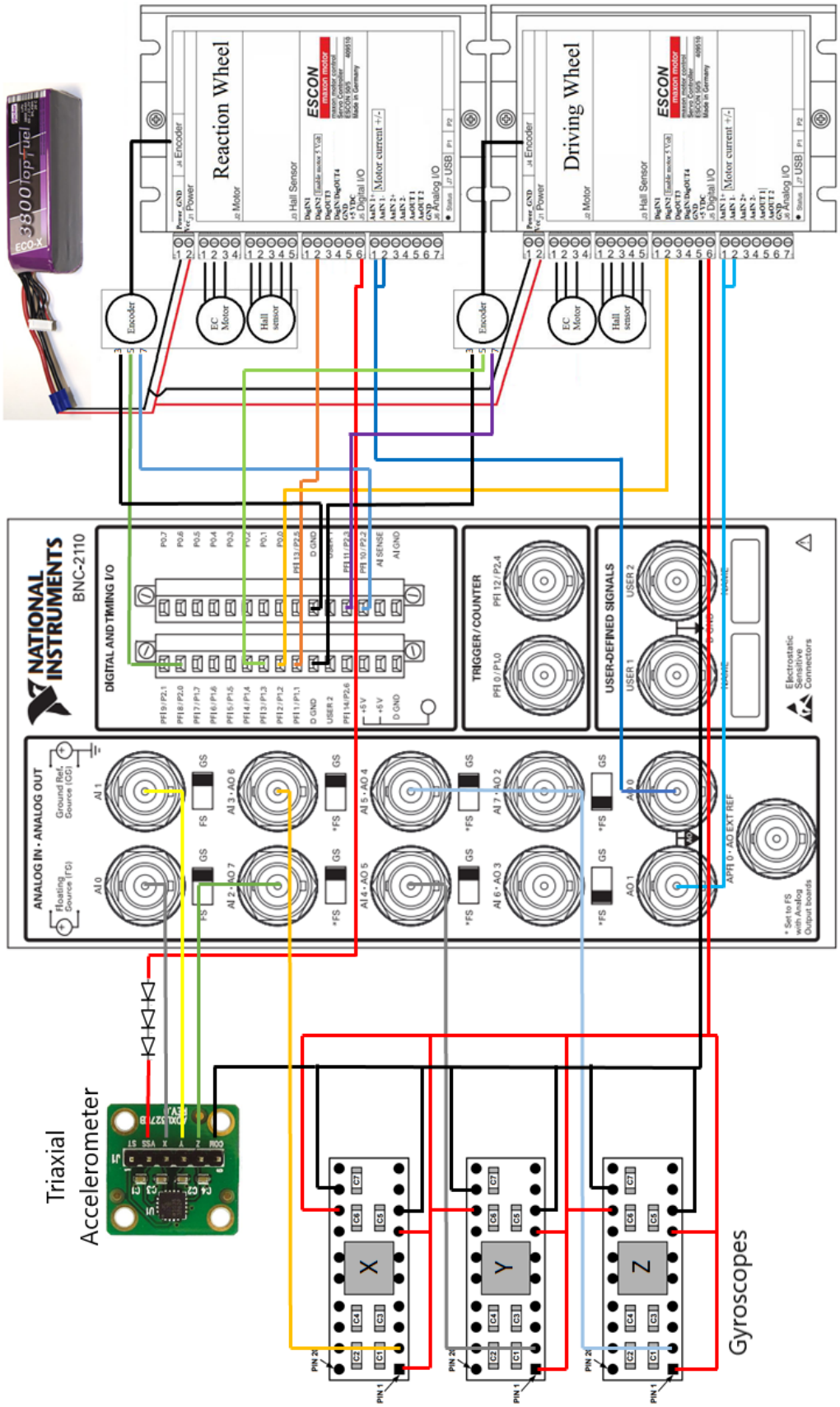


Figure B.1.: Wiring schematic.

C. Simulink Real-Time

The moment exchange unicycle robot is controlled using Simulink Real-Time from Matlab R2015B [39], with a sampling frequency of 1000 Hz. The main Simulink diagram can be seen in figure C.1, which consists of four subsystems. The first subsystem 'Encoders' is used to read out the encoders and determine the angle, angular velocity and angular acceleration of the drive and reaction wheel. The second subsystem 'Sensor block' is used to read out the signals from the gyroscopes and accelerometer and use these to determine the roll, pitch and yaw angles and their derivatives. The third subsystem 'CONTROL' contains the controllers which compute the required inputs into the system from the angles and angular velocities. Finally, the fourth subsystem 'SYSTEM' is used to convert the control inputs (torques) into the required motor currents and feed these into the actual robot.

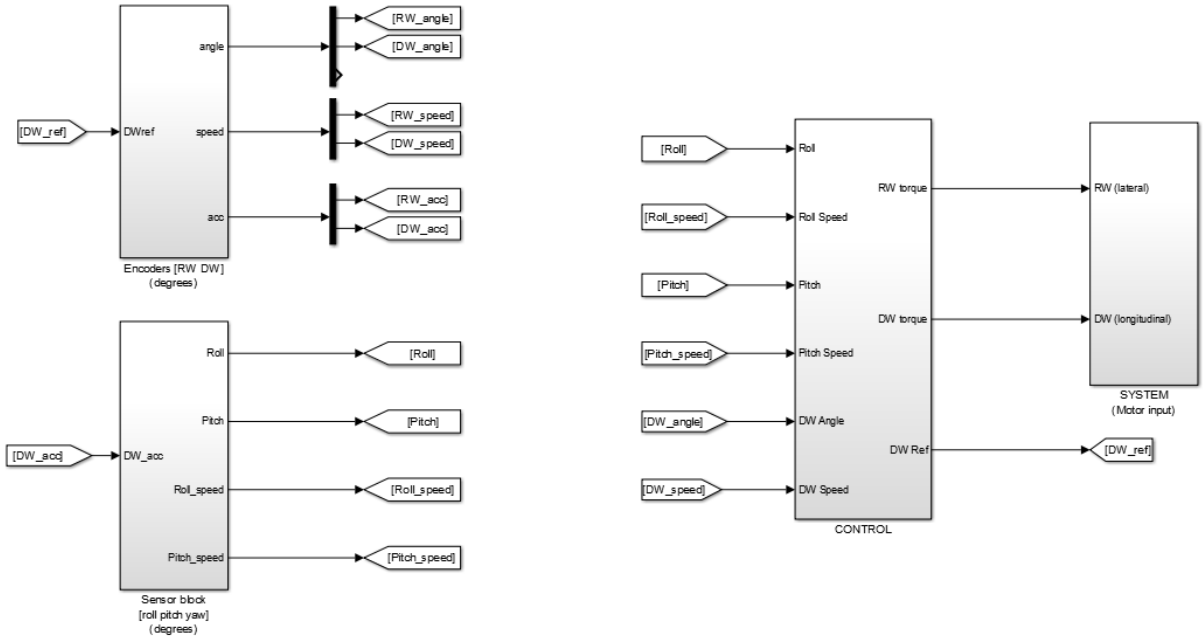


Figure C.1.: Main Simulink Real-Time diagram.

C.1. Encoders Subsystem

The Simulink diagram of the 'Encoders' subsystem can be seen in figure C.2. On the far left side of the diagram the encoders of the reaction and drive wheel motors are read out, after which their signals are converted into the angles of the reaction and drive wheel (in degrees). On the top these signals are named and then linked to the main diagram using an output port. The drive wheel angle and reference (which is obtained from the 'CONTROL' subsystem via the main diagram) are then converted to distances in meters and these signals are shown during operation using a target scope. On the bottom the first and second derivatives of the angles are taken using a third-order state variable filter to obtain the angular velocities and accelerations, after which these signals are also named and linked to the main diagram using output ports. The angular velocities of the wheels are also shown during operation using a target scope.

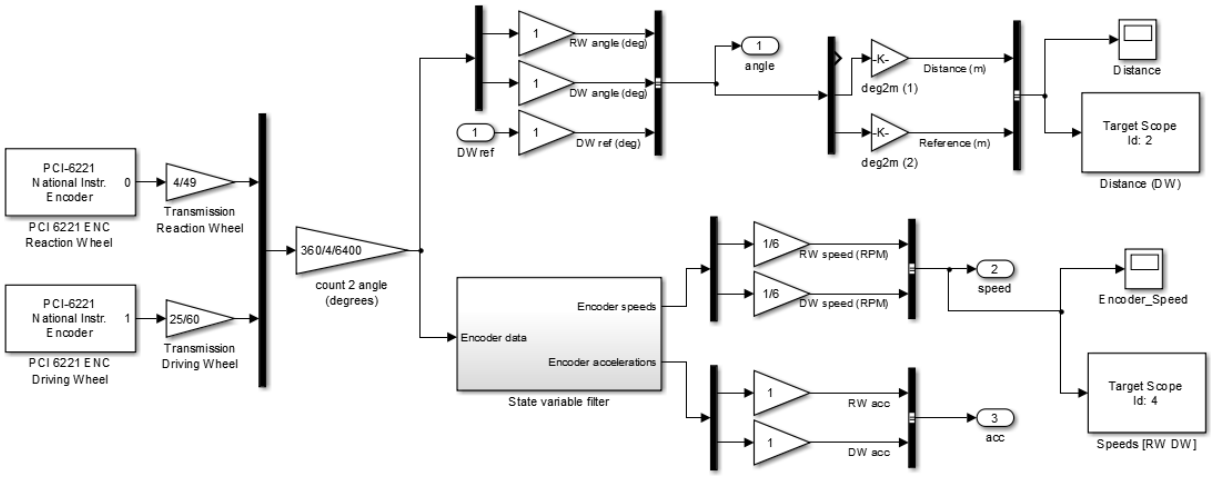


Figure C.2.: Simulink Real-Time diagram of the 'Encoders' subsystem.

C.2. Sensor Block Subsystem

The Simulink diagram of the 'Sensor block' subsystem can be seen in figure C.3. On the far left side of the diagram the output signals from the sensors are read using the analog input block. These signals are then converted into the accelerations and angular velocities measured by the sensors using their sensitivity and calibration values. To calculate the roll and pitch angles, the formulas derived in chapter 3 are used, for which also the acceleration of the drive wheel is used, obtained from the 'Encoders' subsystem via the main diagram. The yaw angle is simply obtained by integrating the gyroscope signal, keeping in mind that the bias changed with temperature and therefore can cause a certain drift when not calibrated properly. The three angles are shown during operation using a target scope. The derivative of the roll angle is obtained using a derivative block in combination with a low pass filter and the derivative of the pitch angle is obtained using the gyroscope signal, filtered by a high pass and low pass filter, since this gave better performance than using a derivative block in this case. Finally, the angles and angular velocities of the pitch and roll are linked to the main diagram using output ports.

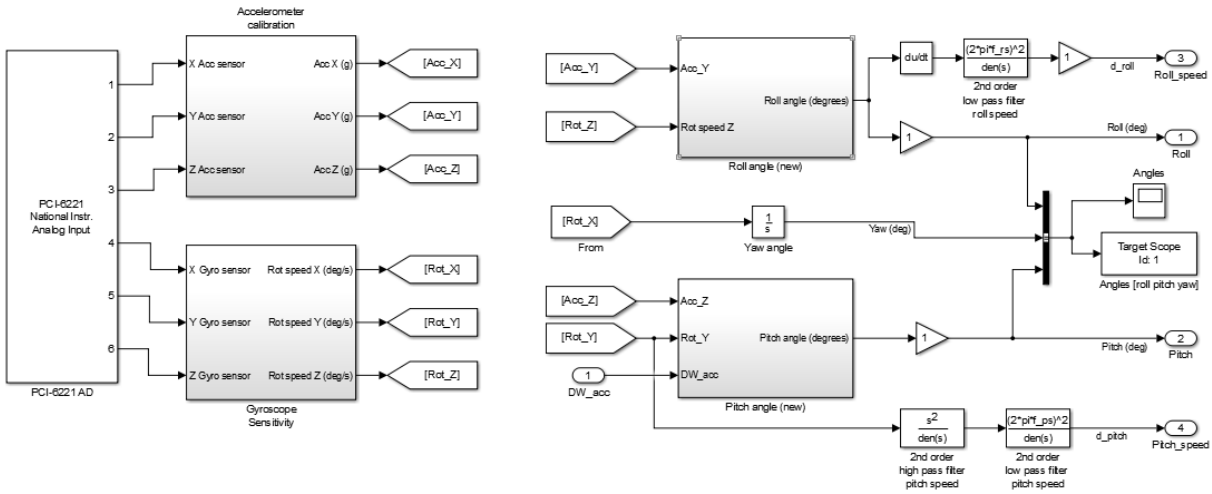


Figure C.3.: Simulink Real-Time diagram of the 'Sensor block' subsystem.

C.3. Control Subsystem

The Simulink diagram of the 'CONTROL' subsystem can be seen in figure C.4. On the top of the diagram the lateral control system can be seen. The angles calculated in the 'Encoders' and 'Sensor block' subsystems are obtained from the main diagram and used to compute the required torques to control the unicycle. As can be seen, the RW speed control uses the output torque of the SMC to adjust for the offset in roll angle. On the bottom of the diagram the longitudinal control can be seen. The sliding mode controller stabilizes the pitch angle, while the PID controller uses the angle and speed (for derivative action) of the drive wheel to compensate for any offset in the angle measurement. The reference signal is used to update the error in drive wheel angle, making the controllers generate the torques required to follow this reference. More details about the controllers can be found in chapter 4.

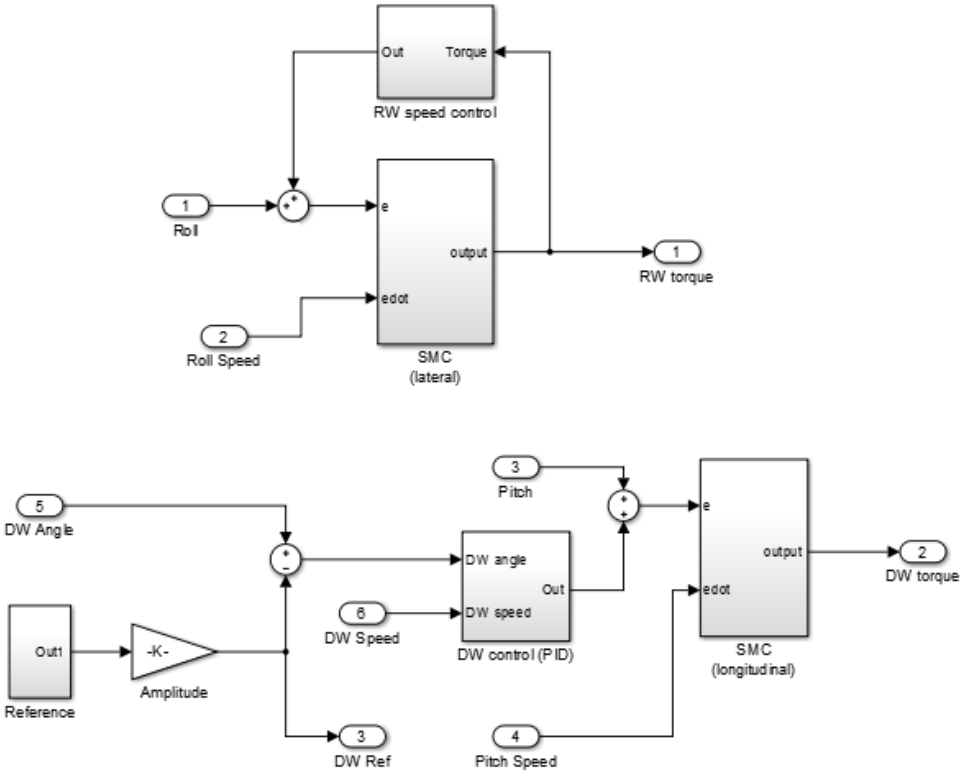


Figure C.4.: Simulink Real-Time diagram of the 'CONTROL' subsystem.

C.4. Motor Input Subsystem

The Simulink diagram of the 'SYSTEM' subsystem can be seen in figure C.5. The required torques to control the unicycle calculated in the 'CONTROL' subsystem are obtained from the main diagram and converted into the currents which should be fed into the motors to obtain these torques. A saturation block is used in both cases to prevent feeding in higher values than the maximum specifications of the motors. A step is used to enable the motors after five seconds of runtime, giving the operator the time to start the experiment and release the unicycle properly as well as giving the filters used for the angles time to settle. The input currents are shown during operation using a target scope and fed to the unicycle using the analog output

block. The gains just in front of the analog output block can be used to turn off the current signals going to the motors, which can be used to run the lateral and longitudinal controllers individually or test the sensors without the motors running.

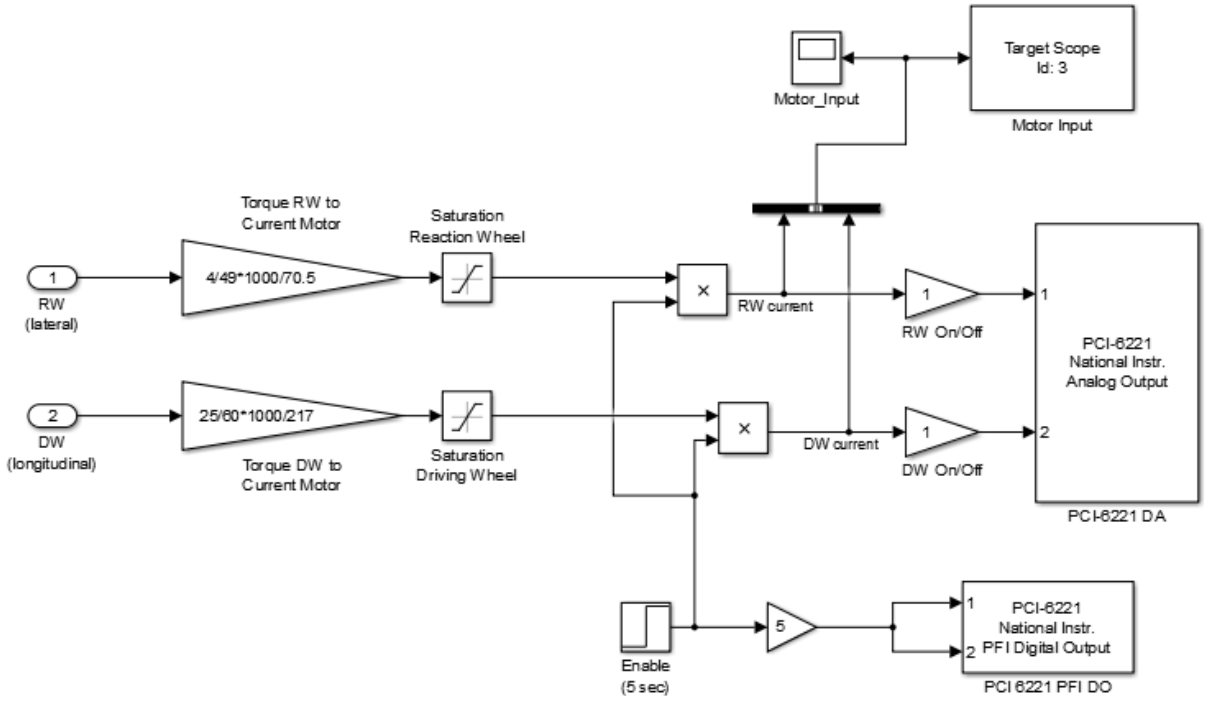


Figure C.5.: Simulink Real-Time diagram of the 'SYSTEM' subsystem.

D. Control Methods

Usually a feedback control loop is used to make sure a certain system variable will be equal to a desired setpoint or reference. For this the system variable will need to be measured, either directly or indirectly, and a control element, like an actuator, should be present. The control loop will then calculate the input for the control element required for the system to follow the reference. The general schematic of a feedback control loop can be seen in figure D.1.

Many different control methods exist, which each use a different approach to convert the measured system variable(s) into input signals for the actuators of the system. For the unicycle robot a couple of these methods will be investigated and compared in chapter 4 in order to find out which one will give the best performance. These methods will be introduced in this chapter.

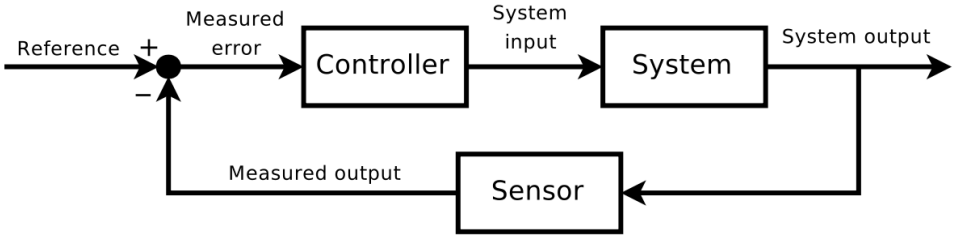


Figure D.1.: General feedback control loop. [43]

D.1. Stability

In general, an equilibrium point of a system is a stable point in case that when starting close enough to the point, the system will always remain arbitrarily close to it. For an asymptomatically stable equilibrium point the system will also always move towards the point.

The stability of a system can be determined by looking at the response of the system to its initial conditions without any additional inputs, its so called natural response. The natural response of a linear single-input and single-output (SISO) system to a set of initial conditions is described by:

$$y = \sum_{i=1}^{n} C_i e^{p_i t} \tag{D.1}$$

Where the constants C_i are determined from the given set of initial conditions and the exponents p_i are the poles of the system. For positive poles the response will grow exponentially, while for negative poles it will decay exponentially, as shown in figure D.2.

In general a controller is added to a system because it is not stable by itself. The stability of the controlled system should then be determined, which can be done by using the Nyquist stability criterion, which says: "When the Nyquist plot of $G(s)$ encircles the point -1 as many times in counter-clockwise direction as there are unstable poles in $G(s)$, then the closed loop transfer function $\frac{G(s)}{1+G(s)}$ will be stable."

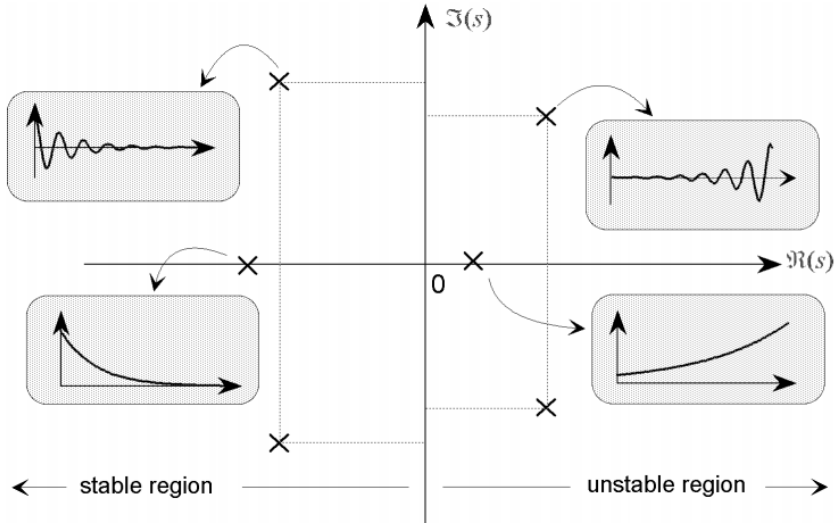


Figure D.2.: General shape of the components of the natural response of a system depending on the location of its poles. [44]

D.2. PD & PID

PD and PID controllers are widely used due to their simplicity and effectiveness. They consist of a Proportional action, a Derivative action and in case of the PID controller also of an Integral action. Each of these actions use the error value, which is the difference between the setpoint and the measured value. The proportional action reacts directly on the error signal and feeds an input into the system directly proportional to the error. Proportional action alone is however usually not enough to stabilize a system at the required setpoint within the required time, since it does not take into account the rate of change of the error. This will cause the system to overshoot once the setpoint is reached, making it oscillate around the desired value. The damping present in the system will eventually make it settle at the setpoint, but this could take quite some time. To make the controlled system settle much faster, artificial damping can be used in the controller, which is done by the derivative action. For this the derivative of the error is used to adapt the input signal and dampen the response of the controlled system. Due to certain system dynamics, there might still be a steady state error present in the system when only the proportional and derivative actions are used. In those cases integral action can be added to compensate for this, which uses the integrated error signal for this. Since taking the derivative of the error might cause amplification of noise, a low-pass filter can be added to the controller in order to prevent it from responding to this noise too much.

The proportional, derivative and integral action of the PD and PID controller simply apply a gain on the error signal, its derivative and its integral value. Since most of the noise will be present in the derivative action, the low-pass filter can be added to this part of the controller, giving the following equation:

$$C(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{s\tau + 1} \tag{D.2}$$

The response of the controlled system can be adjusted by tuning the controller constants K_p, K_i and K_d and the time constant τ of the low-pass filter. A simple way of tuning the controller is by using a heuristic approach, using a systematic trial and error method to first find a good value for the proportional gain, then find a derivative gain which gives good performance and

finally, if necessary, find an integral gain which gives good performance. What can be understood as good performance often depends on the type of application and usually a trade-off has to be made between fast tracking and the amount of overshoot. Under-damped systems will still have overshoot and might oscillate a few times before settling at the reference value but do respond relatively fast, while over-damped systems will take longer to reach the reference value but will not have any overshoot. A so-called critically damped system will have the fastest response for which there will be no overshoot. In case of the unicycle, a fast response is very important to prevent it from falling over, so an under-damped controller will most likely give the best performance.

Rather than using the heuristic method, the PD and PID controller can also be tuned by frequency domain loop-shaping using the method described in [33]. This approach uses the serial form of the PID controller, which has the following equation:

$$C(s) = k_p \frac{(s\tau_z + 1)(s\tau_i + 1)}{s\tau_i(s\tau_p + 1)} \tag{D.3}$$

In here the parameters k_p, τ_z, τ_i and τ_p are uniquely related to the controller constants K_p, K_i, K_d and τ of the parallel form shown in equation (D.2). Each of these parameters can be tuned using a desired cross-over frequency ω_c with the equations derived in [33]:

$$\tau_z = \frac{\sqrt{\frac{1}{\alpha}}}{\omega_c} \quad , \qquad \tau_i = \beta\tau_z \tag{D.4}$$

$$\tau_p = \frac{1}{\omega_c\sqrt{\frac{1}{\alpha}}} \quad , \qquad k_p = \frac{m_{eq}\omega_c^2}{\sqrt{\frac{1}{\alpha}}} \tag{D.5}$$

In these equations, α determines the amount of phase-lead and is usually set between 0.1 and 0.3. Since the start of the derivative action (determined by the transfer-zero in $\frac{1}{\tau_z}$) should be higher in frequency than the stop of the integral action (determined by the transfer-zero in $\frac{1}{\tau_i}$), it should be the case that $\beta \geq 1$. A value of $\alpha = 0.1$ has been chosen, which usually gives good performance, and to make sure the integral action does not deteriorate the phase-advance, it was chosen to use $\beta = 2$. The PID controller then only depends on the desired cross-over frequency ω_c. To tune a PD instead of a PID controller, the same formulas can be used to tune the parameters and the controller equation is simplified by leaving out the integral action: $C(s) = k_p\frac{s\tau_z+1}{s\tau_p+1}$.

D.3. Sliding Mode Control

When designing a controller for a system, there will always be a difference between the actual system and its mathematical model, since assumptions have to be made, parameters are estimated and disturbances cannot be taken into account. Designing a control law that provides the desired performance despite these uncertainties and disturbances is a very challenging task and has led to the interest in developing so-called robust control methods. One approach to such a robust controller is the so-called sliding mode control technique [45, 46].

Consider a system of which the state variables $x_1'(t)$ and $x_2'(t)$ need to be controlled to an equilibrium point located in the origin. First a new variable is introduced: $\sigma = x_2' + cx_1'$. The system dynamics will consist of a known part and an unknown part and in general will look something like:

$$\dot{\sigma} = c_1x_2' + c_2x_2' + f(x_1', x_2', t) + u(t), \qquad \sigma(0) = \sigma_0 \tag{D.6}$$

The unknown part of the system dynamics typically only drives the states to a bounded domain: $f(x_1', x_2', t) \leq D > 0$. To achieve asymptotic convergence of the state variables to the equilibrium point, the variable σ needs to be controlled to zero in finite time using the system input u. This can be achieved by using Lyapunov's stability criterion, which states: "In case there exists a differentiable positive definite function V(x) such that $\dot{V}(x) < 0$ for $x \neq 0$ then the origin is asymptotically stable." For a candidate Lyapunov function $V(\sigma) = \frac{1}{2}\sigma^2$, asymptotic stability will therefore be obtained if $\dot{V}(\sigma) < 0$ for $\sigma \neq 0$.

The derivative of the Lyapunov function is given by:

$$\dot{V}(\sigma) = \dot{\sigma}\sigma = \sigma(c_1 x_1' + c_2 x_2' + f(x_1', x_2', t) + u(t)) \tag{D.7}$$

Assuming $u(t) = -c_1 x_1' - c_2 x_2' + u'$ and taking into account the bounds of the unknown system dynamics, the following is obtained:

$$\dot{V}(\sigma) = \sigma(f(x_1', x_2', t) + u') = \sigma f(x_1', x_2', t) + \sigma u' \leq |\sigma| D + \sigma u' \tag{D.8}$$

Selecting $u' = -\rho\,\mathrm{sign}(\sigma)$, with $\rho > 0$, and substituting this into the equation, the following is obtained ($\sigma\,\mathrm{sign}(\sigma) = |\sigma|$):

$$\dot{V}(\sigma) \leq |\sigma| D + |\sigma|\rho = -|\sigma|(\rho - D) \tag{D.9}$$

When selecting a controller value $\rho > D$, the condition $\dot{V}(\sigma) < 0$ for $\sigma \neq 0$ is satisfied, so asymptotic stability will be obtained.

In general the control input of a sliding mode controller will therefore consist of two parts, a discontinuous part (u') and a so-called equivalent part: $u(t) = u' + u_{eq}$. The design of the controller therefore exists of three parts, the design of the sliding surface (σ) and the design of the discontinuous and equivalent control inputs.

D.3.1. Sliding Surface Design

The first step in designing a sliding mode controller is to design the sliding surface (σ). This is also known as the switching function, since it determines the direction of the discontinuous control and therefore switches the output of this part of the controller. Conventionally, the tracking error (e) is used and the sliding surface is defined as:

$$\sigma = \left(\frac{d}{dt} + \omega\right)^k e, \qquad \omega > 0 \tag{D.10}$$

In which ω can be an arbitrary positive number and k should be equal to $r_{sys} - 1$, where r_{sys} is the relative degree of the system. In case of the unicycle, $k = 1$ and the sliding surface can simply be defined as:

$$\sigma = \dot{e} + \omega e, \qquad \omega > 0 \tag{D.11}$$

As the equation shows, ω actually determines the slope of the sliding surface and therefore sets the bandwidth of the error dynamics. On the sliding surface, for $\sigma = 0$ and $e = y_r - y$, the dynamics show that the system (y) will follow a reference (y_r) as long as it stays on the sliding surface:

$$\begin{aligned} 0 &= \dot{e} + \omega e \\ \dot{y} + \omega y &= \dot{y}_r + \omega y_r \\ y(s + \omega) &= y_r(s + \omega) \\ y &= y_r \end{aligned} \tag{D.12}$$

D.3.2. Discontinuous control input design

The standard discontinuous part of the control input is equal to: $u' = \rho\,\text{sign}(\sigma)$. After reaching the sliding surface the system will however repeatedly cross the line $\sigma = 0$, so high frequency switching will occur. This switching of the input signal is known as chattering and is often undesirable. To prevent the chattering, a continuous estimation of the signum function can be used instead. A few different functions can be used for this and one of the most common is using the sigmoid function:

$$u' = \rho \frac{\sigma}{|\sigma| + \epsilon} \tag{D.13}$$

In which ϵ is a small positive number which should be tuned according to the application, since a higher value will give a smoother signal, but will also make the controller respond slower.

D.3.3. Equivalent control input design

As shown in equation (D.12), the system will follow a reference as long as it stays on the sliding surface. To keep the system on the surface and obtain ideal sliding, the equivalent control is used. Neglecting the disturbances and the discontinuous control input used to compensate for these, the system dynamics will be equal to:

$$\ddot{y} = \frac{1}{m_n} u_{eq} \tag{D.14}$$

In which m_n is the moving mass. Knowing that $\ddot{e} = \ddot{y}_r - \ddot{y}$, ideal sliding motion can be obtained by using:

$$\dot{\sigma} = 0 = \ddot{e} + \omega\dot{e} = -\frac{1}{m_n} u_{eq} + \ddot{y}_r + \omega\dot{e} \quad \rightarrow \quad \frac{1}{m_n} u_{eq} = \ddot{y}_r + \omega\dot{e} \tag{D.15}$$

As can be seen, the equivalent control will consist of a feed-forward part and a derivative control part. Since the unicycle will have to be stabilized in its equilibrium position rather than follow a reference, the feed-forward part can be left out. To improve the response of the controller a proportional part can be added, making the equivalent control equal to:

$$u_{eq} = k_p e + k_d \dot{e} \tag{D.16}$$

Since ω determines the bandwidth of the sliding dynamics, which can be expressed as: $\omega = \omega_c$, the proportional and derivative actions can be tuned using this value. When using $k_d = 2\omega$ and $k_p = \omega^2$, the sliding dynamics become:

$$\dot{\sigma} = \ddot{e} + \omega\dot{e} = -u_{eq} + \omega\dot{e} = -\omega^2 e - 2\omega\dot{e} + \omega\dot{e} \tag{D.17}$$

Knowing that $\omega\sigma = \omega\dot{e} + \omega^2 e$, the following differential equation is obtained and can be solved:

$$\frac{d}{dt}|\sigma| = -\omega|\sigma| \tag{D.18}$$

$$|\sigma| = |\sigma(0)|e^{-\omega t} \tag{D.19}$$

In here $|\sigma(0)|$ represents the initial distance away from the sliding surface. The equations show that using these parameters, the rate at which the sliding surface is reached is determined by the parameter ω. The parameter ρ can therefore be as small as possible, depending on the disturbances since it only has to compensate for these, while the equivalent control brings the system to the sliding surface and ensures that ideal sliding is obtained.

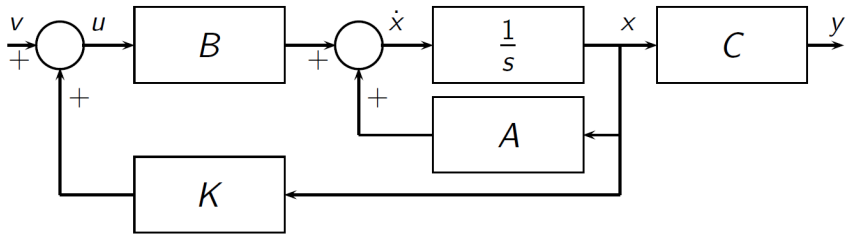


Figure D.3.: State feedback loop.

D.4. State-feedback Control - LQR

A state-feedback controller [47] is a special type of feedback controller which uses the states of a system to control its output. The general control loop with a state-feedback controller K can be seen in figure D.3. Using this control loop, the state space equation of the system will become:

$$\dot{x} = (A - BK)x + Bv \tag{D.20}$$

For stabilization, the setpoint $v = 0$ and the complete dynamics of the system are described by the eigenvalues of $(A + BK)$. To be able to apply state-feedback control, the system should either be completely reachable or stabilizable, meaning that the eigenvalues in the non-reachable sub-space should be stable since these cannot be changed. To find the control matrix K a few different methods can be used.

The linear-quadratic regulator (LQR) is a special algorithm which automatically finds the optimal state-feedback controller with respect to the criterion being applied. For a continuous system this criterion is defined as:

$$J = x^T(t_f)S_f x(t_f) + \int_{t_0}^{t_f} \left[x^T(t)Q(t)x(t) + u^T(t)R(t)u(t)\right] dt \tag{D.21}$$

In which $S_f = S_f^T \geq 0$, $Q(t) = Q^T(t) > 0$ and $R(t) = R^T(t) > 0$. Q and R represent the weighting functions on the states of the system and control inputs respectively. To find the optimal state-feedback control law, the solution for $u(t)$ has to be found such that J is minimized. Using Pontryagin's Principle, the solution can be found to be equal to [48]:

$$u(t) = -R^{-1}(t)B^T(t)S(t)x(t) = K(t)x(t), \qquad K(t) = -R^{-1}B^T(t)S(t) \tag{D.22}$$

To find S the differential algebraic Riccati equation can be used:

$$\dot{S} + SA + A^T S - SBR^{-1}B^T S + Q = 0, \qquad S_f = S(t_f) \tag{D.23}$$

In Matlab the function *care* can be used to automatically find S and the closed-loop eigenvalues of the system by inputting the state-space and weighting matrices: $[S, \mathrm{CL}_{eig}] = \mathrm{care}(A, B, Q, R)$. The stability of the controlled system can be checked by looking at these closed-loop eigenvalues, as long as these are all negative the system will be stable and will behave as shown in the left part of figure D.2.

For non-zero setpoints, the control loop needs to be adapted. If it is known what the state values should be in order to obtain the desired output of the system, like is the case for the unicycle, a setpoint (x_p) can be subtracted from the individual states as shown in figure D.4.

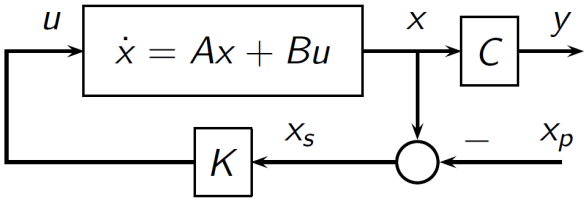
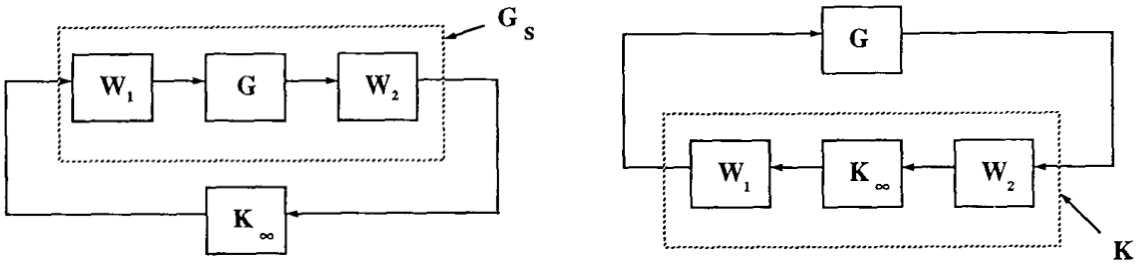


Figure D.4.: State feedback loop with a setpoint matrix x_p for the system states.

D.5. H_∞ Loop-shaping

For the H_∞ loop-shaping controller, the design procedure proposed by [49] was used, similarly as done by [15]. This method uses weighting functions W_1 and W_2, which add a weight on the control and error signals of the plant dynamics respectively. A controller K_∞ is designed and the weighting can be added to the controller instead of the plant without changing the closed-loop behaviour, as shown in figure D.5, such that the controller will become: $K = W_1 K_\infty W_2$.



(a) Weighted plant.

(b) Weighted controller.

Figure D.5.: Control loop including weighting functions W_1 and W_2. [49]

The Matlab function *augw* can be used to convert the system and its weight functions into an augmented plant: $G_s = augw(G, W_1, W_2, 1)$. To synthesize the H_∞ controller K_∞ for the weighted system G_s, the Matlab function *hinfsyn* can be used: $[K_\infty, CL] = hinfsyn(G_s)$, which also gives the closed-loop transfer function CL. This function computes the controller via the γ-iteration, computing the minimal cost function for which the closed-loop system minimizes the H_∞ norm. The weighting functions can then be used to obtain the controller: $K = W_1 K_\infty W_2$.

To find the best values for the weighting matrices, a heuristic method can be used, testing which values give the best results. To check the stability of the controller, the function *hinfsyn* also outputs the closed-loop system, which will be stable when all its poles are located in the negative plane as shown in figure D.2.

D.6. Cascaded Control

A system which has a single input, but has multiple outputs that should be controlled, is called an underactuated system. For such systems, a cascaded control structure can be used, which usually combines a fast inner loop with a slow outer loop. The inner loop is usually used for stability while the outer loop is used for tracking. The general schematic of the cascaded control structure can be seen in figure D.6. In here the slave controller is the inner loop, which in case of the unicycle will stabilize the pitch and roll angles, while the master controller is the outer loop, which updates the reference signal of the stability controller in order to control a second

state variable. The outer loop can thus be used to compensate for any offsets in the angle measurements by updating the reference angle at which the unicycle will settle and can also be used to make the unicycle follow a reference by tilting it forwards or backwards such that the drive wheel will start moving the unicycle accordingly.

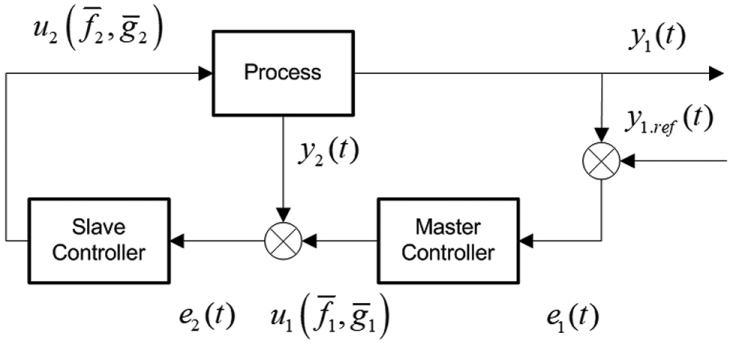


Figure D.6.: Cascaded control structure. [50]

E. Mathematical Model

Assuming that the unicycle will remain close to its upright position and only small angles are obtained, using the axis definitions as shown in figure 2.3 on page 3, the lateral and longitudinal direction can be considered as uncoupled systems, similarly as done in literature [4, 15, 24]. When also assuming rigid bodies, no slip occurring and no friction present in the system, the mathematical models become relatively simple. The neglected effects can be considered as disturbances to the system and be dealt with accordingly.

E.1. Lateral Model

For the lateral system, the driving wheel can be considered as part of the frame, such that the system is essentially an inverted pendulum with a reaction wheel on top, giving the schematics shown in figure E.1.

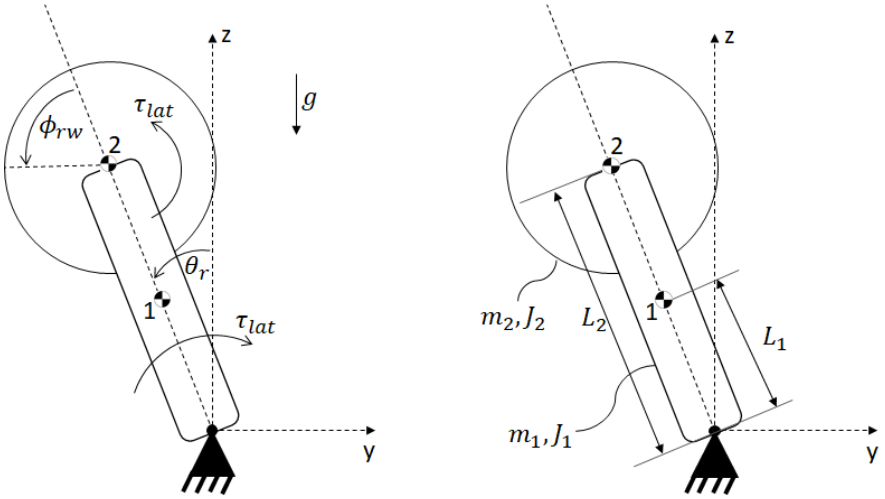


Figure E.1.: Schematics showing the variables and parameters of the lateral system.

E.1.1. Kinematics

The first step in finding the equations of motion is identifying the independent (q_i) and dependent coordinates (q_d) and expressing the dependent coordinates and their derivatives in terms of the independent coordinates (using the positions at the center of mass ◔ of the bodies):

$$q_i = \{\theta_r, \phi_{rw}\} \tag{E.1}$$
$$q_d = \{y_1, z_1, y_2, z_2\} \tag{E.2}$$

$$y_1 = -L_1 \sin(\theta_r), \qquad \dot{y_1} = -L_1 \dot{\theta_r} \cos(\theta_r) \tag{E.3}$$
$$z_1 = L_1 \cos(\theta_r), \qquad \dot{z_1} = -L_1 \dot{\theta_r} \sin(\theta_r) \tag{E.4}$$
$$y_2 = -L_2 \sin(\theta_r), \qquad \dot{y_2} = -L_2 \dot{\theta_r} \cos(\theta_r) \tag{E.5}$$
$$z_2 = L_2 \cos(\theta_r), \qquad \dot{z_2} = -L_2 \dot{\theta_r} \sin(\theta_r) \tag{E.6}$$

E.1.2. Kinetics

To derive the equations of motion, Lagrange's equation is used assuming rigid bodies and no damping and friction present:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \tag{E.7}$$

In here T_i is the kinetic energy ($T = \frac{1}{2}mv^2 + \frac{1}{2}J\omega^2$), V_i is the potential energy ($V = mgz$) and Q_i is the generalized force of the i^{th} degree of freedom. For the lateral direction of the unicycle these become:

$$T_1 = \frac{1}{2}m_1(\dot{y}_1^2 + \dot{z}_1^2) + \frac{1}{2}J_1\dot{\theta}_r^2 = \frac{1}{2}(m_1L_1^2 + J_1)\dot{\theta}_r^2 \tag{E.8}$$

$$T_2 = \frac{1}{2}m_2(\dot{y}_2^2 + \dot{z}_2^2) + \frac{1}{2}J_2(\dot{\theta}_r + \dot{\phi}_{rw})^2 = \frac{1}{2}m_2L_2^2\dot{\theta}_r^2 + \frac{1}{2}J_2(\dot{\theta}_r^2 + \dot{\phi}_{rw}^2 + 2\dot{\theta}_r\dot{\phi}_{rw}) \tag{E.9}$$

$$V_1 = m_1gz_1 = m_1gL_1\cos(\theta_r) \tag{E.10}$$

$$V_2 = m_2gz_2 = m_2gL_2\cos(\theta_r) \tag{E.11}$$

$$Q = \delta_1Q_1 + \delta_2Q_2 = -\tau_{lat}\delta\theta_r + \tau_{lat}\delta(\theta_r + \phi_{rw}) \rightarrow Q_1 = 0, \quad Q_2 = \tau_{lat} \tag{E.12}$$

Next the partial derivatives of Lagrange's equation can be calculated:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}_r}\right) = (m_1L_1^2 + J_1 + m_2L_2^2 + J_2)\ddot{\theta}_r + J_2\ddot{\phi}_{rw} \tag{E.13}$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\phi}_{rw}}\right) = J_2\ddot{\theta}_r + J_2\ddot{\phi}_{rw} \tag{E.14}$$

$$\frac{\partial V}{\partial \theta_r} = -(m_1L_1 + m_2L_2)g\sin(\theta_r) \tag{E.15}$$

$$\frac{\partial V}{\partial \phi_{rw}} = \frac{\partial T}{\partial \theta_r} = \frac{\partial T}{\partial \phi_{rw}} = 0 \tag{E.16}$$

With all terms of Lagrange's equation known, the equation can be rewritten in order to obtain the equations of motion:

$$\begin{bmatrix} m_1L_1^2 + J_1 + m_2L_2^2 + J_2 & J_2 \\ J_2 & J_2 \end{bmatrix}\begin{bmatrix} \ddot{\theta}_r \\ \ddot{\phi}_{rw} \end{bmatrix} + \begin{bmatrix} -(m_1L_1 + m_2L_2)g\sin(\theta_r) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \tau_{lat} \end{bmatrix} \tag{E.17}$$

The coupling in the equations of motion can be removed by rewriting the bottom line and substituting it into the equations of motion:

$$J_2\ddot{\theta}_r + J_2\ddot{\phi}_{rw} = \tau_{lat} \quad \rightarrow \quad \ddot{\phi}_{rw} = \frac{\tau_{lat} - J_2\ddot{\theta}_r}{J_2} = \frac{1}{J_2}\tau_{lat} - \ddot{\theta}_r \tag{E.18}$$

$$(m_1L_1^2 + J_1 + m_2L_2^2 + J_2)\ddot{\theta}_r + J_2(\frac{\tau_{lat}}{J_2} - \ddot{\theta}_r) - (m_1L_1 + m_2L_2)g\sin(\theta_r) = 0 \tag{E.19}$$

$$\ddot{\theta}_r = -\frac{1}{J_{eq,lat}}\tau_{lat} + \frac{(m_1L_1 + m_2L_2)g}{J_{eq,lat}}\sin(\theta_r) \tag{E.20}$$

$$\ddot{\phi}_{rw} = (\frac{1}{J_2} + \frac{1}{J_{eq,lat}})\tau_{lat} - \frac{(m_1L_1 + m_2L_2)g}{J_{eq,lat}}\sin(\theta_r) \tag{E.21}$$

$$J_{eq,lat} = m_1L_1^2 + m_2L_2^2 + J_1 \tag{E.22}$$

E.1.3. Linearisation

The equations of motion can be linearised around the upright position using the small angle approximation, due to which higher order terms can be neglected and the trigonometric functions can be simplified using $\cos(\theta) \approx 1$ and $\sin(\theta) \approx \theta$. Choosing both angles as the outputs of the system, the state space matrix then becomes:

$$\begin{bmatrix} \dot{\theta}_r \\ \ddot{\theta}_r \\ \dot{\phi}_{rw} \\ \ddot{\phi}_{rw} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(m_1 L_1 + m_2 L_2) g}{J_{eq,lat}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{(m_1 L_1 + m_2 L_2) g}{J_{eq,lat}} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_r \\ \dot{\theta}_r \\ \phi_{rw} \\ \dot{\phi}_{rw} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{J_{eq,lat}} \\ 0 \\ \frac{1}{J_2} + \frac{1}{J_{eq,lat}} \end{bmatrix} \tau_{lat} \tag{E.23}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \theta_r \\ \dot{\theta}_r \\ \phi_{rw} \\ \dot{\phi}_{rw} \end{bmatrix} \tag{E.24}$$

E.2. Longitudinal Model

For the longitudinal system, the reaction wheel can be considered as part of the frame, such that the system is essentially a wheeled inverted pendulum, giving the schematics shown in figure E.2.

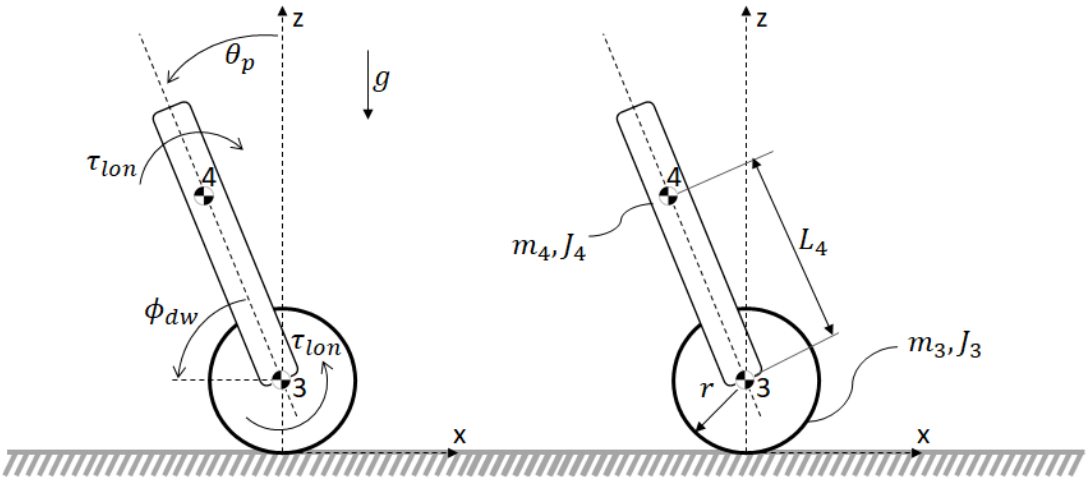


Figure E.2.: Schematics showing the variables and parameters of the longitudinal system.

E.2.1. Kinematics

The first step in finding the equations of motion is identifying the independent (q_i) and dependent coordinates (q_d) and expressing the dependent coordinates and their derivatives in terms of the independent coordinates (using the positions at the center of mass ◔ of the bodies):

$$q_i = \{\theta_p, \phi_{dw}\} \tag{E.25}$$

$$q_d = \{x_3, z_3, x_4, z_4\} \tag{E.26}$$

$$x_3 = -r(\phi_{dw} + \theta_p), \qquad \dot{x_3} = -r(\dot{\phi}_{dw} + \dot{\theta}_p) \tag{E.27}$$
$$z_3 = r, \qquad \dot{z_3} = 0 \tag{E.28}$$
$$x_4 = -r(\phi_{dw} + \theta_p) - L_4 \sin(\theta_p), \qquad \dot{x_4} = -r(\dot{\phi}_{dw} + \dot{\theta}_p) - L_4\dot{\theta}_p \cos(\theta_p) \tag{E.29}$$
$$z_4 = r + L_4 \cos(\theta_p), \qquad \dot{z_4} = -L_4\dot{\theta}_p \sin(\theta_p) \tag{E.30}$$

E.2.2. Kinetics

To derive the equations of motion, Lagrange's equation is used assuming rigid bodies and no damping and friction present:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \tag{E.31}$$

In here T_i is the kinetic energy ($T = \frac{1}{2}mv^2 + \frac{1}{2}J\omega^2$), V_i is the potential energy ($V = mgz$) and Q_i is the generalized force of the i^{th} degree of freedom. For the longitudinal direction of the unicycle these become:

$$T_3 = \frac{1}{2}m_3\dot{x}_3^2 + \frac{1}{2}J_3(\dot{\phi}_{dw} + \dot{\theta}_p)^2 = \frac{1}{2}(m_3r^2 + J_3)(\dot{\phi}_{dw}^2 + \dot{\theta}_p^2 + 2\dot{\phi}_{dw}\dot{\theta}_p) \tag{E.32}$$
$$T_4 = \frac{1}{2}m_4(\dot{x}_4^2 + \dot{z}_4^2) + \frac{1}{2}J_4\dot{\theta}_p^2 = \tag{E.33}$$
$$\frac{1}{2}m_4\left(r^2(\dot{\phi}_{dw} + \dot{\theta}_p)^2 + L_4^2\dot{\theta}_p^2 + 2rL_4(\dot{\phi}_{dw} + \dot{\theta}_p)\dot{\theta}_p\cos(\theta_p)\right) + \frac{1}{2}J_4\dot{\theta}_p^2$$
$$V_3 = m_3gr, \qquad V_4 = m_4gr + m_4gL_4\cos(\theta_p) \tag{E.34}$$
$$Q = \delta_1Q_1 + \delta_2Q_2 = \tau_{lon}\delta(\phi_{dw} + \theta_p) - \tau_{lon}\delta\theta_p \rightarrow Q_3 = \tau_{lon}, \quad Q_4 = 0 \tag{E.35}$$

Next the partial derivatives of Lagrange's equation can be calculated:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\phi}_{dw}}\right) = (m_3r^2 + J_3)\ddot{\phi}_{dw} + (m_3r^2 + J_3)\ddot{\theta}_p \tag{E.36}$$
$$+ m_4r^2\ddot{\phi}_{dw} + m_4r^2\ddot{\theta}_p + m_4rL_4\ddot{\theta}_p\cos(\theta_p) - m_4rL_4\dot{\theta}_p^2\sin(\theta_p)$$
$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}_p}\right) = (m_3r^2 + J_3)\ddot{\phi}_{dw} + (m_3r^2 + J_3)\ddot{\theta}_p + m_4r^2\ddot{\phi}_{dw} + (m_4r^2 + m_4L_4^2 + J_4)\ddot{\theta}_p \tag{E.37}$$
$$+ m_4rL_4\ddot{\phi}_{dw}\cos(\theta_p) - m_4rL_4\dot{\phi}_{dw}\dot{\theta}_p\sin(\theta_p) + 2m_4rL_4\ddot{\theta}_p\cos(\theta_p) - 2m_4rL_4\dot{\theta}_p^2\sin(\theta_p)$$
$$\frac{\partial V}{\partial \theta_p} = -m_4gL_4\sin(\theta_p) \tag{E.38}$$
$$\frac{\partial T}{\partial \theta_p} = -m_4rL_4(\dot{\phi}_{dw} + \dot{\theta}_p)\dot{\theta}_p\sin(\theta_p) \tag{E.39}$$
$$\frac{\partial V}{\partial \phi_{dw}} = \frac{\partial T}{\partial \phi_{dw}} = 0 \tag{E.40}$$

With all terms of Lagrange's equation known, the equation can be rewritten in order to obtain the equations of motion:

$$\begin{bmatrix} (m_3r^2 + m_4r^2 + J_3) & (m_3r^2 + m_4r^2 + J_3 + m_4rL_4\cos(\theta_p)) \\ (m_3r^2 + m_4r^2 + J_3 + m_4rL_4\cos(\theta_p)) & (m_3r^2 + m_4r^2 + m_4L_4^2 + J_3 + J_4 + 2m_4rL_4\cos(\theta_p)) \end{bmatrix} \begin{bmatrix} \ddot{\phi}_{dw} \\ \ddot{\theta}_p \end{bmatrix}$$
$$+ \begin{bmatrix} -m_4rL_4\dot{\theta}_p^2\sin(\theta_p) \\ -m_4rL_4\dot{\theta}_p^2\sin(\theta_p) - m_4gL_4\sin(\theta_p) \end{bmatrix} = \begin{bmatrix} \tau_{lon} \\ 0 \end{bmatrix} \tag{E.41}$$

The coupling in the equations of motion can be removed by rewriting the top line and substituting it into the equations of motion:

$$\left(m_3r^2 + m_4r^2 + J_3\right)\ddot{\phi}_{dw} + \left(m_3r^2 + m_4r^2 + J_3 + m_4rL_4\cos(\theta_p)\right)\ddot{\theta}_p - m_4rL_4\dot{\theta}_p^2\sin(\theta_p) = \tau_{lon} \tag{E.42}$$

$$\rightarrow \ddot{\phi}_{dw} = \frac{1}{m_3r^2 + m_4r^2 + J_3}\tau_{lon} - \frac{m_3r^2 + m_4r^2 + J_3 + m_4rL_4\cos(\theta_p)}{m_3r^2 + m_4r^2 + J_3}\ddot{\theta}_p + \frac{m_4rL_4\dot{\theta}_p^2\sin(\theta_p)}{m_3r^2 + m_4r^2 + J_3}$$

$$J_{eq,lon} = m_3r^2 + m_4r^2 + J_3 \tag{E.43}$$

$$\frac{J_{eq,lon} + m_4rL_4\cos(\theta_p)}{J_{eq,lon}}\tau_{lon} - \frac{\left(J_{eq,lon} + m_4rL_4\cos(\theta_p)\right)^2}{J_{eq,lon}}\ddot{\theta}_p + \frac{\left(J_{eq,lon} + m_4rL_4\cos(\theta_p)m_4rL_4\dot{\theta}_p^2\sin(\theta_p)\right.}{J_{eq,lon}}$$
$$+ \left(J_{eq,lon} + m_4L_4^2 + J_4 + 2 * m_4rL_4\cos(\theta_p)\right)\ddot{\theta}_p - m_4rL_4\dot{\theta}_p^2\sin(\theta_p) - m_4gL_4\sin(\theta_p) = 0 \tag{E.44}$$

This equation can be rewritten into the form $\ddot{\theta}_p = ...$ and can then be plugged into the other equation to obtain a similar equation for ϕ_{dw}. These equations do however contain a lot of different terms and therefore offer little overview, so only the simpler linearised versions will be shown in the next section.

E.2.3. Linearisation

The equations of motion can be linearised around the upright position using the small angle approximation, due to which higher order terms can be neglected and the trigonometric functions can be simplified using $\cos(\theta) \approx 1$ and $\sin(\theta) \approx \theta$. Choosing both angles as the outputs of the system, the state space matrix then becomes:

$$J_{eq,lon2} = J_{eq,lon} + m_4L_4^2 + J_4 + 2m_4rL_4 - \frac{(J_{eq,lon} + m_4rL_4)^2}{J_{eq,lon}} \tag{E.45}$$

$$\begin{bmatrix}\dot{\theta}_p \\ \ddot{\theta}_p \\ \dot{\phi}_{dw} \\ \ddot{\phi}_{dw}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0 \\ \frac{m_4gL_4}{J_{eq,lon2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{(J_{eq,lon}+m_4rL_4)m_4gL_4}{J_{eq,lon}J_{eq,lon2}} & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\theta_p \\ \dot{\theta}_p \\ \phi_{dw} \\ \dot{\phi}_{dw}\end{bmatrix} + \begin{bmatrix}0 \\ -\frac{J_{eq,lon}+m_4rL_4}{J_{eq,lon}J_{eq,lon2}} \\ 0 \\ \frac{\left(J_{eq,lon}+m_4rL_4\right)^2}{J_{eq,lon}^2J_{eq,lon2}} + \frac{1}{J_{eq,lon}}\end{bmatrix}\tau_{lon} \tag{E.46}$$

$$y = \begin{bmatrix}1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\theta_p \\ \dot{\theta}_p \\ \phi_{dw} \\ \dot{\phi}_{dw}\end{bmatrix} \tag{E.47}$$