CHALMERS
UNIVERSITY OF TECHNOLOGY

Modelling, Validation and Control of an Autonomous Bicycle

Master's Thesis in Systems, Control and Mechatronics

UMUR ERDİNÇ

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019

Modelling, validation and control of an autonomous bicycle

UMUR ERDİNÇ

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and Control
Mechatronics Research Group
Chalmers University of Technology
Gothenburg, Sweden 2019

Modelling, validation and control of an autonomous bicycle
UMUR ERDİNÇ

Supervisor: Jonas Sjöberg
Examiner: Jonas Sjöberg

Master's Thesis 2019
Department of Electrical Engineering
Division of Systems and Control
Mechatronics Research Group
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The autonomous bicycle that is designed and used throughout the project

Typeset in LaTeX
Gothenburg, Sweden 2019

Modelling, validation and control of an autonomous bicycle
UMUR ERDİNÇ
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Nowadays autonomous vehicles are becoming more and more popular day by day. The most important factor of design of autonomous vehicles is safety. To test bicycle accident scenarios of the autonomous vehicles, riderless bicycles are needed.
In this master's thesis project, modelling and validation of an autonomous bicycle are done. Simulation and real test results are presented and discussed.
First, an overview of the autonomous bicycle is given. Mechanics of the bicycle and derivation of the bicycle model are explained. Although there are several models available, the model used in this project is a second order non-linear model with steering angle input.
Second, a methodology for validating the parts of the bicycle is presented. The goal is to develop a methodology that can also be applied to any other autonomous bicycles as well. Forward speed motor and steering motor performances; and sensor reading accuracy of the bicycle are validated by following the presented methodology.
Then, control scheme of the bicycle is explained. For self-balancing, an LQR controller is used. For trajectory-tracking, on the other hand, a set of PID controllers are used.
Finally, simulation and real test results are presented and discussed. Sources of some problems are explained, and solutions are suggested. According to the simulation results, the bicycle can follow a straight line or a circle with minimum radius of 5 meters. In the real tests, only self-balancing performance is tested. It is shown that the bicycle can successfully balance itself only with steering input.

Keywords: autonomous, bicycle, self-balancing, trajectory-tracking, LQR, PID, control, validation.

Acknowledgements

Firstly, I would like to express my deepest gratitude to my supervisor, Prof. Jonas Sjöberg for his valuable advices and guidance. His supervision helped me to learn how to conduct a scientific research independently, and his high standards always motivated me to do my best.
I am particularly grateful for Boaz Ash's support. He was the one who constructed the autonomous bicycle used in this project. We worked together for several months. I could not have imagined having a better colleague for this project.
I would like to express my special thanks to Dr. Giuseppe Giordano for his valuable supports and contributions.
Last but not least, I would like to thank my wife Direniş Erdinç, my parents Serpil and Zeyneddin Erdinç, my sister Hande Erdinç and my young siblings Ayşe Idil and Can Yoldaş Çayli for their unconditional support.

Umur Erdinç, Gothenburg, March 2019

Contents

List of Figures

List of Symbols and Abbreviations

φ	roll angle
$\dot{\varphi}$	roll rate
$\ddot{\varphi}$	roll acceleration
δ	steering angle
$\dot{\delta}$	steering rate
a_x	acceleration in x direction
a_y	acceleration in y direction
a_z	acceleration in z direction
ω_x	angular speed in x direction
ω_y	angular speed in y direction
ω_z	angular speed in z direction
φ_{acc}	roll angle estimation due to accelerometer
δ_{eff}	effective steering angle
a	horizontal distance from the center of the rear wheel to the center of mass
b	distance between the rear and front wheel points of contact on ground
c	fork trail
d	horizontal distance from the center of the rear wheel to the inertial measurement unit
h	height of the center of mass
h_{IMU}	height of the inertial measurement unit
λ	fork angle
g	gravitational acceleration
L_x	angular momentum of the system around the x axis
D	inertia product with respect to xz axis
J	moment of inertia of the bicycle around x axis
ρ	radius of curvature
v	forward velocity
ψ	yaw angle
d^{error}	lateral position error
β	angle of velocity of the center of mass with respect to the longitudinal axis of the bicycle
$\dot{\psi}$	yaw rate
m	mass
LP	low-pass
HP	high-pass
dt	time interval between updates
a_{cp}	centripetal acceleration
C	low-pass filter coefficient of the complementary filter
C_{LP}	low-pass filter coefficient used to filter the roll rate

(x, y)	global coordinates of the bicycle
IMU	inertial measurement unit
CM	center of mass
LQR	linear quadratic regulator
RPM	revolution per minute

1
Introduction

1.1 Background

Autonomous vehicles are becoming more popular day by day. Design of autonomous vehicles is a challenging task for engineers. The most important factor of design is safety. Autonomous vehicles need to be tested considering every possible situation. Since vehicles usually share the roads with the bicycles, one of the crash scenarios is the accidents between vehicles and bicycles. Furthermore, it is harder to detect bicycles rather than detecting other vehicles and pedestrians. Hence, testing of bicycles is also an important design stage of autonomous vehicles. For testing purpose engineers need autonomous bicycles, since real cyclists may get hurt during the tests. Autonomous bicycles should have same rolling dynamics with the conventional bicycles for successful tests. For this reason, training wheels cannot be used and autonomous bicycle should also be self-balancing. By using such a realistic test bicycle, engineers can perform every kind of risky crash test throughout designing of autonomous vehicles.
Currently, there are two important studies about self-balancing bicycles where a controller was designed and implemented on a physical bike with success.

1. Tsinghua University designed a bicycle which can balance itself, travel at different velocities and perform turns [1].
2. Cornell University has also developed their own bicycle and has published a paper and several reports about achieving a self-balancing bike [2]. Their work is ongoing and they have published videos of their bicycle balancing itself and performing turns.

Research on bicycles started at the end of 19^{th} century. Whipple [3] and Carvallo [4], [5] derived rigid body dynamics equations and linearised around vertical equilibrium point. They showed that some bicycles can balance themselves, if they are moving at a certain speed. A comprehensive study that explains bicycle dynamics is done by Åström et al. [6].

1.2 Scope of Thesis

Scope of thesis and some factors that are not taken into consideration are listed below.

- In this study, modelling and simulation of an autonomous bicycle are presented. Validation of the bicycle components are done and the performance of the autonomous bicycle is tested.

- Bicycle should balance itself only by steering.
- For this project, initialization of the bicycle is not taken into account. In the real tests, bicycle is initialized by holding it by hand.
- It is assumed that the bicycle always moves on a flat surface with no inclination. Bicycle frame is rigid, only movable parts are wheels and steering bar. Pitch motion is not possible.

1.3 Outline

In the second chapter, an overview of the autonomous bicycle is given. Mechanics of the bicycle is discussed. Sign notation and parameters are shown. Sensors and actuators are explained. State measurements and estimations are presented.
In the third chapter, a methodology for validating performance of each component of the autonomous bicycle is given. Then sensors and actuators of the bicycle are validated and the results of validations are presented.
In the forth chapter, control scheme of the bicycle is given. Design of self-balancing controller and trajectory-tracking controller are discussed.
In the fifth chapter, firstly simulation results of self-balancing tests and trajectory-tracking tests are given and discussed. Then real self-balancing test results are presented and discussed. At the end of the chapter, concluding remarks are given.
In the conclusion of the thesis, what is done during the thesis project is briefly summarized, and possible future works are explained.

2

Overview of the Autonomous Bicycle

In this chapter, a general overview of the bicycle is given. First, model of the bicycle is explained. Governing equation of the bicycle is derived. State-space representation of this equation is presented. Then actuators of the bicycle are given. Sensors are presented and the output equations are derived. Lastly, state estimations are explained and the corresponding equations are given.

2.1 Model of the Bicycle

2.1.1 Mechanics

Using a front view of the bicycle, the roll angle φ is defined as in Figure 2.1. Steering angle δ is shown in Figure 2.3. The effective steering angle δ_{eff}, which is the angle between x axis and the front wheel axis in the top-view of the bicycle, is show in Figure 2.2. Relationship between δ_{eff} and δ is explained in detail in Appendix A.5.

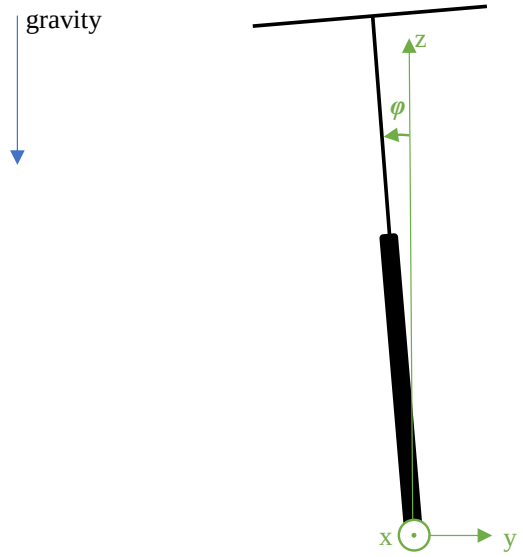


Figure 2.1: Front view of the bicycle and roll angle

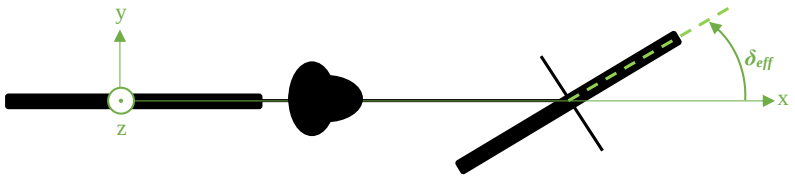


Figure 2.2: Top view of the bicycle and effective steering angle

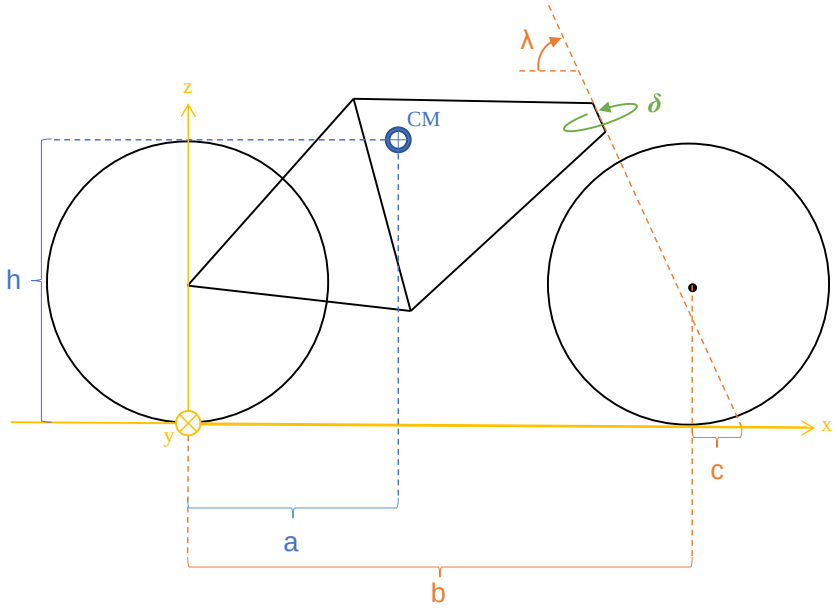


Figure 2.3: Side view and dimensions of the bicycle

Based on physical knowledge, a mathematical model of the bicycle can be derived. The geometrical dimensions of the bicycle are shown in Figure 2.3, where

h : height of the center of mass
b : distance between the rear and front wheel contact points on the ground
a : horizontal distance from the center of the rear wheel to the center of mass
λ : head angle of the fork
c : trail

Using the second Newton's law for rotations, the angular momentum balance of the system around the x-axis placed on the contact points of the wheels as shown in Figure 2.3 becomes (see also [6]):

$$\frac{d\ L_x(\dot{\varphi}, \delta, \lambda)}{dt} = T_G(g, h, \varphi) + T_C(b, h, \delta, \lambda, v^2) + T_F(a, b, c, \delta, \lambda), \tag{2.1}$$

where

g : gravitational acceleration
L_x : angular momentum of the system around the x axis
T_G : torque generated by the gravity
T_C : torque generated by the centrifugal force
T_F : torque due to the geometry of the front fork

A detailed description of these terms can be found in Appendix A.1. After few simplification steps, the final formulation of the mathematical model is

$$\ddot{\varphi} = \frac{g}{h}\varphi + \frac{a\ v\ \sin\lambda}{b\ h}\dot{\delta} + \frac{(h\ v^2 - g\ a\ c)\ \sin\lambda}{b\ h^2}\delta \tag{2.2}$$

2.1.2 State Space Model

A state-space (SS) form of the model (2.2) can be obtained by using 3 states:

- $x_1 = \varphi$
- $x_2 = \delta$
- $x_3 = \dot{\varphi}$

Control input is selected as $u = \dot{\delta}$ (steering rate).
In state-space (SS) form, the model (2.2) becomes

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ \frac{g}{h} & \frac{(h\ v^2 - g\ a\ c)\ \sin\lambda}{b\ h^2} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ \frac{a\ v\ \sin\lambda}{b\ h} \end{bmatrix} u \tag{2.3}$$

To linearise the system, forward velocity v is not selected as a state, since it is in multiplication with another state (x_2) and the control input. Hence, system matrix $\mathbf{A}$ and control matrix $\mathbf{B}$ are functions of v:

$$\dot{x} = \mathbf{A}(v)\ x + \mathbf{B}(v)\ u \tag{2.4}$$

Since varying v changes the plant dynamics, it is measured on-line and control input is calculated accordingly. Control law is explained in Chapter 4 and will not be discussed here.

2.2 Actuators

2.2.1 Steering Motor

Steering motor is selected as Maxon DCX32L Motor (24 V, 110 W) [14]. It is connected to a GPX 32 HP Planetary Gearhead (111:1 reduction). It is controlled via Maxon ESCON Module 50/5 [15]. Motor parameters are given in Appendix A.3.3.

2.2.2 Forward Velocity Motor

Forward velocity motor is Shimano STePS motor [16], which is the built-in motor of the electric bike. It is a brushless DC motor which is rated at 250W. It is controlled via Maxon ESCON Module 50/4 [17]. Motor parameters are given in Appendix A.3.4. Bicycle is always used at the 1^{st} gear to have high torque. This, however, decreases the maximum speed.

2.3 Sensors

In order to control the bicycle, measurements or estimates of the quantities describing the states of the bike are needed. In particular, four quantities need to be measured or estimated: roll angle, roll rate, steering angle and forward velocity. Based on these variables, a feedback control loop is implemented, which provides inputs to the bicycle system, in terms of steering speed ($\dot{\delta}^{ref}$) and desired longitudinal velocity (v^{ref}). Figure 2.4 summarizes the bicycle system and its quantities of interest.

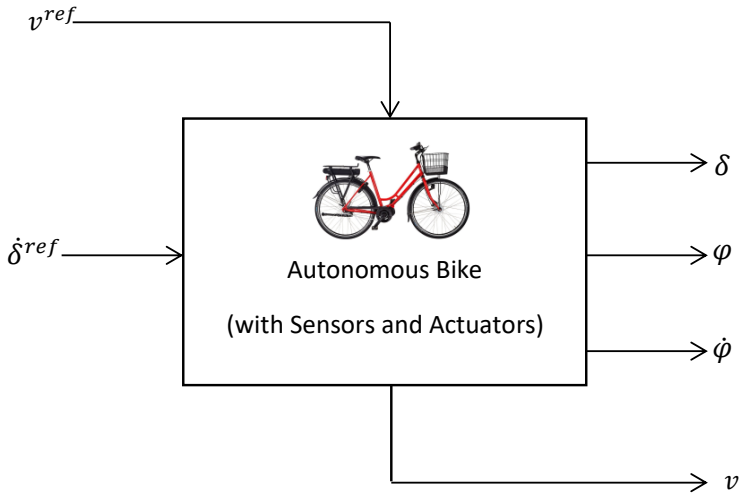


Figure 2.4: Inputs and outputs of the autonomous bicycle

The steering angle δ, can be directly measured via encoders. The forward velocity v is measured via a Hall Effect sensor. Roll rate can be directly measured via gyroscope. On the other hand, roll angle cannot be directly measured and an estimator should be used to find the roll angle. In the following, sensors are presented. Then state estimations and measurements are explained.

2.3.1 Hall Effect Sensor

To measure the forward velocity of the bicycle, a Hall Effect sensor (Honeywell 103SR13A-1) and five neodymium magnets (mounted on the rear wheel as shown in Figure (2.5)) are used [12]. The Hall Effect sensor generates a pulse when a magnet travels past it.

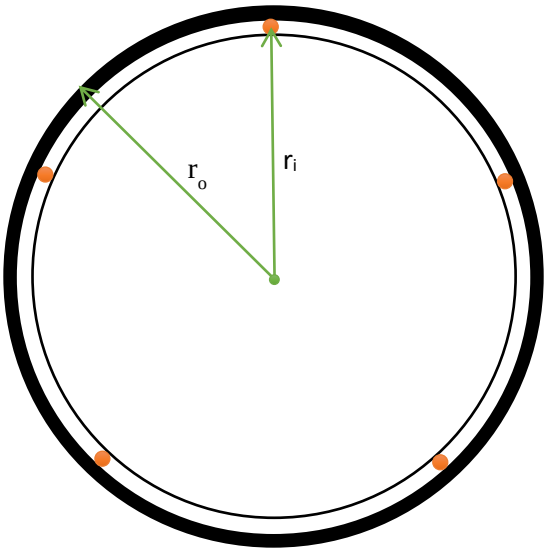


Figure 2.5: Placement of the neodymium magnets on the rear wheel

2.3.1.1 Measurement of the Velocity

Velocity of the bicycle is calculated according to (2.5) and (2.6).

$$v = \frac{r_o}{r_i} \frac{2\pi r_i}{5} \frac{1}{t_{pulse}} \tag{2.5}$$

$$if\ t_{pulse} < 0.65\ s \rightarrow set\ v = 0\ m/s \tag{2.6}$$

where
r_i : radius of the magnet placement circle (inner radius)
r_o : outer radius of the tyre
t_{pulse}: time difference in between two consecutive pulses
(2.6) indicates that, if time difference in between two consecutive pulses are more then 0.65 s, then velocity measurement becomes 0 m/s. For the dimensions of the bicycle ($r_i = 311$ mm and $r_o = 350$ mm), this corresponds to a minimum measurable speed of 0.68 m/s (2.44 km/h). If the speed is smaller than this value, measurement becomes 0 m/s.

2.3.2 Encoder

To measure the steering angle, an encoder (HEDS 5540) attached to the steering motor is used. This encoder has 500 counts per turn, and can operate up to 12000 RPM [13].

2.3.3 Inertial Measurement Unit

The inertial measurement unit (IMU) has a 3-axis gyroscope which measures the rotational speeds and 3-axis accelerometer which measures the force per unit mass (which is equivalent to acceleration). A sketch of IMU is shown in left hand side

of Figure 2.6 with its measurement axes. The IMU used in this project, MPU 6050 [11], is shown at the right hand side of Figure 2.6.

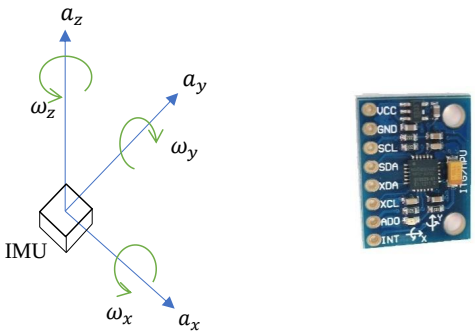


Figure 2.6: 3 measurement axes of IMU (left), MPU 6050 (right)

The notation used in Figure 2.6 is introduced below:
ω_i : gyroscope measurement around the i-th direction [rad/s], $i = x, y, z$
a_i : accelerometer measurement in the i-th direction [g], $i = x, y, z$

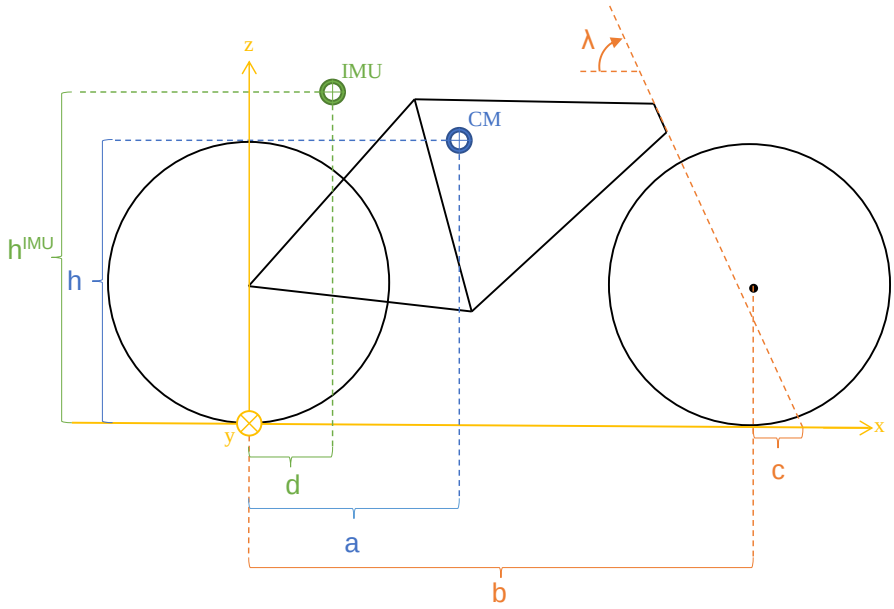


Figure 2.7: Dimensions of the bicycle with the IMU

The position of the IMU is defined by h_{IMU} (*height of the IMU from the ground*), and d (*horizontal distance from the center of the rear wheel to the IMU*), as shown in Figure 2.7.
For the formulations presented in this section, it is assumed that, IMU is mounted on the bicycle so that its local x, y, z axes are perfectly parallel to the x, y, z axes shown in Figure 2.7. Although it is not possible to align the axes perfectly, it is possible to align with a good precision, by checking accelerometer readings during the mounting process. This process is described in Appendix A.7.

Given the IMU position and assuming IMU alignment is done perfectly; the output equations of the IMU, relating the measured signals to the state variables becomes:

$$y_{IMU} = \begin{bmatrix} a_x \\ a_y \\ a_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{v} \\ -h_{IMU}\ G + B\ \cos x_1 + g\ \sin x_1 \\ -B\ \sin x_1 + g\ \cos x_1 \\ x_3 \\ \frac{v}{b}\ \tan(x_2 \sin\lambda) \sin x_1 \\ \frac{v}{b}\ \tan(x_2 \sin\lambda) \cos x_1 \end{bmatrix} \tag{2.7}$$

where G and B are given as:

$$G = \frac{g}{h} x_1 + \frac{(h\ v^2 - g\ a\ c)\ \sin\lambda}{b\ h^2} x_2 + \frac{a\ v\ \sin\lambda}{b\ h} u \tag{2.8}$$

$$B = \frac{v^2}{b^2}\ \tan^2(x_2 \sin\lambda)\ \sqrt{b^2\ \cot^2(x_2 \sin\lambda) + d^2} \tag{2.9}$$

In (2.7), blue terms are centripetal acceleration related terms and green term is related with the lateral acceleration caused by the roll motion. Derivation of (2.7) can be found in the Appendix A.4.

2.4 State Estimations and Measurements

According to state-space model (2.3), there are 3 states to be measured or estimated:

- $x_1 = \varphi$
- $x_2 = \delta$
- $x_3 = \dot{\varphi}$

Furthermore, although velocity is not considered as a state, it also needs to be measured or estimated.
In this section, explanations of the state measurements and estimations are given.

2.4.1 Directly Measured States (Roll Rate, Steering Angle, Velocity)

Directly measured states and sensors used to measure them are:

- $x_2 = \delta \rightarrow$ directly measured via steering encoder
- $x_3 = \dot{\varphi} \rightarrow$ directly measured via IMU (see fourth row of (2.7))
- $v \rightarrow$ directly measured via Hall Effect sensor located at the rear wheel

2.4.2 Estimated State (Roll Angle)

There is no sensor directly measures the roll angle. Hence, it can be indirectly measured by using the following approaches:

1. The roll rate, directly measured by the gyroscope, can be integrated to get the roll angle. However, gyroscope measurements show slow-varying bias. Because of the integration, estimated roll angle drifts by time. Hence, this method is trustable only in the short term since integration introduces low frequency noise to the roll angle estimation.
2. The second approach is to sense where the gravitational acceleration is directed to, by using the accelerometer.

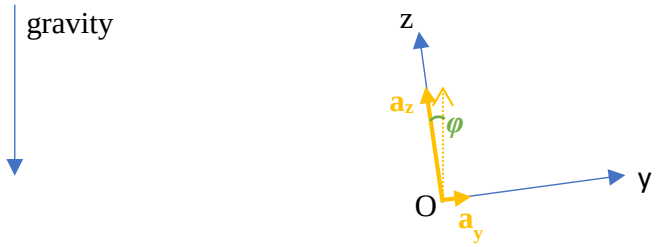


Figure 2.8: Derivation of roll angle by using the accelerometer

Considering Figure 2.8 and (2.7), the roll angle due to accelerometer $\hat{\varphi}_{acc}$ can be found by using the following relation

$$\hat{\varphi}_{acc} = \tan^{-1}\left(\frac{a_y + h_{IMU}\ G - B\ \cos x_1}{a_z - B\ \sin x_1}\right) \tag{2.10}$$

The problem of this approach is that the accelerometers are very sensitive to any acceleration. Hence, the measure of the gravity values can be disturbed a lot in non-stationary conditions with high frequency noise. Thus, the roll angle estimation of the accelerometer would be trustable only in the long term, because of the high frequency noise.

3. The third approach combines the two previous approaches. The idea is to fuse the two sources of information for the roll angle, by introducing a complementary filter that is designed in the following way: a low pass filter is used to remove the high frequency noise from the accelerometers; a high pass filter is used to remove the low frequency noise introduced by the gyroscope measurement integration.

In this study, a complementary filter is implemented and used for roll angle estimation, see Appendix A.6 for details. Roll angle is estimated by using the following formula:

$$\hat{x}_1 = \hat{\varphi} = \; C \; (\hat{\varphi}^{-} \; + \; \dot{\varphi} \;\; dt) \; + (1 - C) \; \hat{\varphi}_{acc} \tag{2.11}$$

where

C : filter coefficient ($0.5 < C < 1$)

$\hat{\varphi}$: roll angle estimation [rad]

$\hat{\varphi}^{-}$: previous roll angle estimation [rad]

$\dot{\varphi}$: gyroscope reading [rad/s]

dt : time interval between updates [s]

$\hat{\varphi}_{acc}$: roll angle estimation derived from accelerometer readings based on (2.10) in [rad]

For a detailed description about how to handle the green and blue terms of (2.10) in real applications, see Appendix A.8.

3

Validations of the State Estimations and Actuator Performances

Control performance of the autonomous bicycle directly depends on 2 factors:

- **State estimations:** Controller uses state estimations as feed-back signals. It is crucial to have accurate state estimations, hence those estimations should be validated. Furthermore, estimations are performed by using the sensor measurements. Thus, sensor performances should also be investigated.
- **Actuator performance:** Control inputs calculated by the controller are performed by the actuators. Therefore, actuators have an important impact on the control performance, and their performance should be validated.

In the first section of this chapter, validation methodology for the subsystems of the autonomous bicycle is presented. In the following sections, each of the subsystems is validated by using the presented methodology.

3.1 Validation Methodology for the Subsystems of the Bicycle

Namely, there are 3 state estimations/measurements and 2 actuator performances to validate:

1. Roll angle and roll rate estimations performed by using the IMU sensor measurements
2. Steering angle measurements obtained by using the encoder sensor
3. Forward velocity measurements obtained by using the Hall Effect sensor
4. Forward velocity motor performance
5. Steering motor performance

Validation tests for each part of the bicycle can be summarized as follows:

1. Roll angle estimations by using the IMU measurements
    - Testing the roll angle estimator for the previously known reference angles, by applying low and high frequency roll motion
2. Steering angle measurements of the encoder
    - Testing the encoder for different speeds to see whether it misses any pulse or not
3. Forward velocity measurements of the Hall sensor

- Comparing the results with a commercially available bicycle computer (speed sensor)
- Testing the hall sensor when the wheel is elevated to see whether it misses any pulse or not

4. Forward velocity motor performance
    - Testing the performance for a reference speed when the wheel is elevated by applying resistance by hand
    - Testing the performance for a reference speed on the ground with varying roll and steering angles
5. Steering motor performance
    - Testing each speed to identify the dead-band and hysteresis
    - Step response test

3.2 Validation of the Roll Angle and Roll Rate Estimations

In this section, validation of the roll angle and roll rate estimations are given. Particularly, results obtained with (2.11) are discussed. In simulations, roll angle estimations are compared with the true roll angles. In real tests, roll angle estimations are compared with the initially known static roll angle values.
The validations that will be done are listed below:
1. Simulation tests
1.1. Comparing the different versions of the estimation filters for a single motion (closing the feed-back loop only with the true roll angle) for self-balancing test (no trajectory-tracking)
1.1.1. with no accelerometer noise
1.1.2. with accelerometer noise
1.1.3. with accelerometer noise and additional constant disturbance on the accelerometer
1.2. Comparing the estimation errors of the different estimation filters for a circular trajectory by closing the feed-back loop with each kind of the filters
1.2.1. Examining the effect of constant accelerometer disturbance on the control performance for the circular trajectory-tracking
2. Real tests (with no forward velocity) for a set of known static roll angles

3.2.1 Simulation Results

In this section, four different versions of the complementary filter (see (2.11)) are simulated and the results are compared:

1. **Complementary filter with no neglected term:** To calculate the roll angle estimation due to accelerometer ($\hat{\varphi}_{acc}$), (2.10) is used.
2. **Complementary filter with neglected centripetal acceleration:** To calculate the roll angle estimation due to accelerometer ($\hat{\varphi}_{acc}$), (2.10) is used by neglecting the blue terms which are related with centripetal acceleration.
3. **Complementary filter with neglected roll acceleration:** To calculate the roll angle estimation due to accelerometer ($\hat{\varphi}_{acc}$), (2.10) is used by neglecting the green term which is related with roll acceleration.
4. **Complementary filter with neglected centripetal acceleration and roll acceleration:** To calculate the roll angle estimation due to accelerometer ($\hat{\varphi}_{acc}$), (2.10) is used by neglecting both the green and blue terms which are related with roll acceleration and centripetal acceleration respectively.

For the sake of clarity, each version will be referred together with their numbers (1, 2, 3, 4) in the following sections.

3.2.1.1 Self-Balancing Test by Closing the Feed-Back Loop Only with the True Roll Angle

In Figures 3.2, 3.3, and 3.4 the roll angle estimation of four versions of the complementary filters are presented and compared with the true (simulated) roll angle.

All simulations are done by closing the loop from the true roll angle, as shown in Figure 3.1. In other words, roll angle feedback is done with the true roll angle for all simulations (*It is not possible in reality, but can be done in simulation.*). Thus, we have only one motion and 4 different complementary filter estimations. Since we have same motion for all simulations, we can plot the results on the same plot. Hence, we can clearly compare each version of the filter.

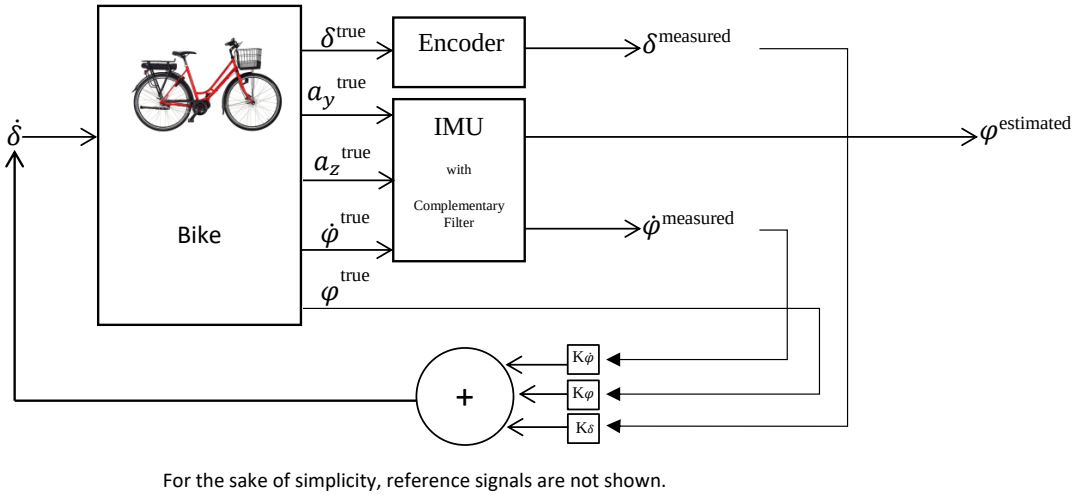


Figure 3.1: Closed loop system with the true roll angle feedback

For all the tests in this section, controller tries to balance the bicycle (no trajectory-tracking, only self-balancing). And in between 1^{st} second and 2^{nd} second bicycle is disturbed with a 0.576 rad/s^2 (33°/s^2) roll angle acceleration.

In the first result presented in Figure 3.2, only gyroscope noise which is recorded from the real sensor is included to the simulation and there is no noise on the accelerometers (For details of sensor noises, see Appendix A.10).

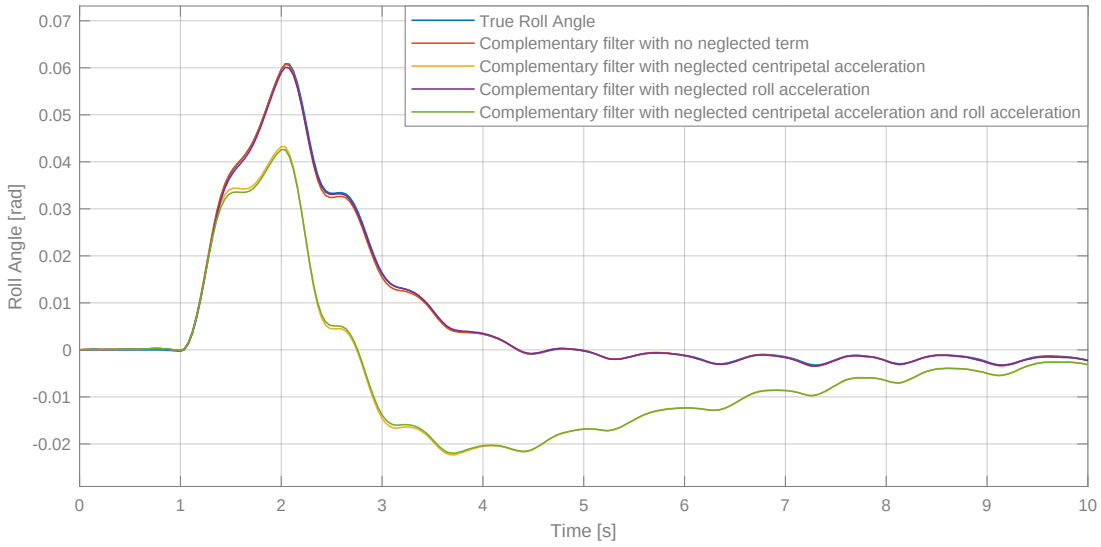


Figure 3.2: Roll angle estimations for different versions of the complementary filter with only gyroscope noise

The smallest roll angle estimation error is achieved with the complementary filter with no neglected term (version 1). In other words, by considering the effects of both centripetal acceleration and roll acceleration on the accelerometer readings. The roll angle estimation obtained by neglecting the effect of the roll acceleration (version 3) is also very close to the true value, but it is still less accurate than the complementary filter with no neglected terms. The roll angle estimations obtained by neglecting the centripetal acceleration (versions 2 and 4), on the other hand, are inaccurate and should not be trusted in the real implementation.

In the second result presented in the Figure 3.3, both gyroscope and accelerometer noises which are recorded from the real sensors are included to the simulation.

In this case, true roll angle and complementary filter (with no neglected term) estimation differ. This is because of non-zero averaged accelerometer noise which causes an offset (To see the noise profile of accelerometer, see Appendix A.10). This off-set, however, is very small (0.005 rad=0.29°). Effect of this off-set on the control performance will be examined in Section 3.2.1.2.

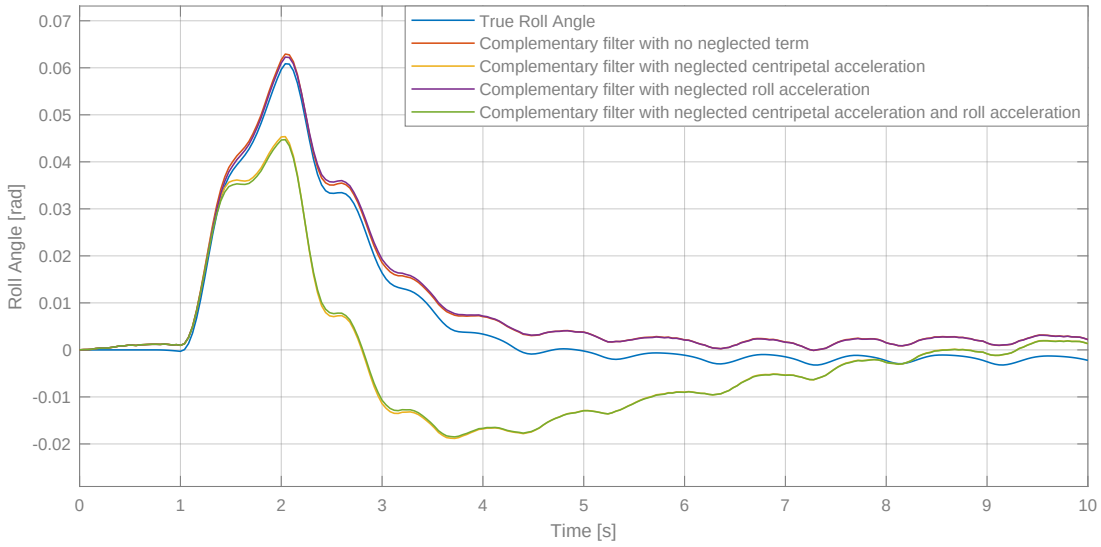


Figure 3.3: Roll angle estimations for different versions of the complementary filter with gyroscope and accelerometer noises

In the third result presented in Figure 3.4, both gyroscope and accelerometer noises which are recorded from the real sensors are included to the simulation. Furthermore, there is 0.1 g constant disturbance in the lateral accelerometer measurement.

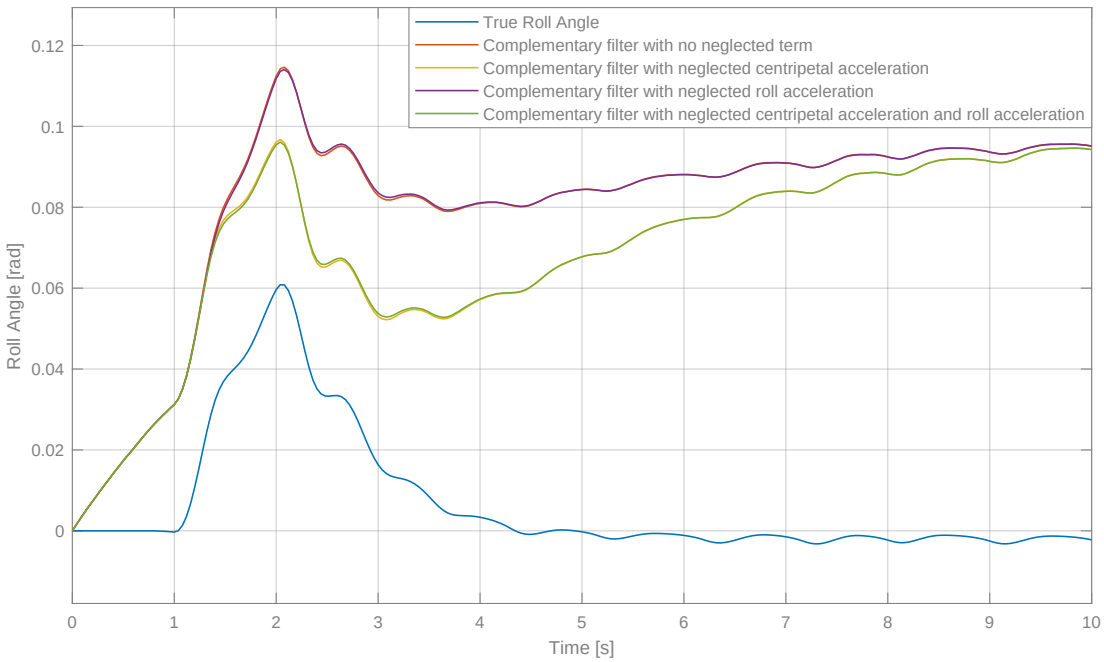


Figure 3.4: Roll angle estimations for different versions of the complementary filter with gyroscope and accelerometer noises and 0.1 g constant disturbance in lateral accelerometer measurement

Similar to the Figure 3.3, constant disturbance in accelerometer caused an off-set

at the roll angle estimations. This time, off-set is larger, since disturbance is larger than the average noise of the accelerometer. And there is gradual increase in the roll angle estimations at the initial second. This is due to the nature of the low-pass filter (used to filter the roll angle estimation of the accelerometer). Effect of the off-set on the control performance will be examined in Section 3.2.1.2.

To sum up the results presented in this section, the lateral acceleration caused by roll motion (green term in (2.7)) is not big enough to affect the roll estimation significantly. On the other hand, the centripetal acceleration terms (blue terms in (2.7)) have higher impact. Non-zero noise or disturbance on the accelerometer readings causes an off-set at the roll angle estimations.

3.2.1.2 Circular Trajectory-Tracking Tests by Closing the Feed-Back Loop with Different Complementary Filter Versions

In the simulation results presented in this section, control loop is closed by using different kinds of complementary filters, as shown in Figure 3.5. So, a unique motion is obtained for each simulation, since the feed-back signal is different for each case. Hence, effect of different complementary filter versions on the control performance can be checked and compared. Note that, control algorithm is not presented yet, and it will be discussed in Chapter 4. For the results presented in this section, there is no need for a deep knowledge on control algorithm.

For each simulation in this section, bicycle follows a circular path with 10 m radius. Both real (measured and recorded) accelerometer and gyroscope noises are included to all of the simulations.

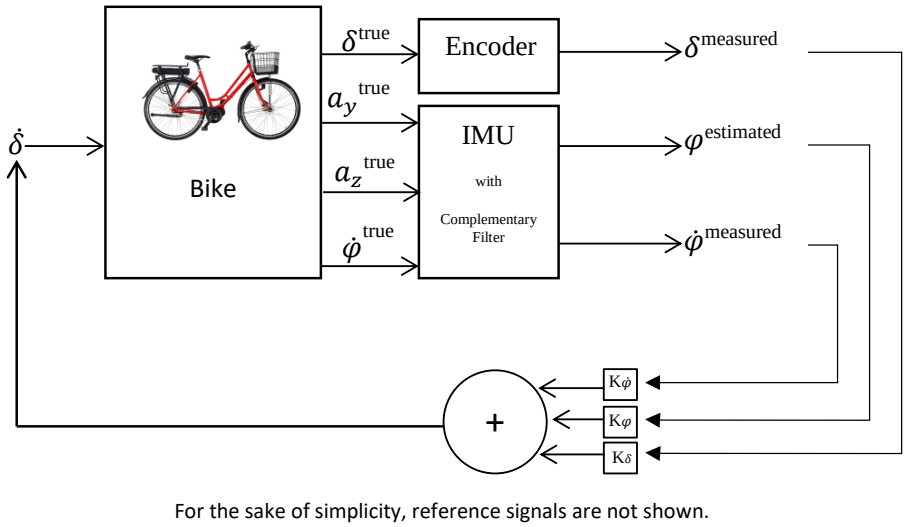


Figure 3.5: Closed loop system with estimated roll angle feed-back

In Figure 3.6, result for the complementary filter with no neglected terms (version 1) is presented. Roll angle error of the filter is always smaller than 0.007 rad, and bicycle can follow the circular path successfully with this version of the filter.

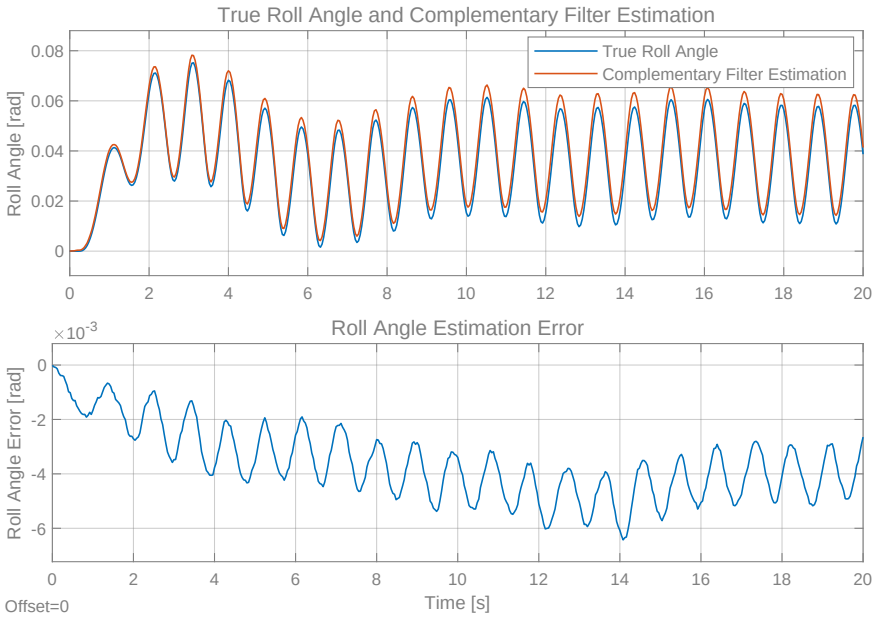


Figure 3.6: Performance of the **complementary filter with no neglected terms** (version 1) for a circular trajectory-tracking simulation

In Figure 3.7, result for the complementary filter with neglected roll acceleration terms (version 3) is presented. Roll angle error of the filter is always smaller than 0.006 rad, and bicycle can follow the circular path successfully with this version of the filter. In fact, results are very similar to the previous results with the complementary filter with no neglected terms (version 1).

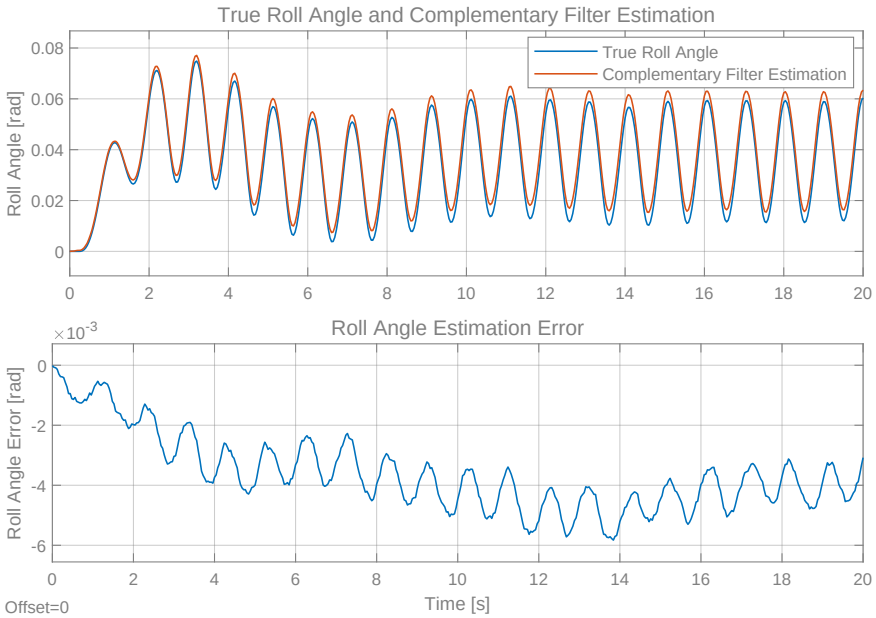


Figure 3.7: Performance of the **complementary filter with neglected roll acceleration terms** (version 3) for a circular trajectory-tracking simulation

In Figure 3.7, result for the complementary filter with neglected centripetal acceleration terms (version 2) is presented. Roll angle error of the filter reaches 0.1 rad; bicycle cannot follow the circular path successfully and it rolls over (roll angle reaches $\pi/2$ rad). In fact, although its result is not presented here, this particular version of the filter cannot stabilize the bicycle even for a self-balancing test without trajectory-tracking. Hence, this version is not good enough to be used in the real bicycle.

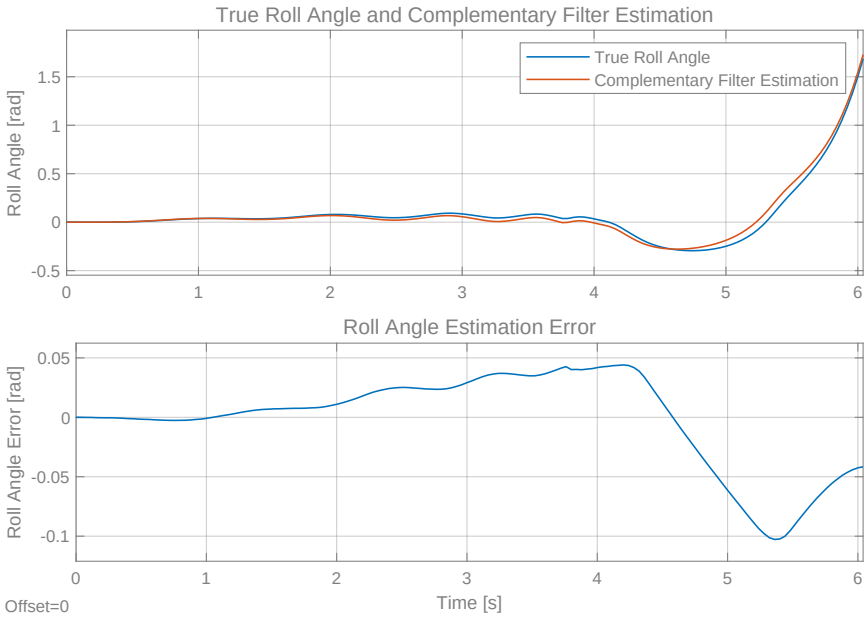


Figure 3.8: Performance of the **complementary filter with neglected centripetal acceleration terms** (version 2) for a circular trajectory-tracking simulation

Lastly, in Figure 3.9, result for the complementary filter with no neglected terms is presented. But this time, a constant disturbance of 0.1 g on the lateral accelerometer reading is included to the simulation. Although the estimation error reaches 0.1 rad, bicycle can follow the path. Hence complementary filter with no neglected terms is strong enough to tolerate the constant disturbance on the accelerometer readings. This kind of disturbance is particularly important in the real life, since an incorrect placement of the IMU causes a constant disturbance on the accelerometer readings.

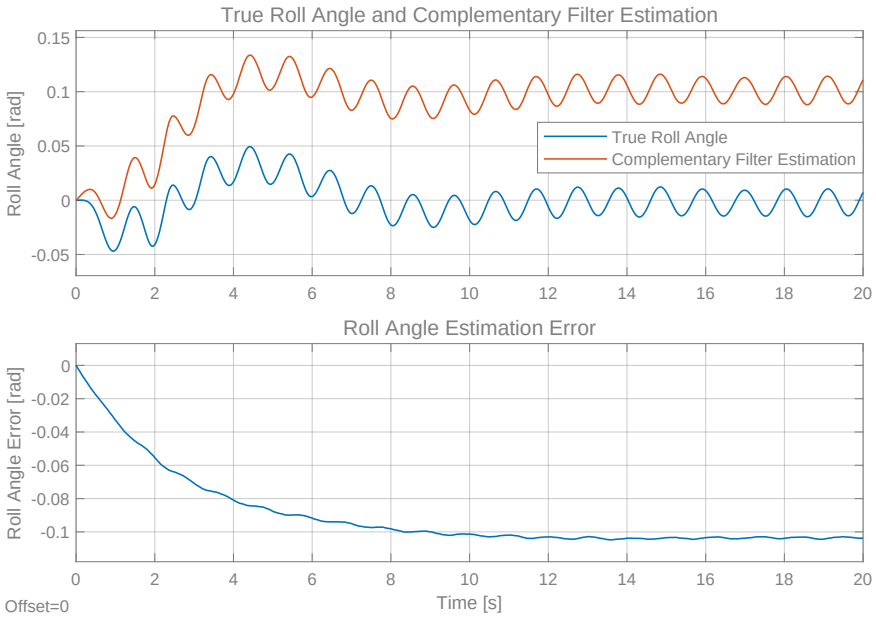


Figure 3.9: Performance of the **complementary filter with no neglected terms** for a circular trajectory-tracking simulation, **0.1g constant disturbance** on the lateral accelerometer measurement

To sum up the results presented in this section, complementary filter with no neglected terms (version 1) gives successful control performance on the simulations, and it can even tolerate constant disturbance on the accelerometers. Roll acceleration term (green term in (2.7)) can be neglected to decrease the computational cost, but centripetal acceleration terms (blue terms in (2.7)) are important and cannot be neglected.

3.2.2 Experimental Results

For an experiment, it is not possible to know the true roll angle, unlike simulation tests presented in the Section 3.2.1. However, we can still compare the roll angle estimation for some predefined and directly measured values of the roll angle. These values are the defined in the following way:

- Non-tilted bicycle at the up-standing still position, i.e. roll angle equal to 0 rad.
- (Lower bound) Tilted bicycle at a manually-measured roll angle value of -0.147 rad ($-8.4°$).
- (Upper bound) Tilted bicycle at a manually-measured roll angle value of $+0.147$ rad ($+8.4°$).

These last two values correspond to the reasonable bounds for the roll angle of the bicycle in a standard biking scenario. Furthermore, these values are small enough for the small angle approximation used in (2.2).

In the first test, which we will call "static test", the bicycle is manually tilted from the zero position to the lower and upper bounds, with zero forward velocity. The estimated roll angle will be then compared with the predefined values. In a second test, which we will call "dynamic test", the bicycle will be tilted with high frequency sinusoidal oscillations, in order to check the dynamic response of the estimation filter.

In Figures 3.10 and 3.11, the experimental results of the two tests are shown. Since the forward velocity of the bicycle is 0, there is no centripetal acceleration, in other words, blue term of (2.10) is 0. The upper and lower limits are shown within green and yellow areas in the figures. These areas represent the confidence region of the estimated roll angle, obtained via propagation of the accuracy of the accelerometers to the roll angle estimation. Since the accuracy of the accelerometer readings is 0.01 g, the accuracy of the angle estimation due to accelerometer can be propagated by using (2.10) as follows:

$$\varphi^{accuracy}_{accelerometer} = \tan^{-1}(0.01g/1g) = 0.01rad \tag{3.1}$$

Hence, any roll angle estimation due to accelerometer is accurate up to ± 0.005 rad. Because of this accuracy margin, upper and lower limits are shown with an interval with ± 0.005 rad in the Figures 3.10 and 3.11, rather than a single line. Similarly, 0 rad is also shown as an (purple) interval with ± 0.005 rad.

In the static test (Figure 3.10), the initial roll angle of the bicycle is set as 0 rad, then it is tilted between the upper and lower limit several times. Each time the roll angle is at its bound, the bicycle is kept still for few seconds. Then, the bicycle is tilted back to the zero roll angle position and then it is left at standstill. The procedure is repeated one more time. The time instances where the bike is left at standstill at the limiting angles are shown with blue arrows at Figure 3.10. For these standstill times, roll angle estimation lies in the marked intervals in Figure 3.10. This validates the correctness of the measurements for these values.

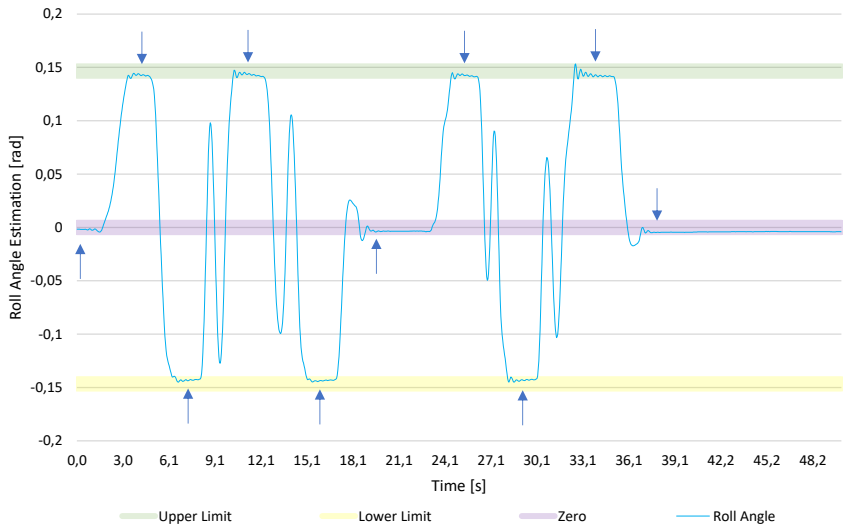


Figure 3.10: Test 1 (static test): Roll angle estimation

In the second test (Figure 3.11), the bicycle is manually tilted mimicking a sinusoidal wave characteristic. Until the 25^{th} second, it is tilted at a low frequency by reaching the limits each time, and then the tilting frequency is increased and tilting amplitude is decreased. At the end, it is left at standstill at the lower limit. According to Figure 3.11, roll angle estimation reaches the limits each time until 25^{th} second which is the expected result. And high frequency oscillations didn't cause any problem during the test, and they didn't cause an inaccurate result at the end of the test. Roll angle estimation is within the lower limit interval at the end, which is the expected value.

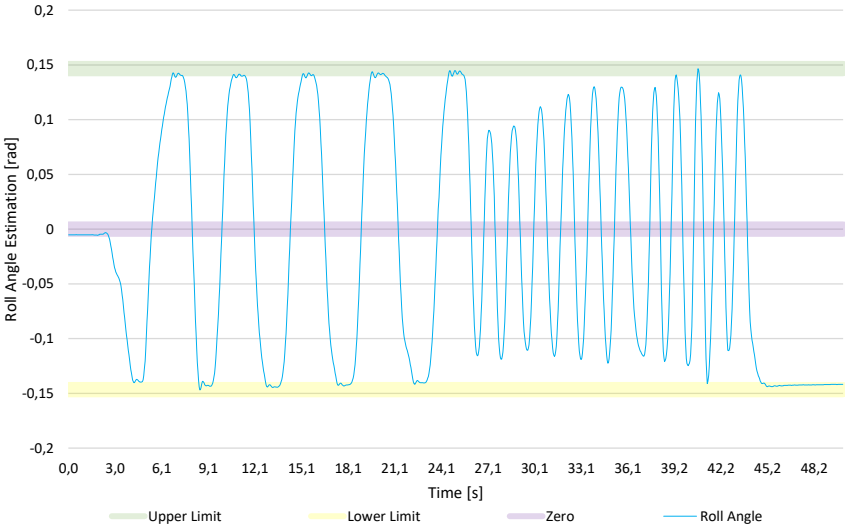


Figure 3.11: Test 2 (dynamic test): Roll angle estimation

To sum up the results obtained in this section, the correctness of the roll angle estimations are validated for the known static roll angles.

3.3 Validation of the Steering Angle Measurements

In this section, steering angle measurements obtained by encoder will be validated. Encoder has sensitivity of 500 counts per turn, and can operate up to 12000 RPM according to data-sheet provided by the manufacturer [13]. Hence, there will be no validation for the correctness of the measurements. The focus of this section is to validate that the encoder never misses any pulse.

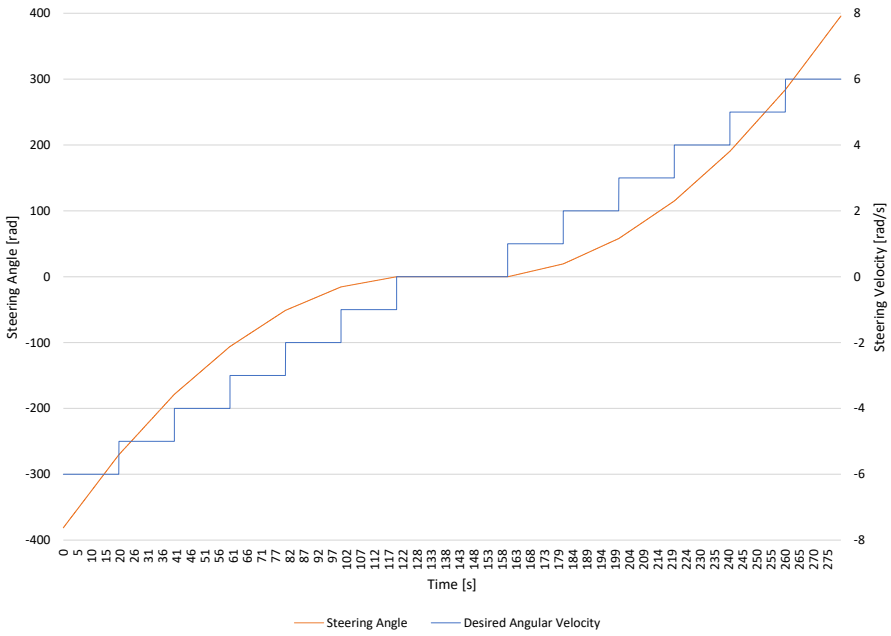


Figure 3.12: Measured steering angles for different steering velocities

In Figure (3.12), measured steering angles for different steering velocities are shown. For each steering velocity from -6 rad/s to +6 rad/s (with 1 rad/s increments), steering angle measurements are measured for 20 seconds. The results indicate that the encoder never misses pulses (steering angle plot is a smooth line) and that it is capable of providing its output for the entire range of motor speeds.

3.4 Validation of the Velocity Measurements

In this section, forward velocity measurements will be validated.

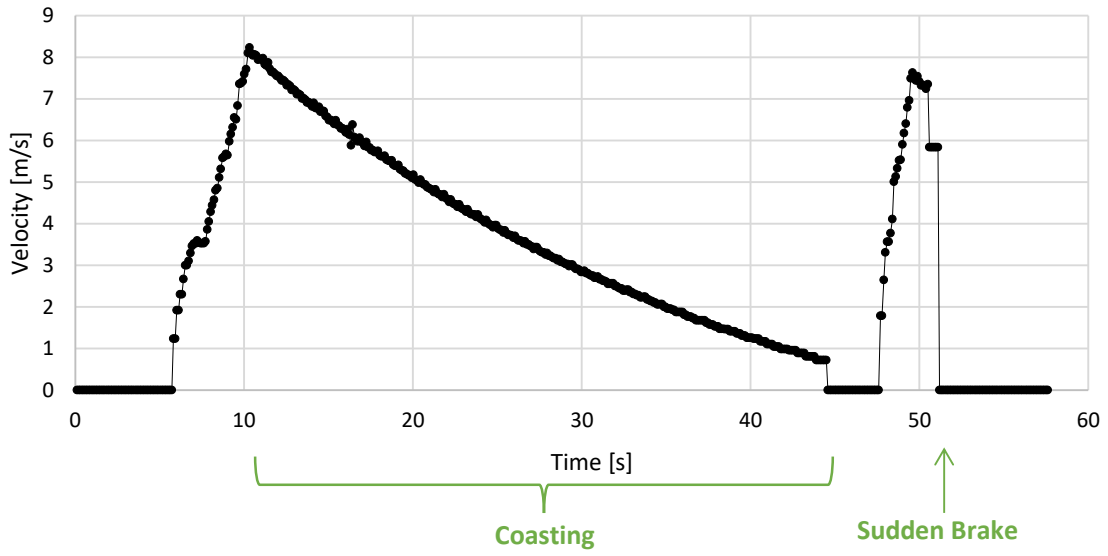


Figure 3.13: Hall Effect sensor velocity measurement graph

In Figure (3.13), a test result which includes coasting and sudden braking (when the rear wheel is elevated) is shown. As it can be seen from coasting part, reading becomes 0 m/s when it reaches to values around 0.68 m/s. It is an expected fact according to Section 2.3.1.1, where it is explained that the minimum measurable speed is 0.65 m/s. Before that point, measurements are smooth.
Results are compared with the commercially available Shimano speed sensor of the bicycle and it is seen that results are in good agreement within ±0.2 km/h (±0.056 m/s).

3.5 Validation of the Forward Velocity Motor Performance

In this section, performance of the forward velocity motor is validated. Several tests are done when the rear wheel is elevated, and also when the bicycle is on the ground. In the first test (see Figure (3.14)), the rear wheel is freely rotating without touching the ground, but external resistance applied to the wheel at the points marked with the yellow dots, resistance is removed at the points marked with the green dots. The bicycle can achieve the desired velocity (2 m/s for this test) even when a resistance is applied.

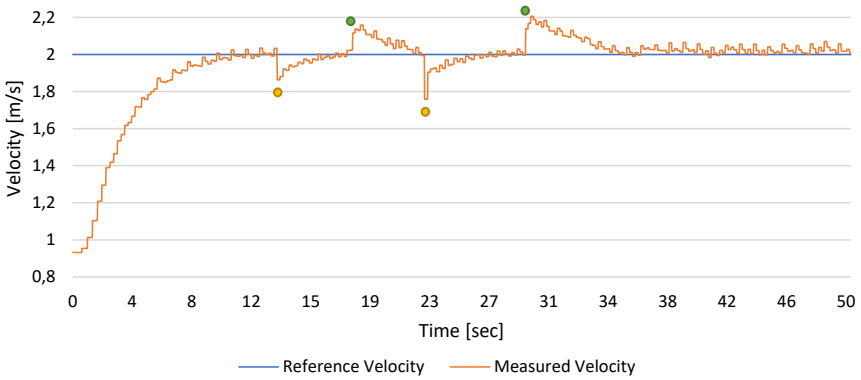


Figure 3.14: Forward velocity motor performance test (without touching the ground) with external resistance

In the second test (see Figure 3.15), performance of the speed controller is evaluated on the ground. Steering control of the bicycle is off, and steering angle is manually ensured to be 0 by holding with hand, and by running with the bicycle. Roll angle is also manually adjusted to be 0, by holding the bicycle. The bicycle reaches the reference velocity and then it can deliver the desired velocity with variation smaller than 0.1 m/s. Acceleration to 2.5 m/s, however, takes a time around 11 seconds.

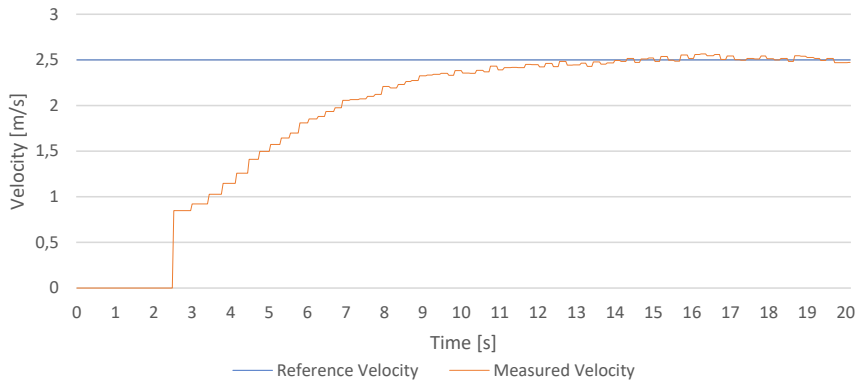


Figure 3.15: Forward velocity motor performance test on the ground with 0 roll angle and 0 steering angle

The test shown in Figure 3.15 is performed by holding the steering bar of the bicycle and running with it while steering angle and roll angle is adjusted to be (nearly) 0 all the time. This is not the case for the real self-balancing tests. In the real rides, steering angle and roll angle will be usually non-zero and they will usually change from time to time. Hence, another test is done, by rolling and steering the bicycle by hand quite often and with large angular deviations. The result is shown in Figure 3.16.

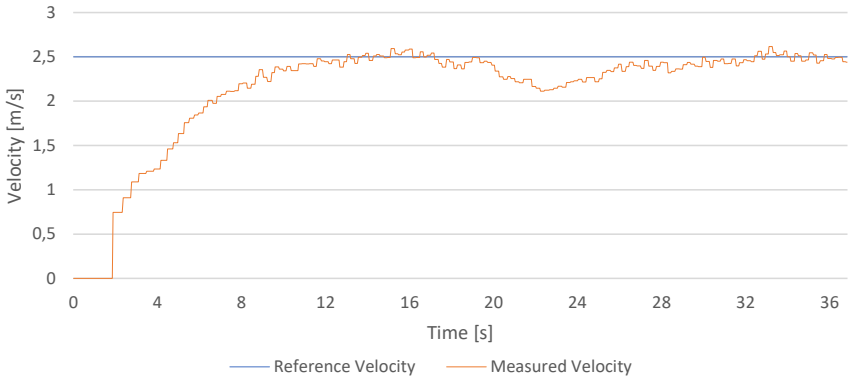


Figure 3.16: Forward velocity motor performance test on the ground with varying roll and steering angles

The velocity drop around 23^{rd} seconds is due to the fact that, bicycle is turned in a very sharp corner with radius of (approximately) 2 m (Because, the test is done at indoors with area limitations.). Even after that drop, and before that drop, forward reference of 2.5 m/s is followed with deviations smaller than 0.1 m/s. This validates the performance of the forward velocity motor under the existence of large roll and steering angle changes.

3.6 Validation of the Steering Motor Performance

In this section, steering motor performance is validated by several tests.

In the first test, steering motor input velocity is set in between -7.5 rad/s and +7.5 rad/s. Then, the velocity of the steering motor is measured. Reference versus measured steering motor speeds are shown in Figure 3.17.

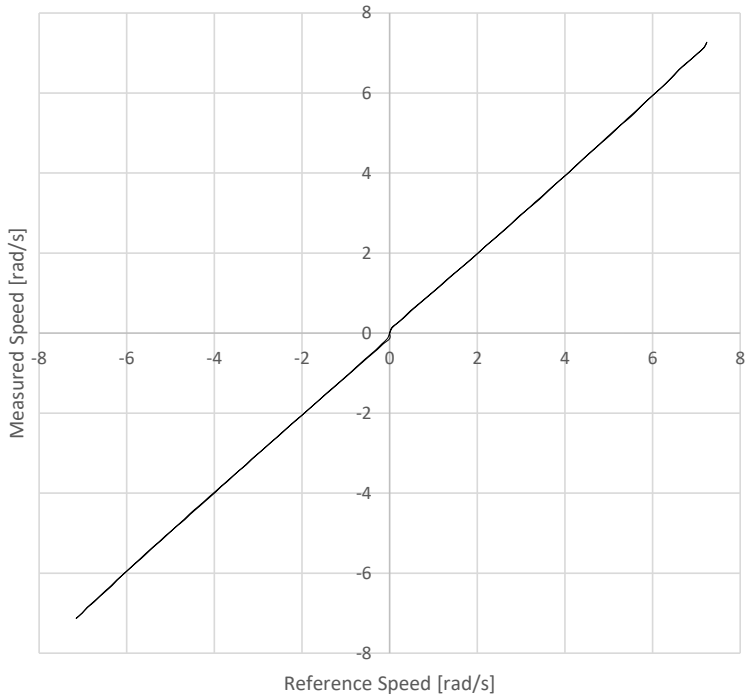


Figure 3.17: Reference vs measured steering motor speeds

Reference speed is achieved for all speeds except a very small amount of deadband. Furthermore, there is nearly no hysteresis. Deadband limits are measured as ±0.05 rad/s. In other words, if the desired speed of the steering motor is less than 0.05 rad/s, motor shaft does not rotate.

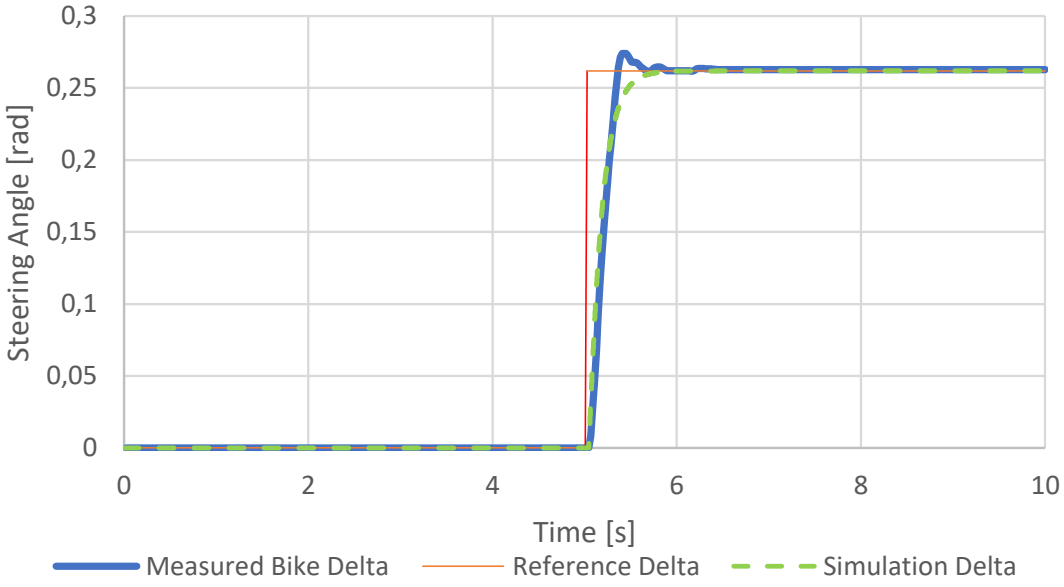


Figure 3.18: Step response (δ=+15°) test for the steering motor

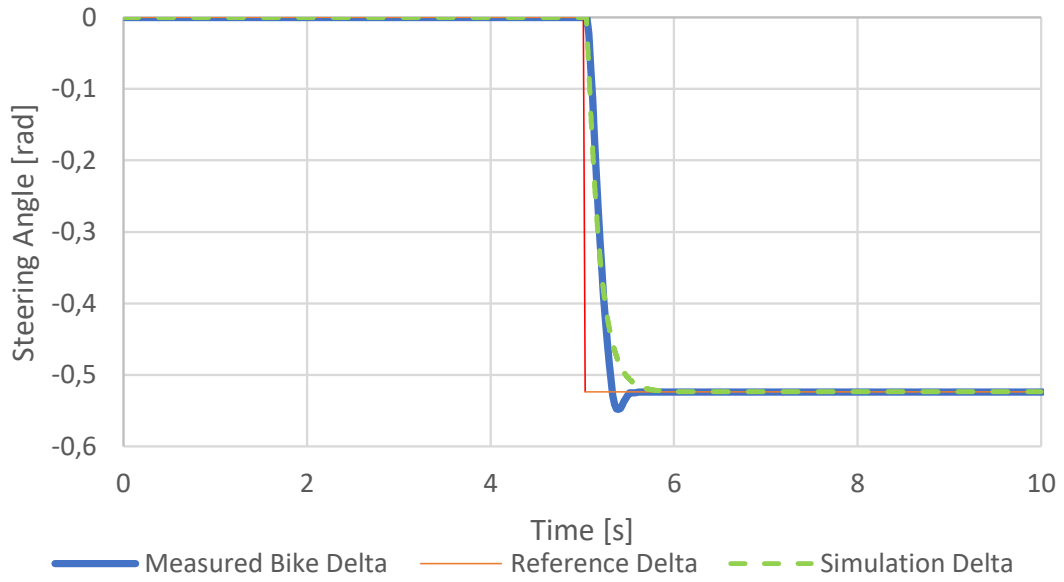


Figure 3.19: Step response (δ=-30°) test for the steering motor

In the second kind of test, a step input of steering angle reference is given to the steering motor and the closed-loop controller (self-balancing LQR explained in Chapter 4) of the bicycle is used.
In Figures (3.18) and (3.19), step response test results for +15° and -30° steering angles are shown. Tests are performed both in simulation and real bicycle. Simulation results are in agreement with real test results. Total response time for both tests are 0.8 seconds and there is 0.045 second delay. This delay is also included to the Simulink simulation.

4

Control and Software of the Bicycle

In this chapter, a brief overview of the control schemes used in the autonomous bicycle is presented first. Then control schemes are explained and discussed in detail.

4.1 Overview of the Control Schemes and Signal Connections

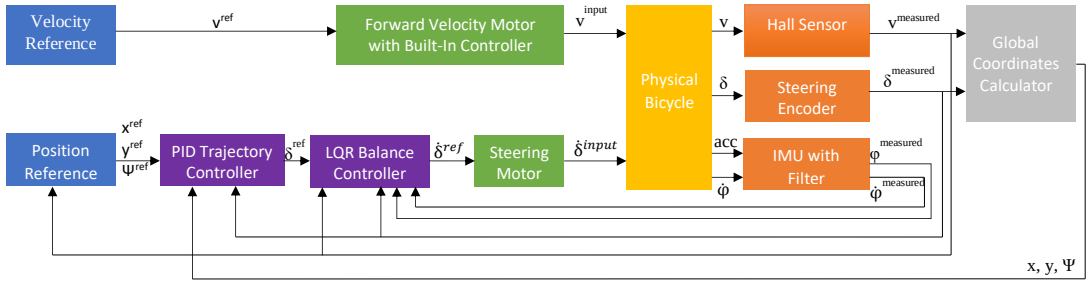


Figure 4.1: Subsystems of the autonomous bicycle system

In Figure 4.1, subsystems of the bicycle and signals in between them are shown. In the figure, there are 5 kinds of subsystems, except the bicycle itself:

- **Reference blocks** (blue) that the reference velocity and positions are supplied to the controllers. References are pre-defined by the user.
- **Controllers** (purple) for trajectory-tracking and self-balancing.
- **Actuators** (green) to run the bicycle in the forward direction and turn the steering bar.
- **Sensors** (orange) to measure the states.
- **Calculator** (gray) that calculates the global coordinates of the bicycle which are used for trajectory-tracking control.

Bicycle, motors and the sensors are the physical parts that exist in the real life. Reference blocks, controllers and the calculator are the components that are coded in the software of the bicycle.
Validations and explanations of the physical parts are given in the Chapter 3. In this chapter, control algorithm and software will be explained and discussed.

4.2 Calculation of the Global Coordinates

In this section, global coordinate calculation is presented.
In Figure 4.2, kinematic model of the bicycle is shown.

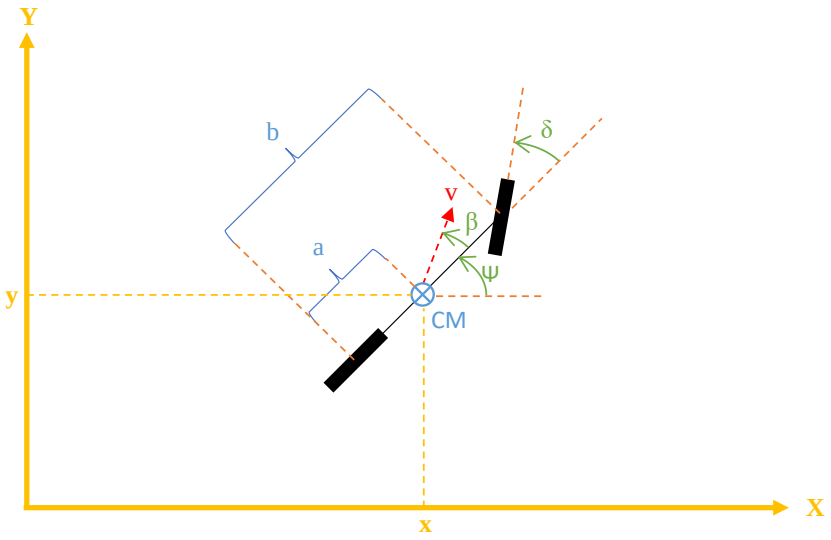


Figure 4.2: Kinematic bicycle model

β is the angle of velocity of the center of mass with respect to the longitudinal axis of the bicycle and defined as follows:

$$\beta = \tan^{-1}\left(\frac{a}{b}\tan\delta\right) \tag{4.1}$$

Yaw rate $\dot{\psi}$ can be derived as follows:

$$\dot{\psi} = \frac{v\tan\delta}{b} \tag{4.2}$$

Derivation of $\dot{\psi}$ is explained in Appendix A.1. ψ can be found by integrating $\dot{\psi}$.
Given x and y are the global coordinates of the bicycle as shown in Figure 4.2, their derivatives $\dot{x}$ and $\dot{y}$ can be found as follows:

$$\dot{x} = v\ \cos(\psi + \beta) \tag{4.3}$$

$$\dot{y} = v\ \sin(\psi + \beta) \tag{4.4}$$

x and y can be found by integrating $\dot{x}$ and $\dot{y}$.
In conclusion, by using (4.2), (4.3) and (4.4), global coordinates and yaw angle of the bicycle can be found.

4.3 Self-Balancing Controller

In this section, LQR controller that is used for the self-balancing is presented.
Differential equation and state-space equations of the bicycle were already introduced in (2.2) and (2.3) in Chapters 2.1.1 and 2.1.2, respectively. For the sake of readability, they are given again below:

$$\ddot{\varphi} = \frac{g}{h}\varphi + \frac{a\ v\ \sin\lambda}{b\ h}\dot{\delta} + \frac{(h\ v^2 - g\ a\ c)\ \sin\lambda}{b\ h^2}\delta \tag{4.5}$$

$$\dot{x} = \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}0 & 0 & 1\\ 0 & 0 & 0\\ \frac{g}{h} & \frac{(h\ v^2 - g\ a\ c)\ \sin\lambda}{b\ h^2} & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}0\\ 1\\ \frac{a\ v\ \sin\lambda}{b\ h}\end{bmatrix}u \tag{4.6}$$

where the states are $x_1 = \varphi$, $x_2 = \delta$, $x_3 = \dot{\varphi}$ and input is $u = \dot{\delta}$.
Furthermore, output equation is given below:

$$y = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} = \begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} \tag{4.7}$$

where y represents the output vector.
(4.5) is a linear dynamical system of second order with three real poles:

$$p_{1,2} = \pm\sqrt{\frac{g}{h}}\ ,\ p_3 = 0 \tag{4.8}$$

Poles are real numbers and found as ± 4.0915 and 0 for the particular bicycle used for this project. Since one of the poles is at the right half-plane in the complex plane, uncontrolled bicycle plant is unstable.
To balance the bicycle, an LQR (linear quadratic regulator) is used. Noting that state-space system ((4.6) and (4.7)) is controllable and observable, control law of the bicycle is

$$u = \dot{\delta} = -K\ x \tag{4.9}$$

where

$$K = \begin{bmatrix}K_\varphi & K_\delta & K_{\dot{\varphi}}\end{bmatrix} \tag{4.10}$$

By inserting (4.9) and (4.10) into (4.6), closed loop state-space system becomes:

$$\dot{x} = \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}0 & 0 & 1\\ -K_\varphi & -K_\delta & -K_{\dot{\varphi}}\\ \frac{g}{h} - K_\varphi & \frac{(h\ v^2 - g\ a\ c)\ \sin\lambda}{b\ h^2} - K_\delta & -K_{\dot{\varphi}}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} \tag{4.11}$$

By inserting (4.9) and (4.10) into (4.5), closed loop differential equation of the bicycle becomes:

$$\ddot{\varphi} = \frac{b\ g - a\ v\ \sin\lambda\ K_{\varphi}}{b\ h}\varphi - \frac{a\ v\ \sin\lambda\ K_{\dot{\varphi}}}{b\ h}\dot{\varphi} + \frac{(h\ v^2 - g\ a\ c - h\ a\ v\ K_{\delta})\ \sin\lambda}{b\ h^2}\delta \tag{4.12}$$

Cost function to be minimized by the LQR is given below:

$$J(u) = \int_0^{\infty} \left(x^T Q x + u^T R u\right) dt \tag{4.13}$$

where Q and R matrices are defined as:

$$Q = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 10 \end{bmatrix}\ ,\ R = 20 \tag{4.14}$$

Q stands for the costs of the states, and R stands for the cost of the input.

By using a sampling time of 0.04 s, and using the MATLAB command "lqrd" [7], a discrete full-state-feedback regulator is designed for an interval of velocities from 1.5 m/s to 10 m/s. Obtained LQR gains are shown in Figure 4.3.

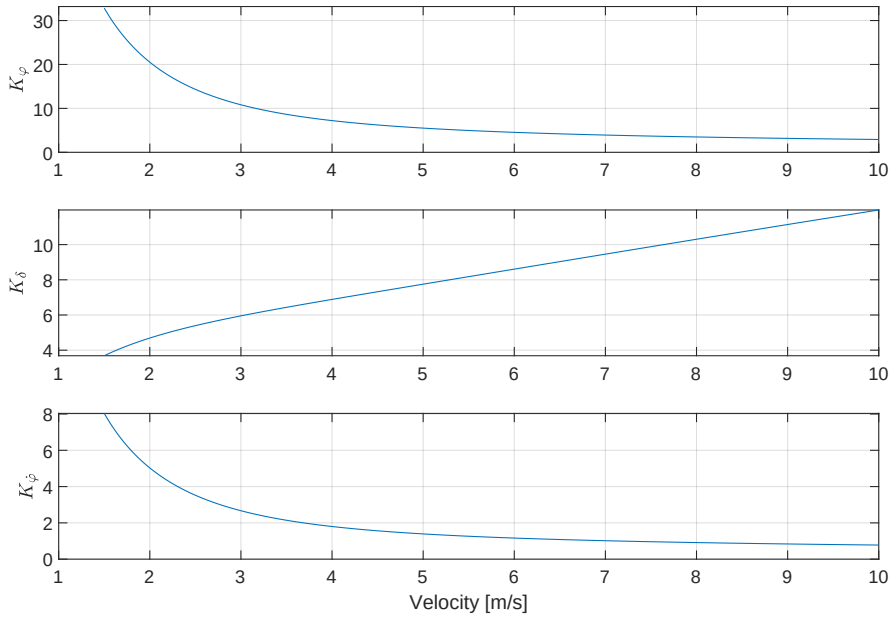


Figure 4.3: LQR controller gains for different velocities (discrete time)

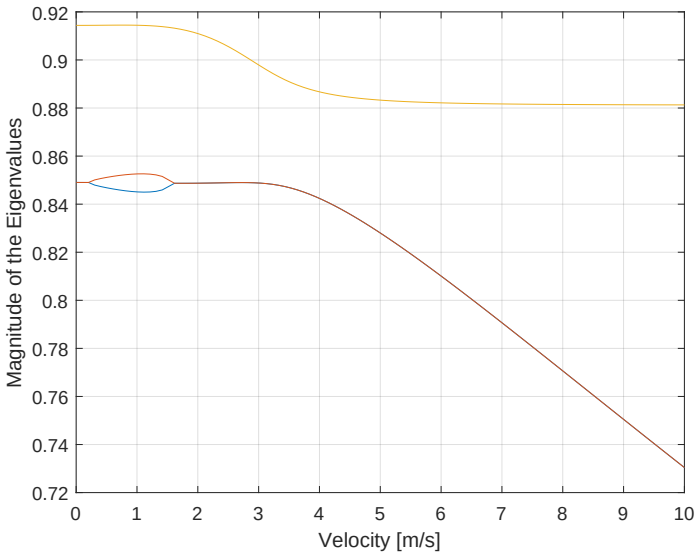


Figure 4.4: Magnitudes of the eigenvalues of the closed-loop bicycle plant for different velocities (discrete time)

For the stability analysis, the closed-loop system (4.11) is discretized and its eigenvalue magnitudes are shown in Figure 4.4. For a stable discrete system, its eigenvalues should lie in the unit circle, hence eigenvalue magnitudes should be smaller than 1. According to Figure 4.4, magnitude of the largest eigenvalue is always smaller than 0.92, hence closed-loop system is always stable. It is an expected fact, because LQR control law always gives optimal and stable controller. As velocity increases, eigenvalue magnitudes decrease, hence system becomes more stable.
Frequency domain analyses of the plant and feed-back loop are given in Appendix A.2.

4.4 Trajectory-Tracking Controller

For the trajectory-tracking controller, 3 cascaded PID controllers are used as shown in Figure 4.5 (Hence, "PID trajectory controller" block shown in Figure 4.1 is actually including 3 different PID controllers.).

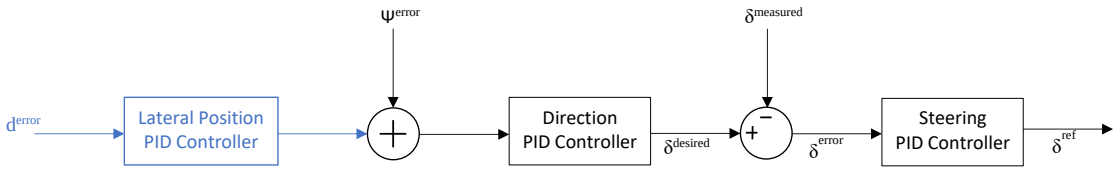


Figure 4.5: Cascaded PID controllers for trajectory-tracking

d^{error} is lateral error and can be calculated as:

$$d^{error} = \cos(\psi^{ref})(y - y^{ref}) - \sin(\psi^{ref})(x - x^{ref}) \tag{4.15}$$

where x and y define the calculated position of the bicycle (by using (4.3) and (4.4)); and x^{ref} and y^{ref} specify the reference position.
ψ^{error} is angular error and can be calculated as:

$$\psi^{error} = \psi - \psi^{ref} \tag{4.16}$$

where ψ defines the yaw angle of the bicycle (by using (4.2)); and ψ^{ref} specifies the reference yaw angle.
Considering only the components shown in black in Figure 4.5, direction PID controller is used to obtain the $\delta^{desired}$ from ψ^{error}. Then, δ^{ref} for the bicycle is found by using steering PID controller from $\delta^{error} = \delta^{desired} - \delta^{measured}$.
By using these 2 PID controllers, bicycle can follow a pre-determined path by using yaw angle feedback only. Using only yaw angle feedback can cause the bicycle to have a big lateral error with respect to the reference path. For this reason, a lateral position PID controller with the input of d^{error}, as shown with blue in Figure 4.5 can be added. The signal calculated from this lateral position PID controller is added to ψ^{error} signal. With the help of this method, input of direction PID controller includes the information about both lateral error and yaw angle error. Proportional, integral and derivative coefficients of the PID controllers are given in Appendix A.11.

4.5 Built-in Velocity Controller

For controlling the forward velocity, built-in controller of the Maxon ESCON Module 50/4 controller unit is used. Feed-back signal in this closed-loop controller is the voltage drop. Since the controller does not directly measure the real speed of the motor, this control method is not very precise, but it is a cheap and easy solution for the motor control [8].
To model the response of the motor in the simulation, a step reference input of 2 m/s is given, and the corresponding motor response is measured as shown in Figure 4.6.

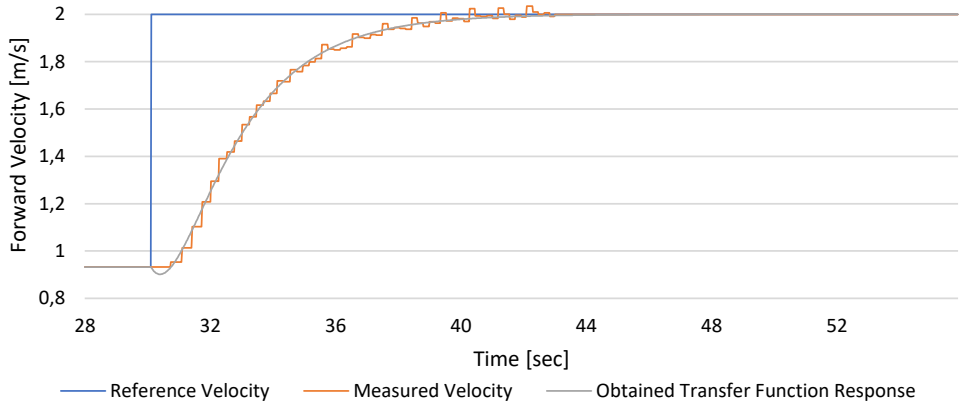


Figure 4.6: Measured velocity and obtained transfer function response plot

From the measured response, a 2^{nd} order transfer function to model the dynamics of the motor is derived by using System Identification toolbox of MATLAB [9]:

$$G_{motor}(s) = \frac{-0.2156s + 0.556}{s^2 + 1.678s + 0.5558} \tag{4.17}$$

The step response of the transfer function (4.17) is also shown in Figure 4.6.

5
Results

In this chapter, simulation and real test results are presented. Both self-balancing and trajectory-tracking performances are tested in simulations. In the real tests, only self-balancing test results are presented. At the end of the chapter, concluding remarks are given.

5.1 Simulation Results

In this section, several simulation results done in MATLAB Simulink software will be presented. Bicycle dimensions used for all of the simulations are given in Appendix A.3.1.
Furthermore, real (measured and recorded) noise data for the accelerometers and gyroscopes are used, as explained and shown in Appendix A.10. Other simulation configurations are presented in Appendix A.9.
First, self-balancing results for a constant forward velocity with no external disturbance will be shown. Then an external disturbance (which simulates pushing the bicycle to the side by hand) will be added. Then, a variable velocity will be added. Lastly, simulation will be started with non-zero states. Hence, complexity of the simulations will be increased step by step.
Then, trajectory-tracking simulations will be presented. The same variable speed and non-zero initial states are used also for the trajectory-tracking simulations. First, a straight line path reference will be used. Then circular paths will be discussed.

5.1.1 Self-Balancing Results

In this section, simulation results for only self-balancing tests are shown. In other words, there is no feed-back for the trajectory-tracking, and δ^{ref} shown in Figure 4.1 is 0 for all times. Hence, the aim of the controller is to make all states zero, which is the equilibrium point of the differential equation (2.2).
In Figure 5.1, states and input for the self-balancing test with a constant velocity of 2.5 m/s are shown. For the first 7 seconds, bicycle goes with zero true roll angle. But because of the non-zero mean accelerometer noise (shown in Appendix A.10), roll angle estimation grows up to 0.004 rad (0.23°). Due to non-zero estimated roll angle, LQR controller calculates a non-zero steering rate input, as shown in the last plot of Figure 5.1. However, due to dead-band of the steering motor (as described in the Appendix A.9), steering motor cannot turn for such a small steering rate.

When roll angle grows to 0.004 rad, calculated steering rate input goes beyond the dead-band limits of the steering motor, and steering motor starts to turn at 8^{th} second. This causes a non-zero true roll angle, and bicycle starts to roll.

LQR controller successfully balances the bicycle when it starts to roll. Delay and dead-band at the steering rate input can be seen at the last plot of the Figure 5.1. Noises of the roll rate and steering angle can also be seen from the same figure. Throughout the test, there is a difference about 0.004 rad in between true and measured roll angles because of non-zero accelerometer noise, but this difference does not result in failing of self-balancing test and can be compensated by the LQR controller.

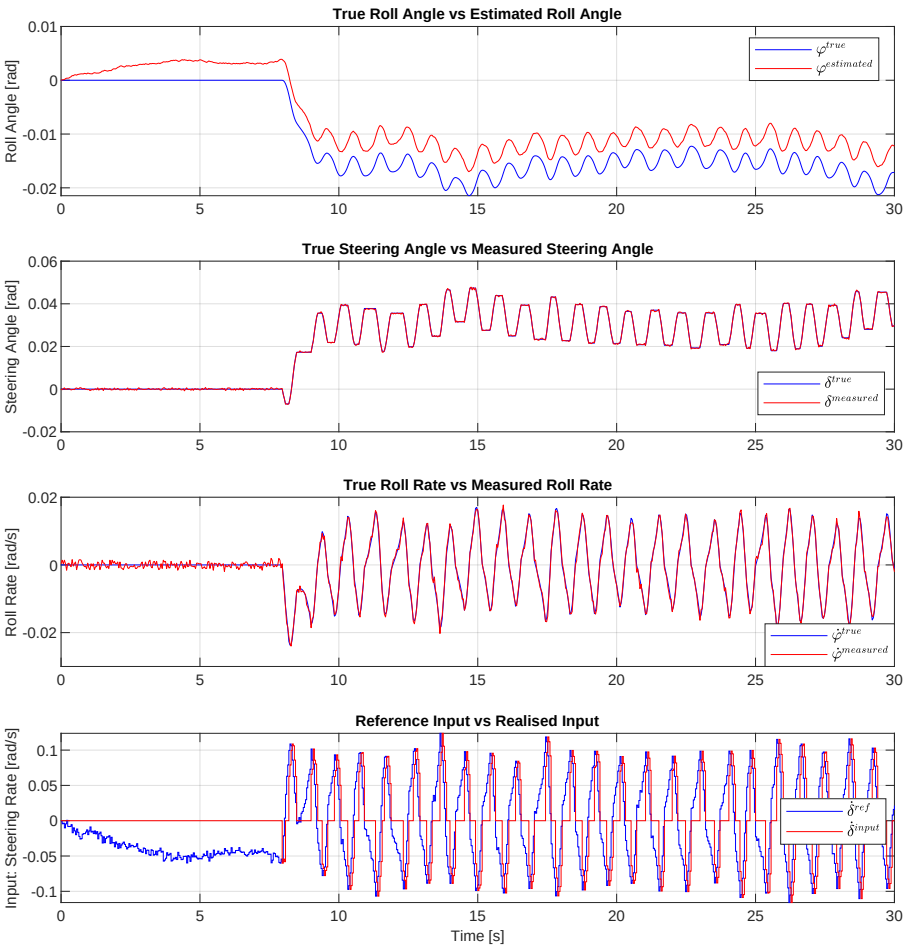


Figure 5.1: States and input at the self-balancing test

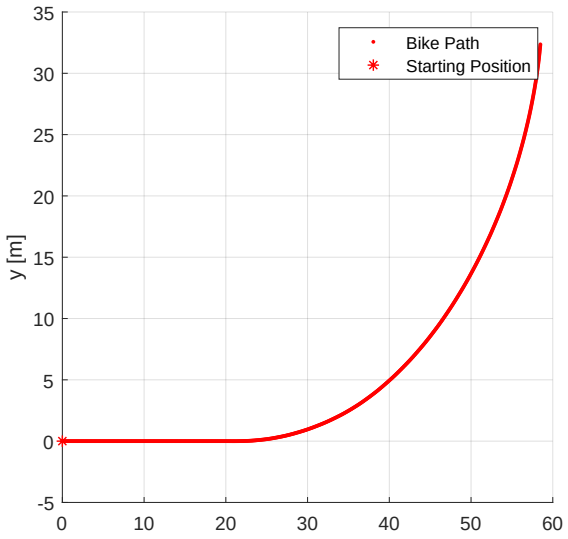


Figure 5.2: Path followed at the self-balancing test

In Figure 5.2, path followed during the self-balancing test is shown. For the initial 20 m, a straight path is followed, since no steering happens. But after the 20^{th} m (8^{th} second) bicycle starts to turn left, since steering motor starts to turn. It does not necessarily follow a straight line, since there is no trajectory-tracking controller used for this test.

5.1.1.1 Effect of External Disturbance

To test the performance of the LQR balance controller, an external roll acceleration disturbance is applied to the bicycle. Its magnitude is -0.52 rad/s^2 ($-30°$/s^2) and it is applied for 1 second with a period of 10 seconds as shown in Figure 5.3.

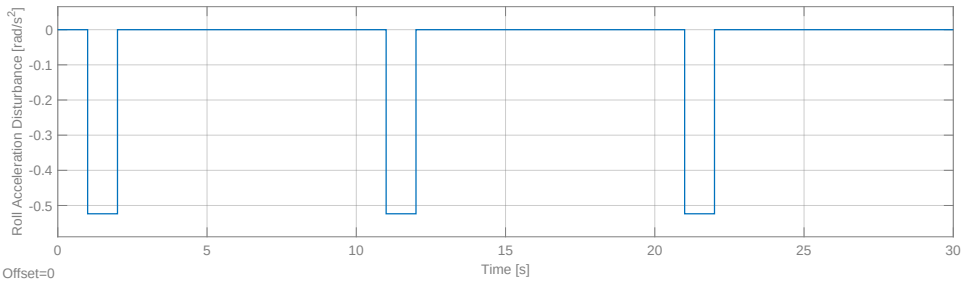


Figure 5.3: External Roll Acceleration Disturbance

As shown in Figure 5.4, LQR controller succeeds to stabilize the bicycle, although roll angle reaches up to -0.08 rad (-4.6°) at the times when external disturbance is applied. Steering rate input reaches up to 0.5 rad/s and steering angle reaches up to 0.21 rad (12°).
In Figure (5.5), path followed during the test is shown. 3 obvious break points of the path can be seen. Since the negative roll acceleration disturbance means bicycle is pushed to the left (according to the sign convention introduced in Figure 2.1), steering bar turns to the left to balance the bicycle. Hence, bicycle turns to left significantly at the times when disturbances applied.

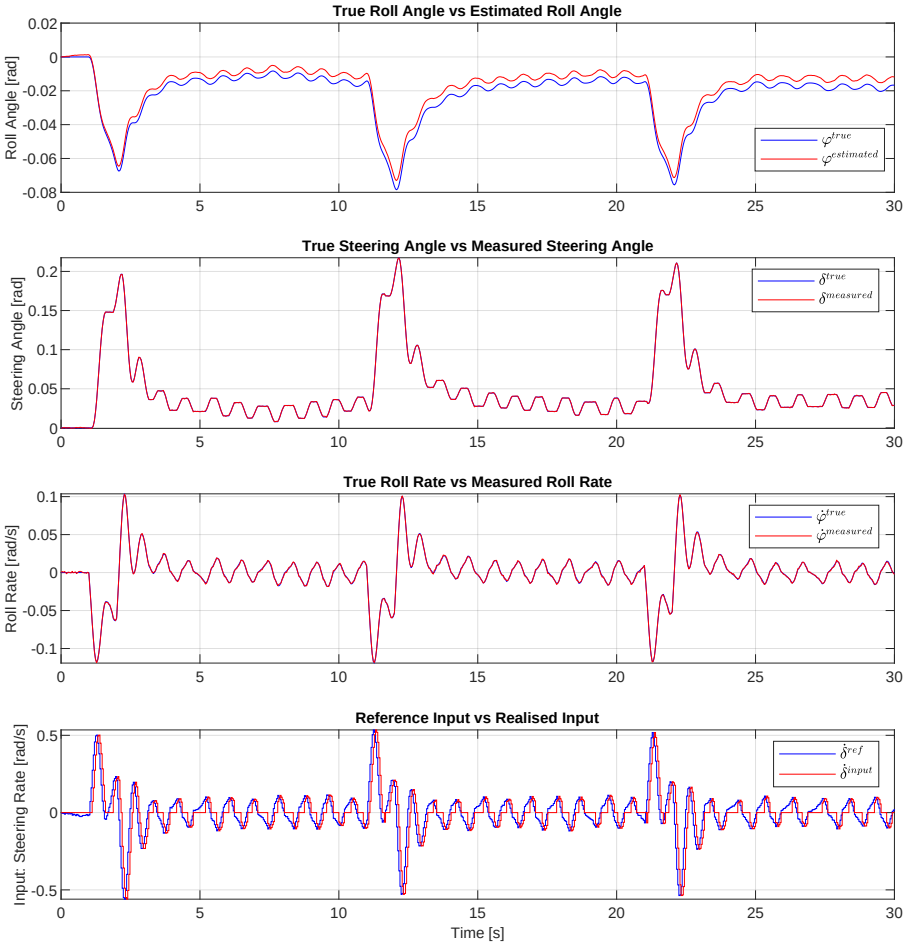


Figure 5.4: States and input at the self-balancing test with external disturbance

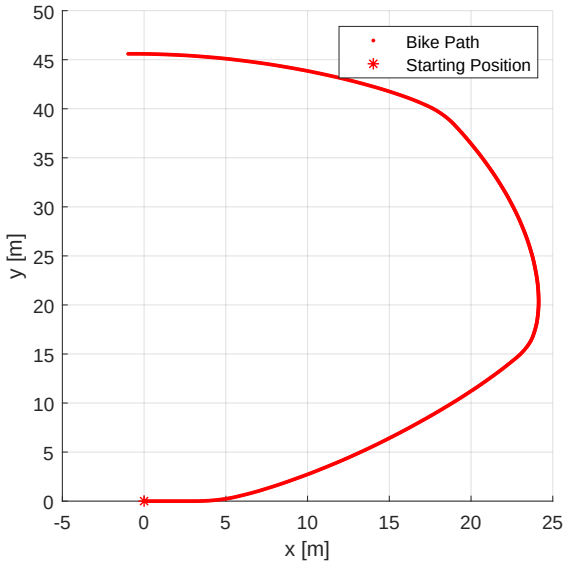


Figure 5.5: Path followed at the self-balancing test with disturbance

5.1.1.2 Effect of Variable Speed

This time, a variable speed of $2.5 + 0.5\sin(1.88t)$ m/s is used together with the disturbance shown in Figure 5.3. That means, a sinus wave with a magnitude of 0.5 m/s and frequency of 0.3 Hz is added to 2.5 m/s velocity. This variable speed is shown in Figure 5.6. Thus, a low performance velocity controller with oscillating velocity output is simulated.

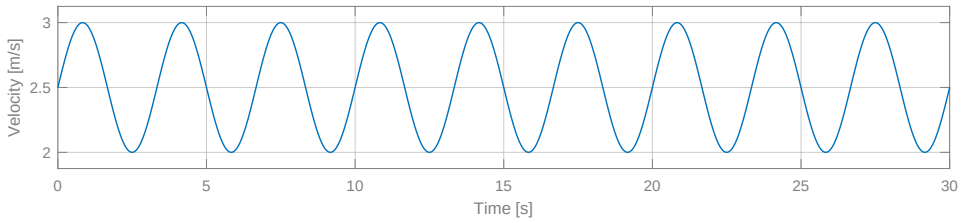


Figure 5.6: Variable speed used for the simulation

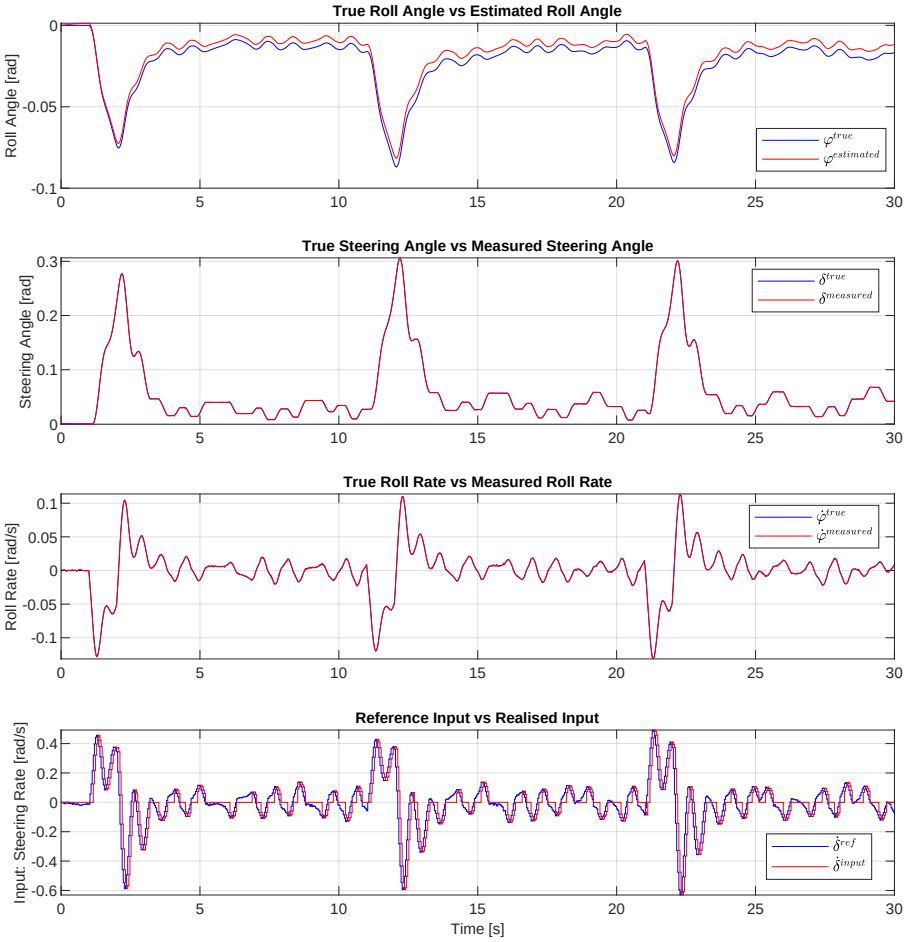


Figure 5.7: States and input at the self-balancing test with external disturbance and variable speed

States and input are shown in Figure 5.7 and followed path is shown in Figure 5.8. The results for variable speed are very similar to the results with a constant speed (Figures 5.4 and 5.5). Hence, bicycle can be stabilized with non-constant speeds since the LQR gains are calculated as a function of the forward velocity which changes the plant dynamics.

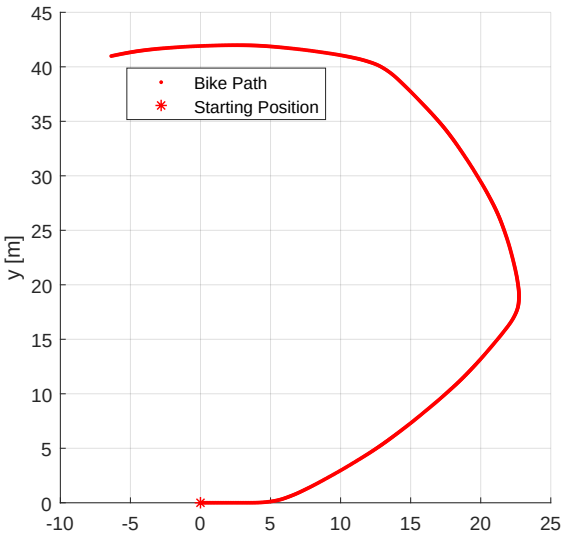


Figure 5.8: Path followed at the self-balancing test with disturbance and variable speed

5.1.1.3 Effect of Non-Zero Initial States

In the last scenario of the self-balancing simulation results, non-zero initial states are used, together with the disturbance shown in Figure 5.3 and variable speed shown in Figure 5.6. Initial roll angle and steering angles are set to 0.035 rad (2°) and roll rate is set to 0.035 rad/s (2°/s). Usage of non-zero initial states are more meaningful than the usage of zero initial states, since at the time when controller starts to work, bicycle may have a non-zero roll angle, roll rate and steering angle.

States and input are shown in Figure 5.9 and followed path is shown in Figure 5.10. As time passes, roll angle estimation gets close to the true roll angle, and non-zero initial states do no cause roll-over. LQR controller still successfully balance the bicycle.

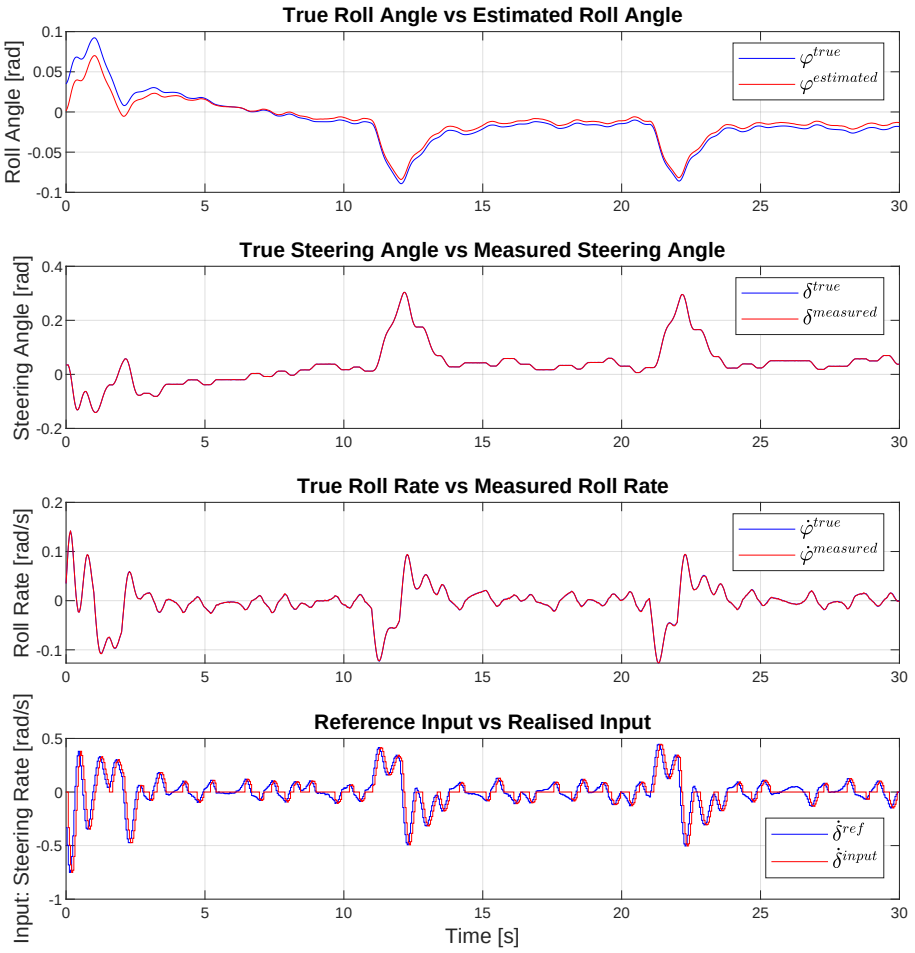


Figure 5.9: States and input at the self-balancing test with external disturbance, variable speed and non-zero initial states

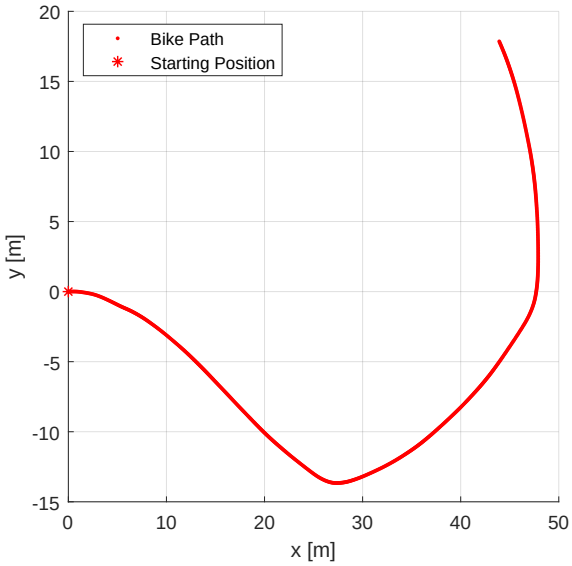


Figure 5.10: Path followed at the self-balancing test with disturbance, variable speed and non-zero initial states

5.1.2 Trajectory-Tracking Results

In this section, trajectory-tracking simulation results are presented. In other words, all controllers shown in Figure 4.1 is used: PID trajectory-tracking controllers (described in Section 4.4) are used to calculate the reference steering angle, and LQR self-balancing controller is used to ensure the self-balancing.
In all of the simulations presented in this section, velocity profile of the bicycle is same as the one shown in Figure 5.6. Initial roll angle and steering angles are set to 0.035 rad (2°) and roll rate is set to 0.035 rad/s (2°/s) for all simulations.

5.1.2.1 Straight Line Path

In the Figure 5.11, states and input are shown for the straight line path. After stabilizing the bicycle, roll angle is oscillating around ±0.03 rad (1.7°) and steering angle is oscillating around ±0.13 rad (7.4°). One can also see the clear delay in the input from the last plot of Figure 5.11.

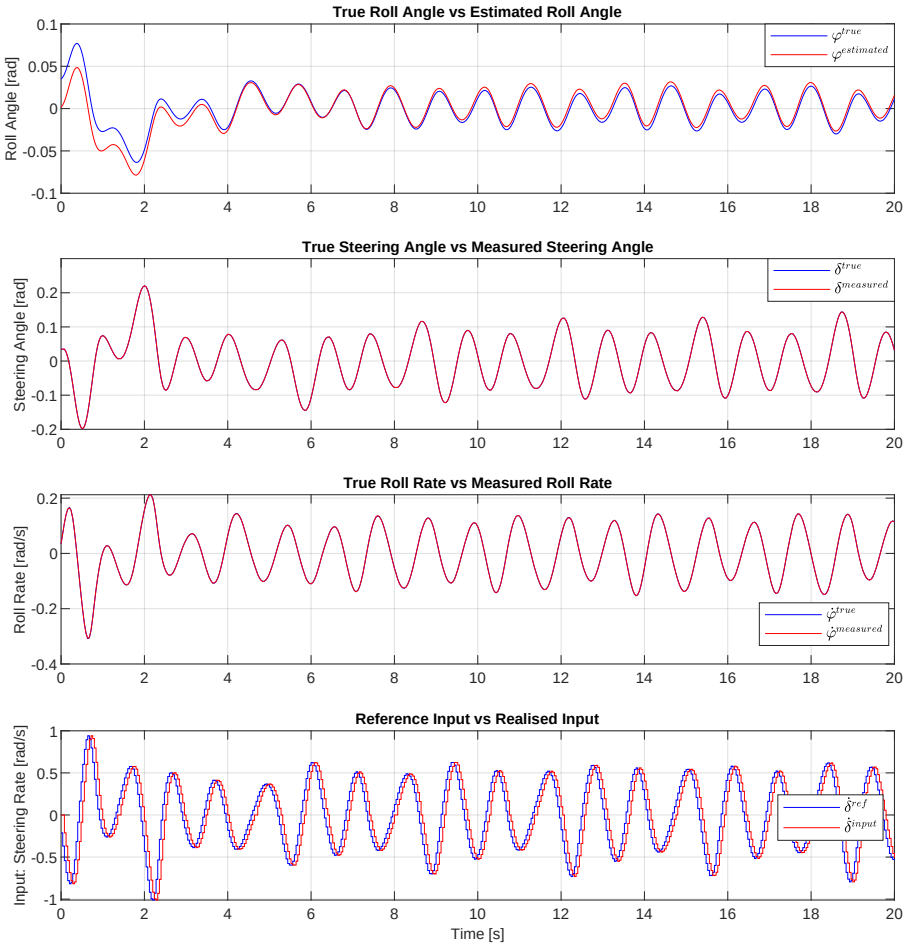


Figure 5.11: States and input at the trajectory-tracking test for a straight line

In the Figures 5.12 and 5.13 followed path and trajectory-tracking errors for the straight line reference are shown. At the very beginning, lateral error grows up to -0.5 m, but then it decreases to -0.1 m by time.

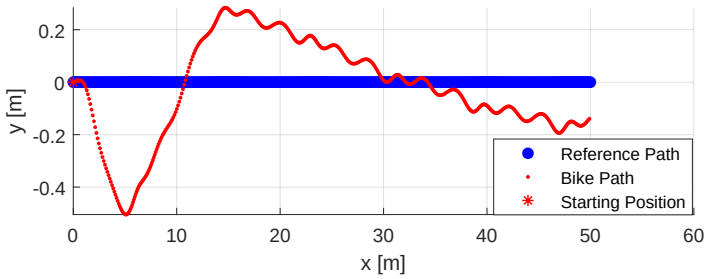


Figure 5.12: Path followed at the trajectory-tracking test for a straight line

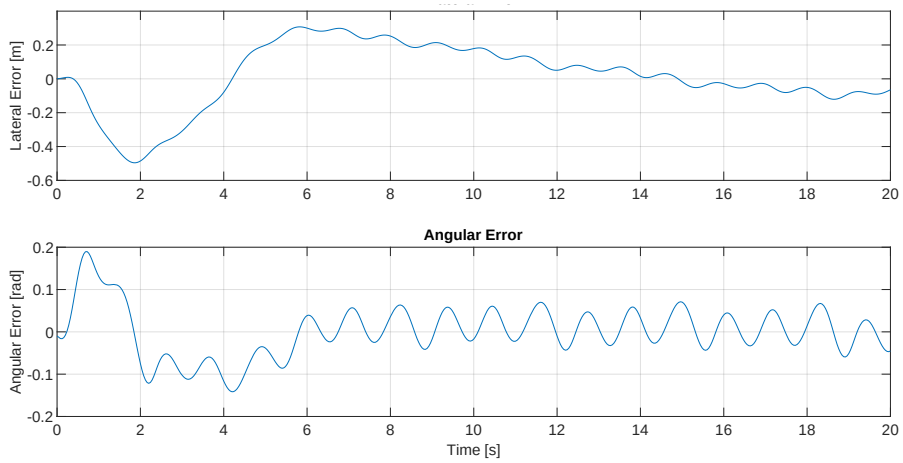


Figure 5.13: Trajectory-tracking errors for the straight line test

5.1.2.2 Circular Paths

In the Figure 5.14, states and input are shown for the circular path with 20 m radius. After stabilizing the bicycle, roll angle is oscillating in between 0.08 rad (4.6°) and 0 rad (0°). Steering angle, on the other hand, is oscillating in between 0.1 rad (5.7°) and −0.2 rad (−11.5°). Their averages are not 0 since the bicycle is turning on a circle.

In the Figures 5.15 and 5.16 followed path and trajectory-tracking errors for the circle (with 20 m radius) reference are shown. At the very beginning, lateral error grows up to 0.42 m, but then it decreases to 0.1 m by time and continues to oscillate around 0 m.

In the Figure 5.17, followed path for circles with 5 and 4 m are shown. For the circle with 5 m radius, lateral error grows up to 0.9 m at the beginning and then decreases to 0.25 m by time and continues to oscillate around 0 m. Hence, as circle radius becomes smaller, lateral error increases. 5 m radius is the minimum integer radius that the bicycle can follow with the existing controller parameters. Bicycle cannot follow the circle with 4 m radius and it rolls over at the 6^{th} second.

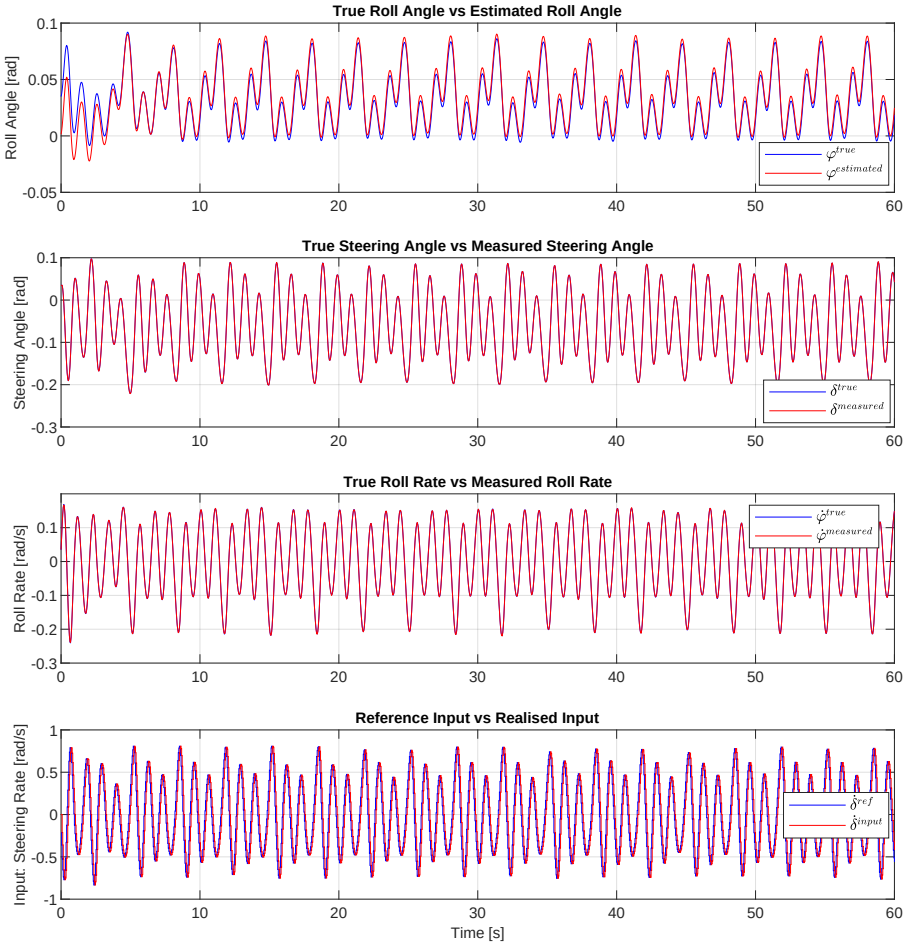


Figure 5.14: States and input at the trajectory-tracking test for a circle with 20 m radius

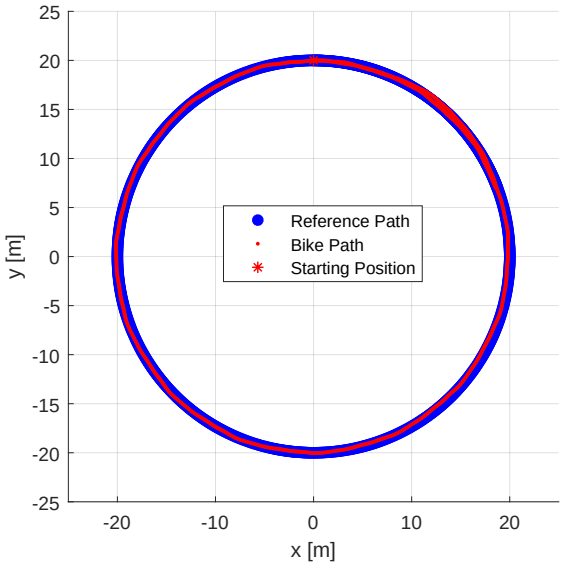


Figure 5.15: Path followed at the trajectory-tracking test for a circle with 20 m radius

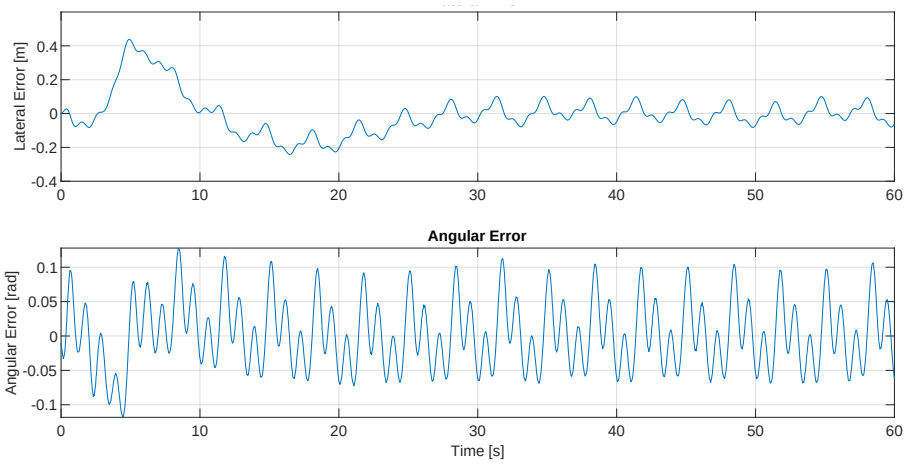
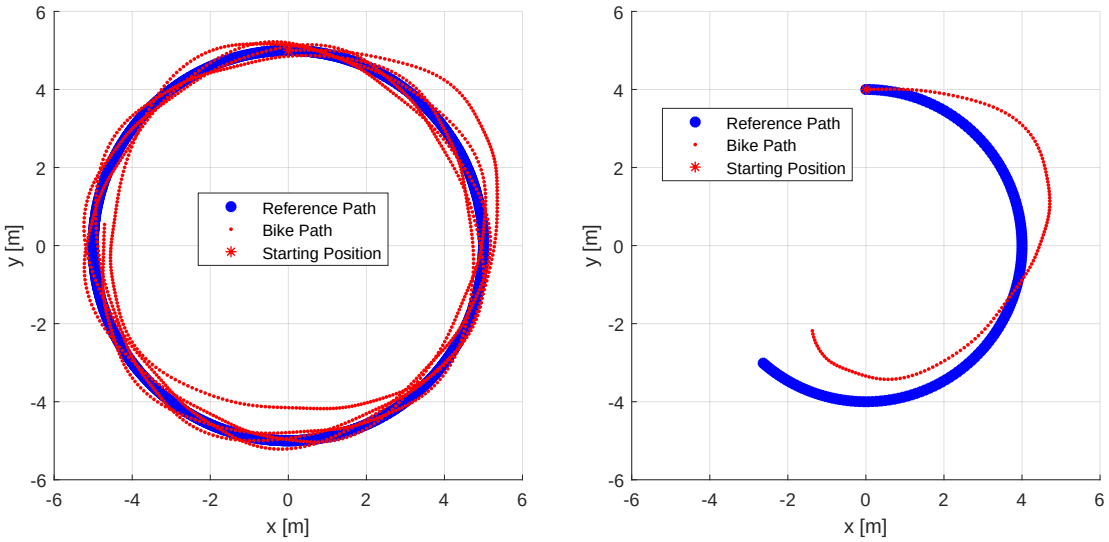


Figure 5.16: Trajectory-tracking errors for the circular path with 20 m radius



(a) 5 m radius **(b)** 4 m radius

Figure 5.17: Path followed at the trajectory-tracking tests for circles with different radii

5.2 Real Test Results

In this section, a self-balancing test done with the real bicycle will be presented. Test is performed with the bicycle described in Appendix A.3. To rapidly accelerate the bicycle, it is pushed by hand at the start of the test, without touching the steering bar. Reference forward velocity of the bicycle is set to 2.5 m/s.
Several different LQR cost matrices Q and R are tried on the real bicycle. The best performance is achieved with the following Q and R matrices:

$$Q = \begin{bmatrix} 10000 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 100 \end{bmatrix}, \; R = 30 \tag{5.1}$$

In Figures 5.18 and 5.19, roll angle estimations and roll rate measurements are shown. Throughout the test, roll angle does not exceed -0.07 rad (-4°). Negative roll angle means bicycle is leaned to the left, according to sign convention described in Figure 2.1. Roll rate of the bicycle is oscillating around 0 rad/s by reaching the limits at ±0.2 rad/s (11.5°/s).

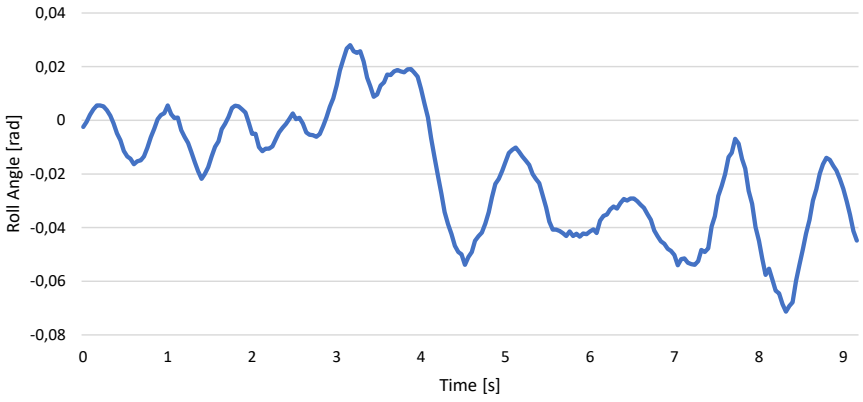


Figure 5.18: Roll angle estimations during the real test

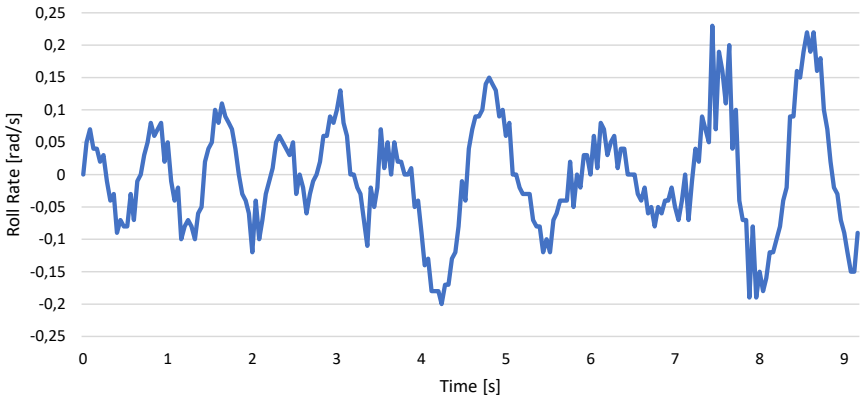


Figure 5.19: Roll rate measurements during the real test

Steering angle measurement is shown in Figure 5.20. Steering angle is positive in general, and it reaches the maximum value of 0.38 rad (21.8°). Positive steering angle indicates that the bicycle is steering to the left, according to the sign convention described in Figure 2.2. Steering to the left is expected, since the bicycle is rolled to the left.

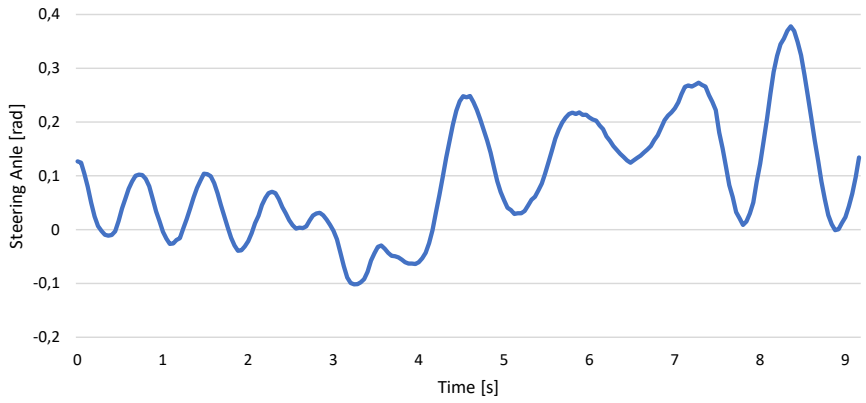


Figure 5.20: Steering angle measurements during the real test

In Figure 5.21, reference steering rate ($\dot{\delta}^{ref}$ in Figure 4.1) calculated according to LQR and measured steering rates ($\dot{\delta}^{input}$ in Figure 4.1) are shown (In fact, steering angle is measured by encoder and steering rate is calculated by integrating it.). Their trends are similar in general, but there is a delay. This input delay was already identified during the validation of the steering motor and included to the simulations, hence it is expected. Steering rate reaches up to ±1 rad/s at most, which is quite far from the maximum available speed of the steering motors (which is ±7.5 rad/s).

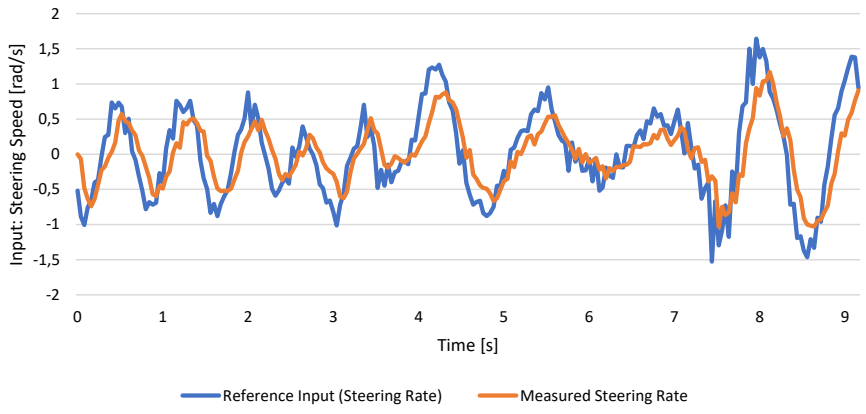


Figure 5.21: Reference steering rate input and measured steering rate during the real test

In Figure 5.22, followed path during the real test is shown (Path calculation is explained in Section 4.2). It is also clear from the steering angle plot (Figure 5.20) that bicycle is turning to the left. Since trajectory-tracking controller is not in use,

the bicycle is free to go anywhere, and it does not necessarily follows a specific path. In fact, even in the self-balancing simulations, bicycle was tending to turn to one side (as shown in Figure 5.2). Hence, it is expected to have such a trajectory.

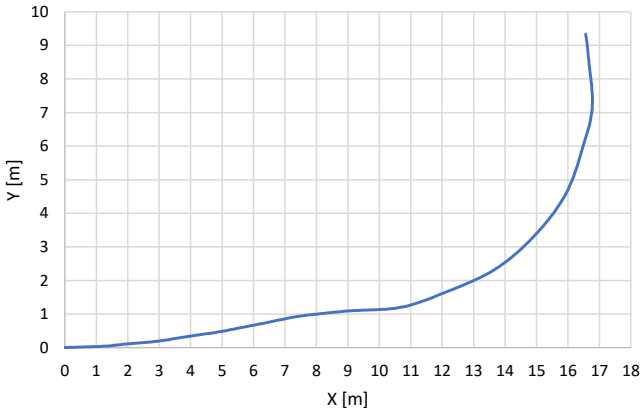


Figure 5.22: Followed path during the real test

In Figure 5.23, velocity measurement is shown. At first, bicycle is pushed by hand to help it to accelerate. Because, the test area is small and bicycle needs to reach the reference velocity (2.5 m/s for this test) as quick as possible. Even when the rear wheel is elevated, it takes more than 8 seconds to reach the reference speed (of 2 m/s) according to Figure 3.14 of Section 3.5. Hence, it is necessary to push the bicycle at the initialization. Furthermore, speed control is not consistent, and bicycle speed can decrease down to 2.3 m/s although the reference is 2.5 m/s.

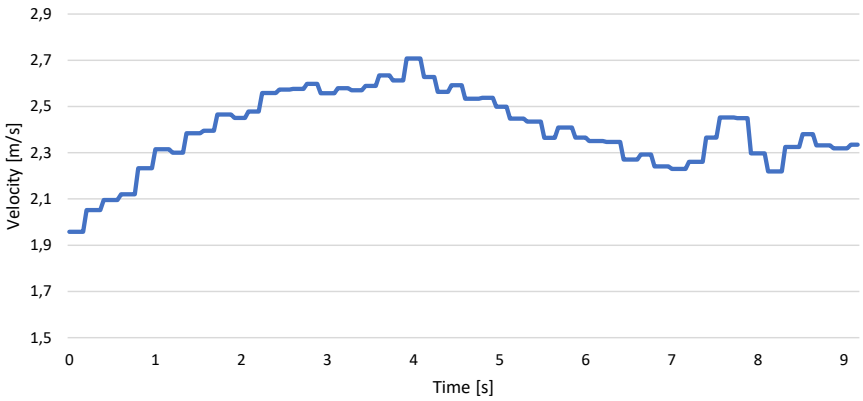


Figure 5.23: Velocity measurements during real test, reference velocity: 2.5 m/s

In Figure 5.24, velocity measurements for the real self-balancing test (shown in Figure 5.23) and speed control validation test (shown in Figure 3.16) are plotted together. Dashed lines illustrate where the data loggings start (For the blue line, there is a sudden increase at 3^{rd} second, since data logging started there.).

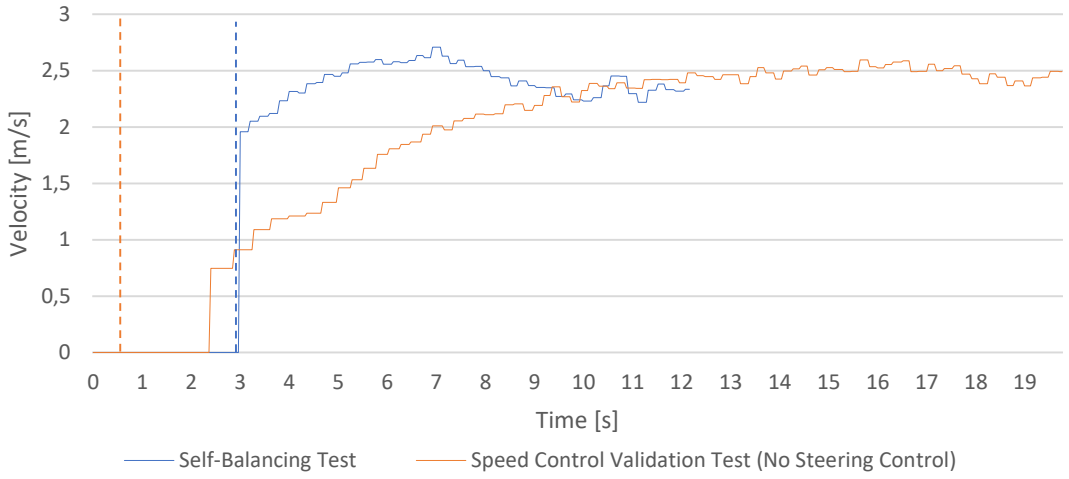


Figure 5.24: Velocity measurements during the real self-balancing test and speed control validation test, reference velocity=2.5 m/s for both tests

At the start of the self-balancing test (blue line), bicycle is pushed by hand to help it to accelerate. Then, forward velocity drops after 7^{th} seconds, according to Figure 5.24. However, according to speed control validation test (orange line), it takes 11 seconds for bicycle to reach the reference velocity of 2.5 m/s when it is not pushed. Hence, the reason of deceleration during the self-balancing test is that, forward velocity motor didn't reach the reference velocity yet, although the bicycle reached the reference velocity by the help of push. Around 11^{th} second, deceleration of the self-balancing test stops and bicycle starts to increase its speed, since motor reached the the bicycle speed at that time instance. Hence, a longer acceleration time (thus a larger test area and a longer test duration) is required for a more consistent self-balancing test.

5.3 Concluding Remarks

In this section, final comments on the simulation and real test results will be given. Self-balancing and trajectory-tracking performances of the bicycle are first evaluated in the simulation. All of the limiting factors, like input delay and motor dead-band are included to the simulation, as explained in Appendix A.9. However, it is not possible to include all of the factors that have impact on the real test.
In the simulations, it is shown that, an LQR controller can stabilize the bicycle, and the bicycle can follow a pre-determined path with the help of PID controllers. In the real tests, self-balancing is successfully proven.
Although both of the real results and simulation results show that LQR controller balances the bicycle, some differences in between the simulation results and the real test results are natural. Because,

- Disturbances existing in the real life (wind, bumps on the ground, etc.) may affect the performance.
- While some parameters may be easy to determine (wheel base, mass, etc.), some other parameters are hard to determine with a good precision (location of the center of mass, etc.). Hence, using inaccurate parameters at the simulations gives unrealistic results.
- Model used in the simulation (2.2) may not be good enough, more advanced models may be required. Bicycle model (2.2) used in the modelling in this project is a second order differential equation which is not including steering torque. It is not possible to catch some details with this model. For example, a bicycle can balance itself without any controller input if the forward speed is above some critical speed (which means the uncontrolled bicycle plant is stable in between certain speeds). Moreover, there are forth order bicycle models that make it possible to investigate the bicycle dynamics in more detail [6]. Another important point about the modelling of the bicycle is that, there are significant modifications on the weight distribution of the bicycle (since an electronic equipment box is located at the rear side of the bicycle). These modifications can affect the bicycle dynamics, which make the model (2.2) inaccurate. The bicycle used in this project is a female bicycle and has significant amount of frame elasticity. That is also important to consider in the modelling.

6
Conclusion

In this master's thesis project, modelling, validation and testing of a self-balancing bicycle is done.
Important aspects of this study are as follows:

- State estimator specifically derived for the autonomous bicycle is presented and validated.
- A methodology for validating all sub-systems of the bicycle is explained. All parts of the bicycle are validated by using this methodology.
- In the simulation environment, it is shown that
  - the bicycle can balance itself with steering input calculated by the LQR controller
  - the bicycle can follow a pre-determined trajectory with the PID controller
- In the real tests, self-balancing is successfully proven.

Certain aspects that need to be improved and possible future works are as follows:

- Speed control performance of the autonomous bicycle designed in this project is not decent. As a future work, closing the feed-back loop by using the Hall Effect sensor measurements should be considered. Furthermore, a more powerful motor controller unit should be used.
- Trajectory-tracking simulation results are presented, but real trajectory-tracking tests are not performed. A next step for the existing bicycle should be to perform the real trajectory-tracking tests.
- For the future work, trying different controllers and different plant models may be a good continuation. Particularly, steering torque can be included to the plant model. Furthermore, cascaded PID trajectory-tracking controller can be changed with another non-cascaded controller.

Bibliography

[1] H. Jiarui, S. Sotirios, Y. Yongchao, X. Kaiyuan, and C. Kai, “Auto-Balancing Bicycle,” 2015.

[2] A. Sharma, S. Wang, and Y. M. Zhou, “Towards a maximally-robust self-balancing robotic bicycle without reaction-moment gyroscopes nor reaction wheels,” no. September, pp. 0–17, 2016.

[3] F.J.W. Whipple, “The stability of the motion of a bicycle,” Quart. J. Pure Appl. Math., vol. 30, no. 120, pp. 312–348, 1899.

[4] E. Carvallo, “Théorie du mouvement du monocycle et de la bicyclette,”J. de L’Ecole Polytechnique, vol. 5, pp. 119–188, 1900.

[5] E. Carvallo, “Théorie du movement du monocycle. part 2: Theorie de la bicyclette,” J. d l’Ecole Polytechnique, vol. 6, pp. 1–118, 1901.

[6] Karl J Astrom, Richard E Klein, and Anders Lennartsson. “Bicycle dynamics and control: adapted bicycles for education and research”. In: IEEE Control Systems 25.4 (2005), pp. 26–47.

[7] "Design discrete linear-quadratic (LQ) regulator for continuous plant", Mathworks.com, 2019. [Online]. Available: https://se.mathworks.com/help/control/ref/lqrd.html. [Accessed: 28- Jan- 2019].

[8] "Technology – short and to the point", Maxonmotor.com, 2014. [Online]. Available: https://www.maxonmotor.com/medias/sys_master/root/8815459401758/control-Technology-short-and-to-the-point.pdf?attachment=true. [Accessed: 28- Jan- 2019].

[9] "System Identification Toolbox", Mathworks.com, 2019. [Online]. Available: https://www.mathworks.com/products/sysid.html. [Accessed: 28- Jan- 2019].

[10] H. Goldstein, C. Poole and J. Safko, Classical Mechanics. Addison Wesley, 2000, p. 187.

[11] InvenSense Inc., “MPU-6000 and MPU-6050 Product Specification Revision 3.4,” MPU-6050 datasheet, Nov. 2010 [Revised Aug. 2013].

[12] Honeywell, “Hall-effect Position Sensors with Sealed Housing,” 103SR Series datasheet, Mar. 2014.

[13] Maxon Motor, “Encoder HEDS 5540 500 CPT, 3 Channels,” Maxon Sensor datasheet, May 2017.

[14] Maxon Motor, “DCX 32 L Graphite Brushes DC motor Ø32 mm,” Maxon DC Motor datasheet, May 2017.

[15] Maxon Motor, “ESCON 50/5 Servo Controller,” P/N 409510 datasheet, Nov. 2015.

[16] Shimano, “Shimano Steps,” E6000 Series datasheet, Aug. 2017.

[17] Maxon Motor, “ESCON Module 50/4 EC-S Servo Controller,” P/N 446925 datasheet, Nov. 2015.
[18] G. Starr, "Calculating Motorcycle Center of Mass", Me.unm.edu, 2015. [Online]. Available: http://www.me.unm.edu/ starr/moto/cm.pdf. [Accessed: 29- Jan- 2019].
[19] Ksuweb.kennesaw.edu. [Online]. Available: http://ksuweb.kennesaw.edu/ plaval/-math4490/rotgen.pdf. [Accessed: 20- Feb- 2019].

A

Appendices

A.1 Derivation of the Equation (2.2) of Motion [3]

In this section, derivation of the equation of motion (2.2) will be done. For this purpose, firstly yaw rate of the bicycle will be derived. Then, equation of motion will be derived by using the angular momentum balance equation.

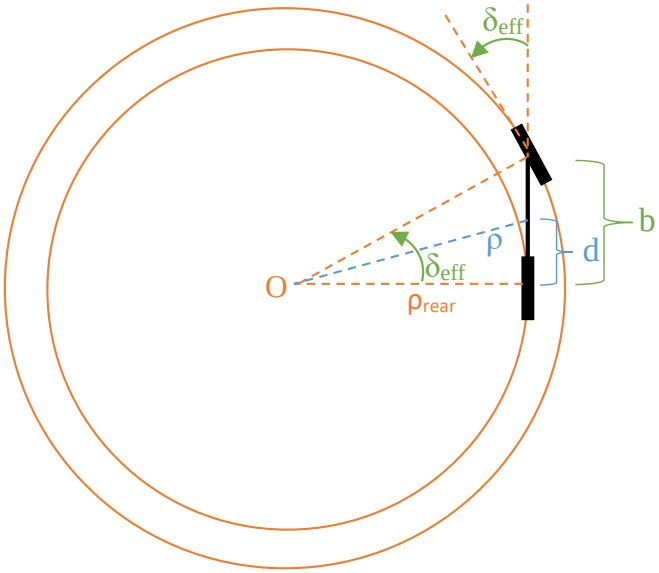


Figure A.1: Top view of the bicycle on a circular path

According to the Figure A.1, path radius of the rear wheel of the bicycle ρ_{rear} can be defined as

$$\rho_{rear} = b \cot \delta_{eff} \tag{A.1}$$

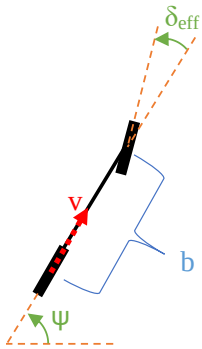


Figure A.2: Yaw angle definition

According to Figure A.2 and (A.1), yaw rate $\dot{\psi}$ can be defined by considering the contact point rear wheel as

$$\dot{\psi} = \frac{v}{\rho_{rear}} = \frac{v \tan \delta_{eff}}{b} \tag{A.2}$$

where δ_{eff} is the effective steering angle and defined as

$$\delta_{eff} = \delta \sin \lambda \tag{A.3}$$

Derivation and explanation of (A.3) is given in Appendix A.5.
The angular momentum of the system around the x axis is defined in (A.4) [10].

$$L_x = J_{xx}\omega_x + J_{xy}\ \omega_y + J_{xz}\ \omega_z \tag{A.4}$$

where J_{xx} is the moment of inertia of the bicycle around the x axis (shown in Figure 2.3), J_{xy} and J_{xz} denote the products of inertia for the x-y and x-z planes, respectively.
Since angular rate around x axis is roll rate ($\omega_x = \dot{\varphi}$), angular rate around z axis is yaw rate ($\omega_z = \dot{\psi}$) and ω_y is 0 for all times (*pitch motion is not allowed, bicycle frame is rigid and ground is flat*), (A.4) becomes

$$L_x = J_{xx}\dot{\varphi} + J_{xz}\ \dot{\psi} \tag{A.5}$$

Inserting (A.2) into (A.5) as

$$L_x = J_{xx}\dot{\varphi} + \frac{v\ J_{xz}}{b} \tan \delta_{eff} \tag{A.6}$$

Angular momentum balance of the system around x axis (shown in Figure 2.3) becomes

$$\frac{d\ L_x}{dt} = m\ g\ h\ \sin\varphi + \frac{m\ v^2\ h}{b} \tan \delta_{eff} - \frac{m\ g\ a\ c}{b}\delta_{eff} \tag{A.7}$$

where the green term is due to torque generated by the gravity and the blue term is due to centrifugal force. The red term is because of the geometry of the front fork: For a non-zero c, the center of mass of the bicycle frame is shifted when the front wheel is turned, which results an extra torque (red term) around the x axis.

Inserting (A.3) and (A.6) into (A.7), approximating $J_{xx} \approx m\ h^2$ and $J_{xz} \approx -\ m\ a\ h$ [6], and for the small angle approximation for both steering angle and the roll angle (such that $\tan\delta \approx \delta$ and $\sin\varphi \approx \varphi$),

$$\ddot{\varphi} - \frac{g}{h}\varphi = \frac{a\ v\ \sin\lambda}{b\ h}\dot{\delta} + \frac{(h\ v^2 - g\ a\ c)\ \sin\lambda}{b\ h^2}\delta \tag{A.8}$$

If λ and d are assumed to be 90° and 0, respectively, then (A.8) becomes

$$\ddot{\varphi} - \frac{g}{h}\varphi = \frac{a\ v}{b\ h}\dot{\delta} + \frac{v^2}{b\ h}\delta \tag{A.9}$$

A.2 Frequency Analysis

Consider the closed loop system shown in Figure A.3. Frequency analysis of the closed loop system can be done by considering the system as 2 parts:

1. Bicycle plant
2. Feed-back loop (everything except the bicycle plant)

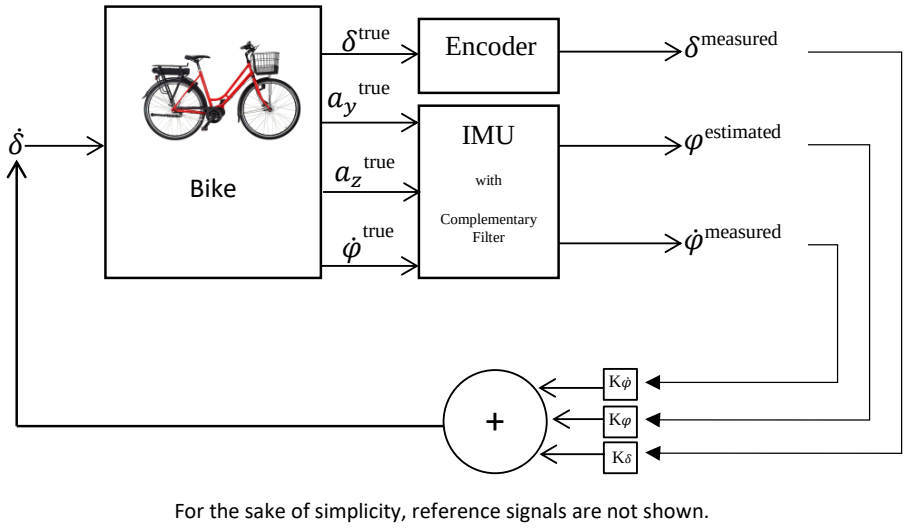


Figure A.3: Closed loop system with estimated roll angle feed-back

A.2.1 Frequency Analysis of the Bicycle Plant

In Figure A.4, Bode plots of the plant (described with the state-space system (2.3)) are shown for 2 different speeds: 1.5 m/s and 4 m/s:

- 1^{st} plot: system gain from steering rate input to true roll angle output
- 2^{nd} plot: phase shift from steering rate input to true roll angle output
- 3^{rd} plot: system gain from steering rate input to true steering angle output
- 4^{th} plot: phase shift from steering rate input to true steering angle output
- 5^{th} plot: system gain from steering rate input to true roll rate output
- 6^{th} plot: phase shift from steering rate input to true roll rate output

3^{rd} and 4^{th} plots are trivial plots, since they relate steering rate to steering angle. 5^{th} plot, on the other hand, is important for the frequency analysis of the bicycle. It gives the system gain from steering rate input to true roll angle output. It has low-pass filter characteristics with a cut-off frequency of 0.45 Hz. Thus, any steering input with a frequency higher than 0.45 Hz would be attenuated in roll dynamics.

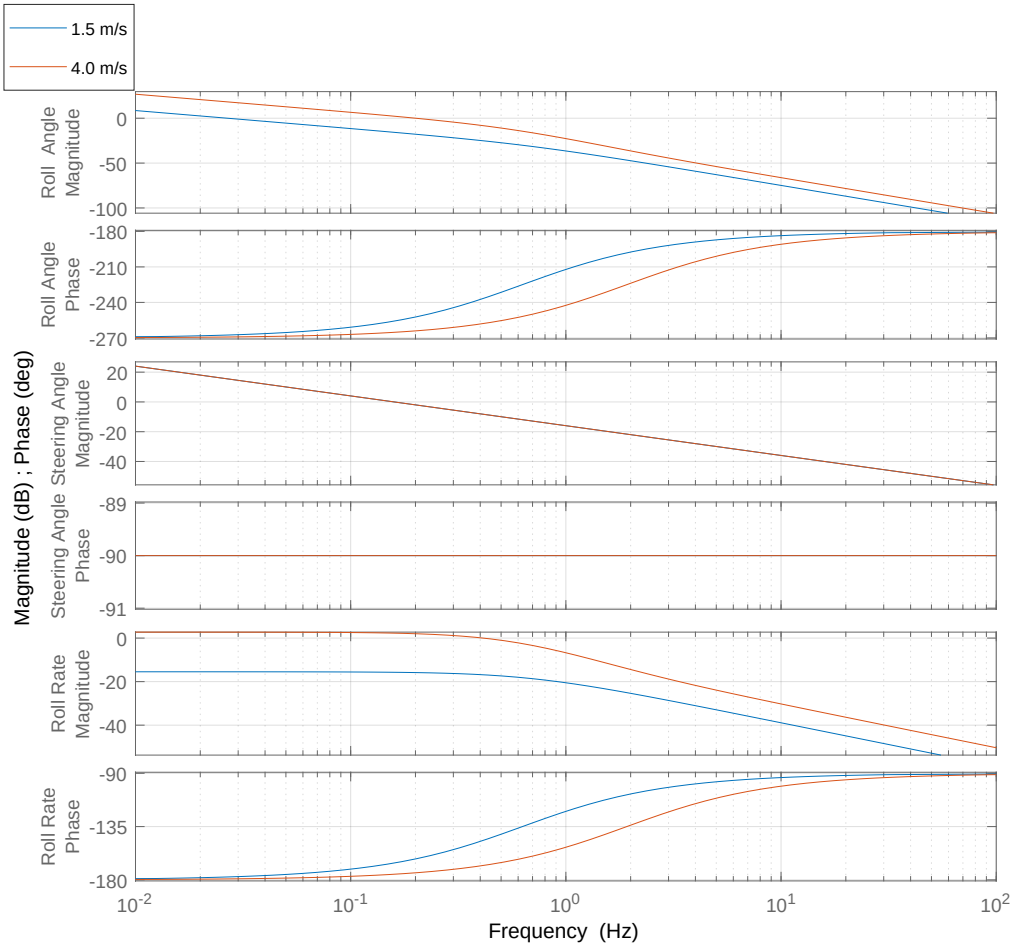


Figure A.4: Bode plot of the bicycle plant for 1.5 m/s and 4 m/s

A.2.2 Frequency Analysis of the Feed-Back Loop

In Figure A.5, Bode plots of the feed-back loop (from bicycle plant outputs to steering rate input as shown in Figure A.3) are shown for 2.5 m/s forward speed:

- 1^{st} column plots: system gain and phase shift plots from true roll angle to steering rate input
- 2^{nd} column plots: system gain and phase shift plots from true steering angle to steering rate input
- 3^{rd} column plots: system gain and phase shift plots from true roll rate to steering rate input

Feed-back loop basically consists of complementary filter and LQR controller. LQR controller acts like a constant gain, hence complementary filter determines the frequency characteristics of the feed-back loop.

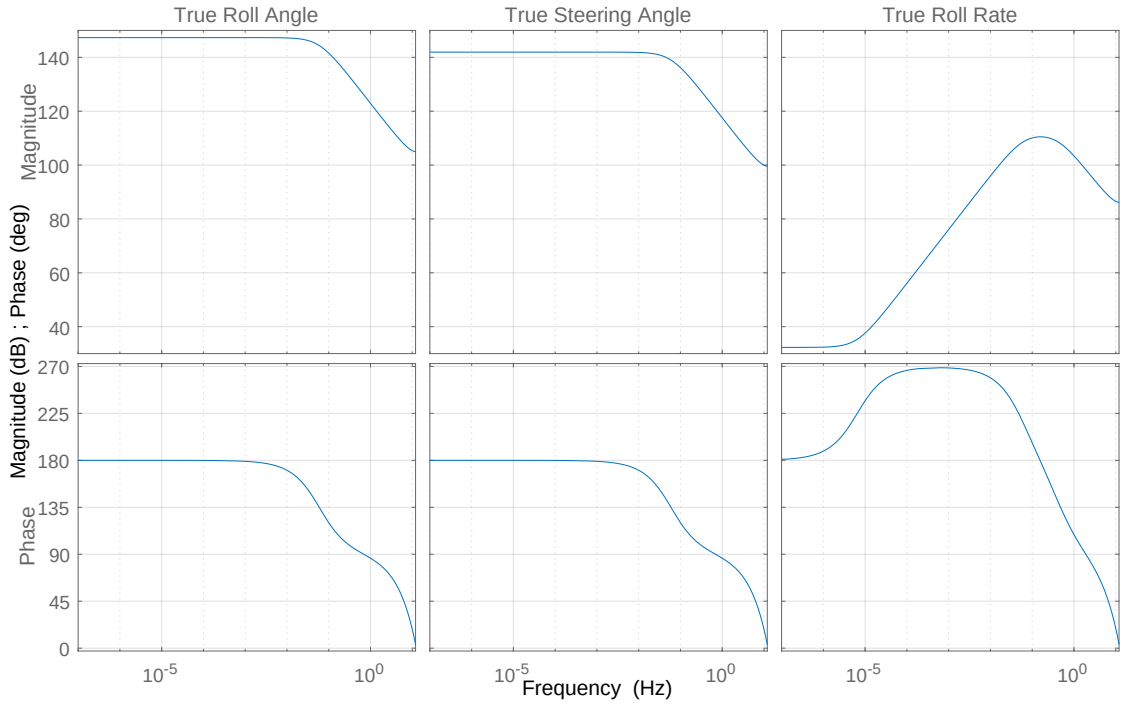


Figure A.5: Bode plot of the feed-back loop for 2.5 m/s

According to Figure A.5, magnitude plots of the first and second columns have low-pass filter behaviour with 0.05 Hz cut-off frequency. This behaviour is expected, because roll angle and steering angle terms are existing in the acceleration terms (see (A.16) and (A.17)), and accelerometer measurements are used with low-pass filter as explained in Figure A.11.
The magnitude plot of the third column of Figure A.5, on the other hand, has a high-pass filter-like behaviour, since roll rate is filtered with a high-pass filter in the complementary filter.

A.3 Components of the Bicycle

A.3.1 Bicycle

The bicycle model used in this project is "Skeppshult cykEL 7" designed for female bikers. It has Shimano STePS e-bicycle system and has 7 gears. Its electric motor is 250 W, 50 Nm Shimano motor.
Dimensions of the bicycle used in this project are given below, where the corresponding notation is explained in Figure 2.7:
$h = 0.586$ m (see [18] for the method to find center of mass location)
$b = 1.095$ m
$c = 0.06$ m
λ=70°
$a = 0.325$ m (see [18] for the method to find center of mass location)
h_{IMU}=0.45 m
$d = 0$ m (assumed to be 0, since its effect in (A.16) and (A.17) is negligible)

A.3.2 Microprocessors

Microprocessors used in this project and their connections to the rest of the electronic equipments are shown in Figure A.6.

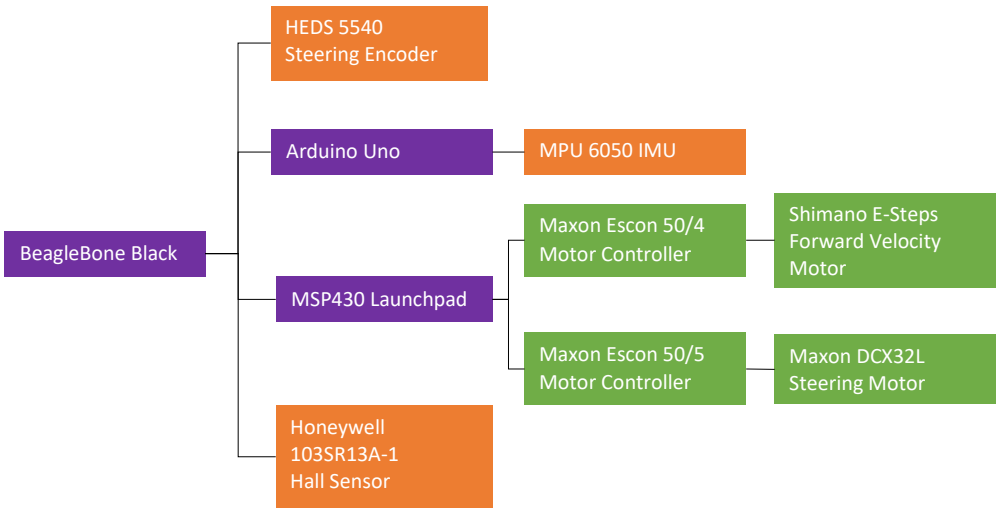


Figure A.6: Connection of the microprocessors and the rest of the electronic parts

- **BeagleBone Black:** It is used as the main computer of the autonomous bicycle. It takes all the measurements, calculates the control signal according to the control law, sends the inputs to motors (through MSP430), it records and stores the data.
- **Arduino Uno:** It is used in between IMU and BeagleBone Black. It takes the IMU data, processes it and sends to BeagleBone Black.
- **MSP430 Launchpad**: It takes the input signals for the motors from the BeagleBone Black, processes the signals and sends them to the motor controller units.

A.3.3 Parameters of the Steering Motor

- Speed constant = 350 RPM/V
- Thermal time constant winding = 42.2 s
- Maximum permissible speed = 8000 RPM
- Nominal current = 4.12 A
- Maximum output current = 10 A

Mode of operation of the steering motor is chosen as “open loop speed control” from ESCON Studio software. PWM method is used for set values:

- Speed at 10% : −8000 RPM
- Speed at 90% : +8000 RPM

A.3.4 Parameters of the Forward Velocity Motor

- Speed constant = 814.2 RPM/V
- Thermal time constant winding = 4.0 s
- Number of pole pairs = 1
- Minimum forward speed = 0.8 m/s (3600 RPM)
- Maximum speed = 3.9 m/s (18000 RPM)
- Acceleration and deceleration ramp: 3000 RPM/S
- Maximum permissible speed = 18000 RPM
- Nominal current = 4 A
- Maximum output current limit = 10 A

Mode of operation of the steering motor is chosen as “closed loop speed control” from ESCON Studio software. Set value type is “RC servo set value”.

A.4 Derivation of the IMU Output Equation (2.7)

In this section, IMU output equation (2.7) will be derived. Namely, there are 6 quantities to be found: accelerometer measurements in 3 axes, and gyroscope measurements in 3 axes.
Consider the dimensions and the coordinate frame given in Figure A.7.

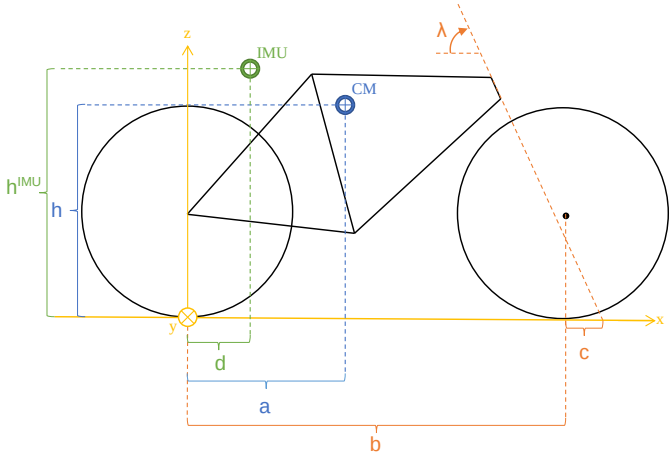


Figure A.7: Dimensions of the bicycle with the IMU

"CM" is the location of the center of mass, and "IMU" is the location of the inertial measurement unit. Measurements will be found by considering IMU location, and its own coordinate frame $x_{IMU}y_{IMU}z_{IMU}$ as shown in Figure A.8.

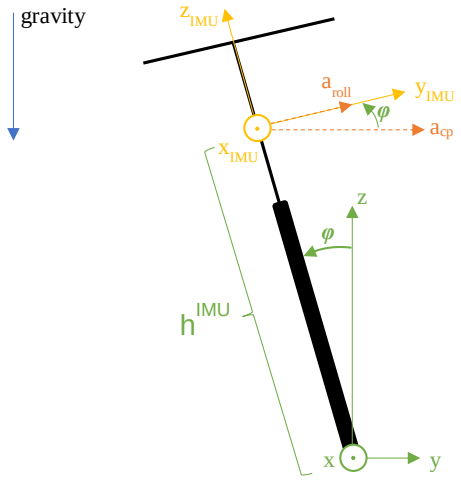


Figure A.8: Front view of the bicycle

According to Figure A.8, 2 different acceleration components are available in lateral direction:

- Centripetal acceleration due to steering (a_{cp}),
- Lateral acceleration due to roll (a_{roll}).

In the following, firstly these 2 acceleration components will be derived. Then sensor measurements will be formulated.

A.4.1 Derivation of Centripetal Acceleration

Centripetal acceleration a_{cp} exists when the bicycle is steering to any direction, and its formulation is given below.

$$a_{cp} = \dot{\psi}^2 \ \rho \tag{A.10}$$

where $\dot{\psi}$ is the yaw rate of the bicycle, which is given in (A.2) and ρ is the radius of curvature at the IMU placement location, described in Figure A.1. For the details of their derivations, revisit Appendix A.1.
ρ can be found by considering Figure A.1 and (A.1)

$$\rho = \sqrt{\rho_{rear}^2 + d^2} = \sqrt{b^2 \cot^2 \delta_{eff} + d^2} \tag{A.11}$$

where δ_{eff} is the effective steering angle and defined as

$$\delta_{eff} = \delta \sin \lambda \tag{A.12}$$

Derivation and explanation of (A.12) is given in Appendix A.5.
Inserting Equations (A.11) and (A.2) into (A.10), centripetal acceleration is found as:

$$a_{cp} = \frac{v^2}{b^2} \tan^2 \delta_{eff} \sqrt{b^2 \cot^2 \delta_{eff} + d^2} \tag{A.13}$$

A.4.2 Derivation of Lateral Acceleration due to Roll Motion

When the bicycle has roll acceleration, the location where the IMU is located has lateral translational acceleration a_{roll} as shown in Figure A.8. Its formulation is given below:

$$a_{roll} = -\ddot{\varphi} \ h_{IMU} \tag{A.14}$$

Minus indicates that for the positive roll acceleration $\ddot{\varphi}$, lateral acceleration is negative (opposite to the direction shown in Figure A.8).

A.4.3 IMU Output Equations

Accelerometer reading in x_{IMU} (forward) direction is the forward acceleration of the bicycle:

$$a_x = \dot{v} \tag{A.15}$$

According to Figure A.8, lateral acceleration due to roll motion (a_{roll}) lies only on y_{IMU} (lateral) axis. Centripetal acceleration (a_{cp}), on the other hand, has both y_{IMU} (lateral) axis and z_{IMU} (vertical) axis components. Hence, y and z components of the accelerometer readings are defined as:

$$a_y = -h_{IMU} \ \ddot{\varphi} + \left[\frac{v^2}{b^2} \ \tan^2 \delta_{eff} \ \sqrt{b^2 \ \cot^2 \delta_{eff} + d^2} \right] \ \cos \varphi + g \ \sin \varphi \tag{A.16}$$

$$a_z = -\left[\frac{v^2}{b^2}\ \tan^2\delta_{eff}\ \sqrt{b^2\ \cot^2\delta_{eff} + d^2}\right]\ \sin\varphi + g\ \cos\varphi \tag{A.17}$$

Last terms of (A.16) and (A.17) are the components of gravitational acceleration. Note that, d^2 in (A.16) and (A.17) is much smaller than $b^2\cot^2\delta_{eff}$ and can be neglected.
Yaw rate of the bicycle was already found in (A.2). Considering the coordinate frames shown in Figure A.8, yaw rate will always be parallel to z axis shown with green, hence it will have 2 components on the y_{imu} and z_{imu} depending on the roll angle φ:

$$\omega_y = \frac{v\tan\delta_{eff}}{b}\sin\varphi \tag{A.18}$$

$$\omega_z = \frac{v\tan\delta_{eff}}{b}\cos\varphi \tag{A.19}$$

Note that, it is assumed that road is flat surface (no slope), bicycle frame is rigid with no suspension. Hence, pitch motion (rotation around y axis of the bicycle) is not possible.
Lastly, gyroscope reading in x direction is directly equal the roll rate:

$$\omega_x = \dot{\varphi} \tag{A.20}$$

Note that, (A.20) does not depend on the height of the IMU, h_{IMU}; since the angular speed of a rigid body (bicycle frame) is same for all the points on the rigid body.
Considering (A.15), (A.16), (A.17), (A.18), (A.19) and (A.20); (2.7) can be written in terms of state and input variables.
$\ddot{\varphi}$ in (A.16) is derivative of a state, and can be replaced by the right hand side of (2.2) to make the output equations including only states and input.

A.5 Effective Steering Angle

In this section, meaning and derivation of the effective steering will be explained. In Figure A.9, steering angle δ and effective steering angle δ_{eff} are shown. For this section, a new coordinate frame xyz will be used for the sake of simplicity. xyz frame is obtained by moving the center of the xyz frame (shwon in Figure 2.2) to the front wheel hub, without rotating it.

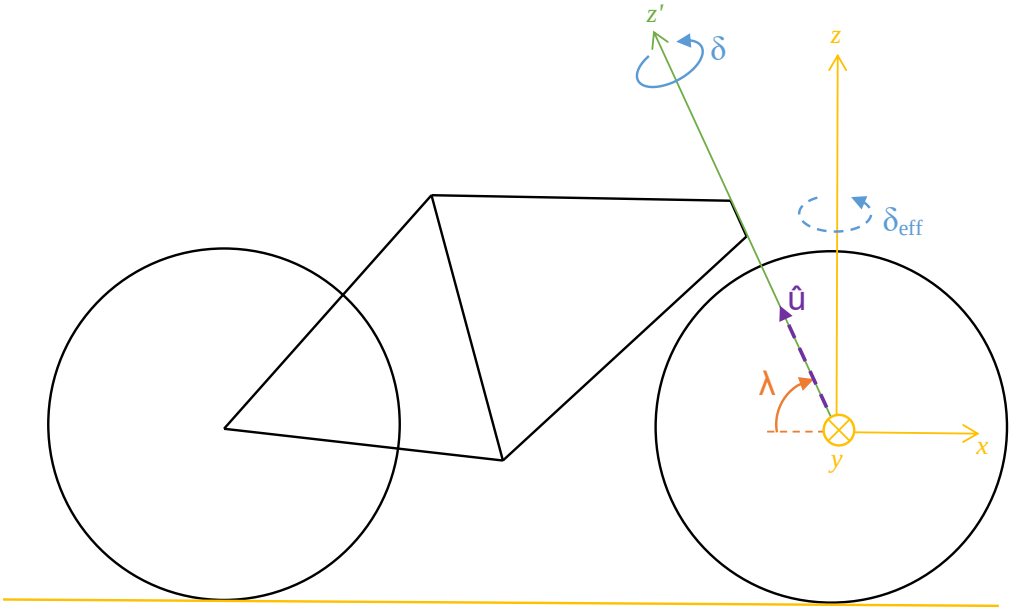


Figure A.9: Effective steering angle

According to Euler's rotation theorem, rotation matrix for a rotation of δ about z' axis is defined as below [19]:

$$R_{z'}(\delta) = \begin{bmatrix} (1-\cos\delta)u_x^2 + \cos\delta & (1-\cos\delta)u_x u_y - \sin\delta u_z & (1-\cos\delta)u_x u_z + \sin\delta u_y \\ (1-\cos\delta)u_x u_y + \sin\delta u_z & (1-\cos\delta)u_y^2 + \cos\delta & (1-\cos\delta)u_y u_z - \sin\delta u_x \\ (1-\cos\delta)u_x u_z - \sin\delta u_y & (1-\cos\delta)u_y u_z + \sin\delta u_x & (1-\cos\delta)u_z^2 + \cos\delta \end{bmatrix} \tag{A.21}$$

where u_x, u_y and u_z are the components of the unit vector that lies on z' axis. Hence,

$$\hat{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} -\cos\lambda \\ 0 \\ \sin\lambda \end{bmatrix} \tag{A.22}$$

When the steering bar is straight, direction of the front wheel can be shown with a unit vector that lies on x axis:

$$P_{straight} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{A.23}$$

When the steering bar is turned with an angle of δ as shown in Figure A.9, the vector $P_{straight}$ will be rotated such that:

$$P_{steered} = R_{z'}(\delta) P_{straight} \tag{A.24}$$

$P_{steered}$ is the new vector that is obtained by rotating $P_{straight}$. By using (A.22) and (A.21) in (A.24):

$$P_{steered} = \begin{bmatrix} (1-\cos\delta)\cos^2\lambda + \cos\delta \\ \sin\delta\sin\lambda \\ -(1-\cos\delta)\cos\lambda\sin\lambda \end{bmatrix} \tag{A.25}$$

Effective steering angle can be found by considering the top view of the bicycle (x-y plane) as shown in Figure A.10.

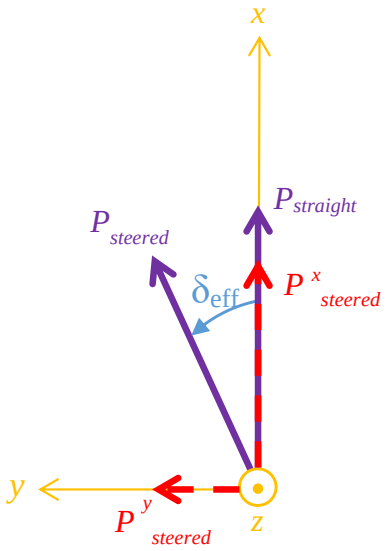


Figure A.10: Top view that describes the direction of the front wheel

$$\delta_{eff} = tan^{-1}\frac{P^{\,y}_{steered}}{P^{\,x}_{steered}} = tan^{-1}\frac{\sin\delta\sin\lambda}{(1-\cos\delta)\cos^2\lambda + \cos\delta} \tag{A.26}$$

By using the small angle approximation for δ; $\tan\delta \approx \delta$, $\sin\delta \approx \delta$ and $\cos\delta \approx 1$, (A.26) becomes:

$$\delta_{eff} = \delta\sin\lambda \tag{A.27}$$

Hence, although the steering bar is rotated by the steering angle δ, bicycle rotates according to the effective steering angle δ_{eff} which is less than δ for $\lambda < \pi/2$.

A.6 Complementary Filter

The complementary filter is a good solution to the problems of the angle estimation only by using gyroscope or accelerometer measurements. In complementary filter, a low-pass filter is used with accelerometer estimation and a high-pass filter is used with the integrated gyroscope measurements. This approach is illustrated in Figure A.11.

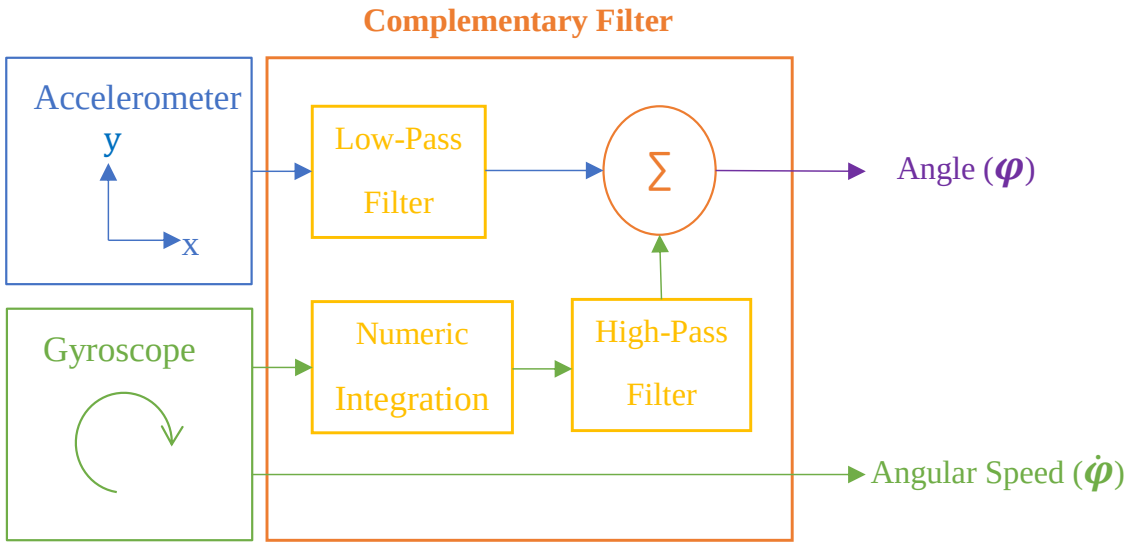


Figure A.11: Complementary filter

For fusing the information from gyroscopes and accelerometers together, following formulation is used:

$$\hat{\varphi} = C\ (\hat{\varphi}^{-} + \dot{\varphi}\ dt) + (1 - C)\ \hat{\varphi}_{acc} \tag{A.28}$$

where
C : filter coefficient ($0.5 < C < 1$)
$\hat{\varphi}$: roll angle estimation [rad]
$\hat{\varphi}^{-}$: previous roll angle estimation [rad]
$\dot{\varphi}$: gyroscope reading [rad/s]
dt : time interval between updates [s]
$\hat{\varphi}_{acc}$: roll angle estimation derived from accelerometer readings based on (2.10) in [rad]

For $0.5 < C < 1$, red term in (A.28) represents high-pass filter, and green term represents low-pass filter.

The filter coefficient C is chosen in order to find the best trade-off between gyroscope and accelerometer measurements. Large C increases the contribution of gyroscope measurements to roll angle estimation, and small C increases the contribution of accelerometer measurements. C is selected by doing a trial and error test:

1. C is initially set to 0.95 and estimation results are observed.

2. No drift (accumulation of error) is observed. But the angle estimation is very sensitive to any acceleration.
3. C is increased to 0.995.
4. Angle estimation starts to drift. In other words, gyroscope noise accumulates because of integration.
5. C is decreased to 0.985. With the new value of C, drift problem is solved. Furthermore, roll angle estimation is not too sensitive to any acceleration as it was when C was 0.95.

A.6.1 Validation of the Complementary Filter Used in This Project

In this test, the roll angle estimation obtained from the complementary filter is compared with the two other simple roll angle derivations based on accelerometers and gyroscopes. The test is performed in the following way:
IMU is detached from the bicycle and put onto a table (*Hence, blue and green terms in (2.10) are assumed to be zero, since the test is to illustrate the workability of the complementary filter in general, not necessarily on the bicycle.*). For the first a couple of seconds and the last a couple of seconds, the IMU is at a rest condition. In between these times, the IMU is shaken with arbitrary movements realized by hand. The results are shown in Figure A.12. At the end of the test (at 40^{th} second), there is a huge amount of drift in the integrated gyroscope values. And roll angle estimation derived from accelerometer values is too noisy since it is affected by any acceleration changes. Complementary filter roll angle estimation, on the other hand, is both noise and drift free.

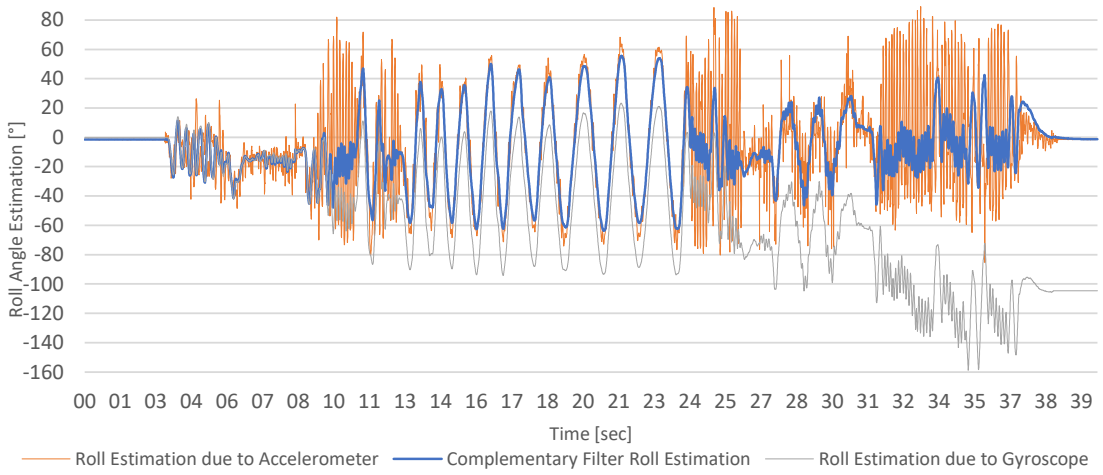


Figure A.12: Roll angle estimation due to complementary filter vs accelerometer vs gyroscope

A.7 Mounting the IMU

Mounting IMU with a good angular precision is important. x, y, z axes of the IMU should be parallel to the x, y, z axes shown in Figure A.7. The procedure for mounting the IMU is given below:

1. Locate the IMU to its mounting point.
2. Check a_z. It should be 1 g. a_x and a_y should be 0 g. Adjust the orientation accordingly. This ensures that, z axis of the IMU is directed to the correct direction.
3. Lean the bicycle to one side when the steering bar is straight. a_z decreases and a_y becomes non-zero. a_x should be still 0. If it becomes non-zero, then rotate the IMU around z axis to make a_x 0. This ensures that, x and y axes of the IMU are also directed to the correct directions.

A.8 Details of Output Equation (2.7)

Considering output equation (2.7), steering angle δ exists in the blue terms. Its value is already directly measured by using an encoder, hence the measured value of δ is used to estimate the roll angle φ. The green term, on the other hand, is obtained by replacing $\ddot{\varphi}$ by using (2.2). Hence, the model of the plant is used to obtain the green term. In the real life, model of the plant is not trustable to use. Since $\dot{\varphi}$ is already measured directly by using the gyroscope, it can be derivated and $\ddot{\varphi}$ can be obtained. However, since $\dot{\varphi}$ measurement is noisy, derivation would give inaccurate results. Thus, a low-pass filter should be applied before the derivation. The following low-pass filter is used to filter the high frequencies of the gyroscope readings:

$$\dot{\varphi}^{filtered,new} = C_{LP}\ \dot{\varphi}^{filtered,previous} + (1 - C_{LP})\ \dot{\varphi}^{measured} \tag{A.29}$$

where $0.5 < C_{LP} < 1$. For $C_{LP} = 0.8$, measured and filtered roll rates are shown in Figure A.13

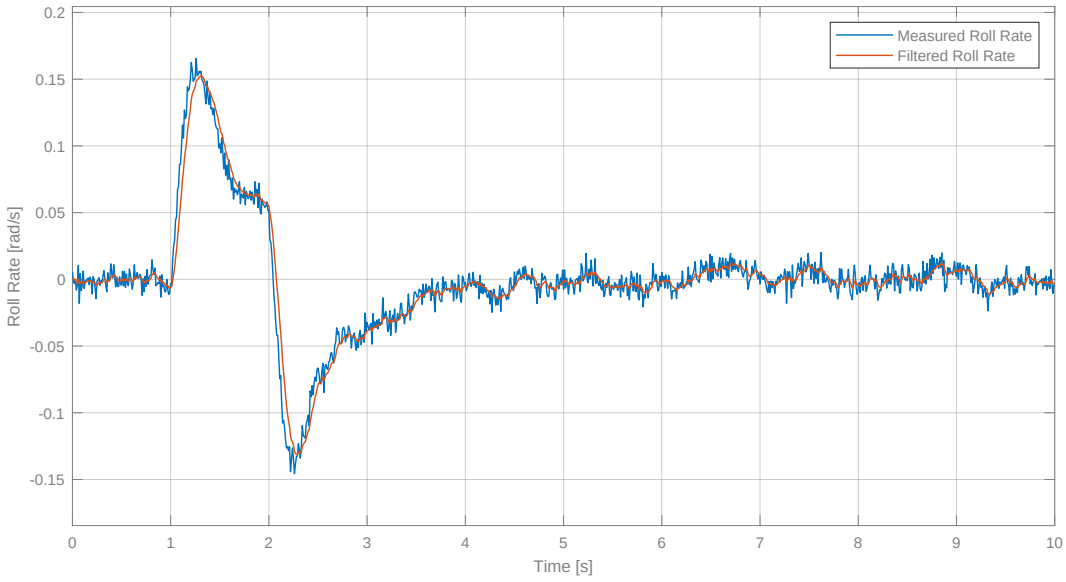


Figure A.13: Measured and filtered roll rates

A.9 Simulation Configurations

For the simulations, real noise data is collected for the gyroscope and accelerometers which are shown in Appendix A.10. Furthermore, for the steering encoder, a white Gaussian noise with a variance of 10^{-7} rad^2 is used.
For the steering motor, maximum and minimum steering speeds are limited with ± 7.5 rad, which is the maximum speed of the steering motor. A delay of 0.045 second is assumed. And a dead-band of 0.055 rad/s is used. That means, if the steering speed is below 0.055 rad/s, steering motor would not turn at all.
For all of the simulations, a sampling time of 0.04 second is used. All of the signals are simulated as discrete signals, except the plant which is continuous.

A.10 Analysis of the Noise of the Sensors

The IMU is put a flat surface and its noise values are recorded. The gyroscope measurements variance is obtained as 0.0022 °/s and a standard deviation is obtained as 0.047 °/s. The gyroscope noise for 1 second is shown in Figure A.14. The average gyroscope reading is measured as -0.003 °/s. It is non-zero, but very small. At the initialization of the bicycle IMU is always calibrated. This bias is always measured and it is subtracted from the gyroscope measurements

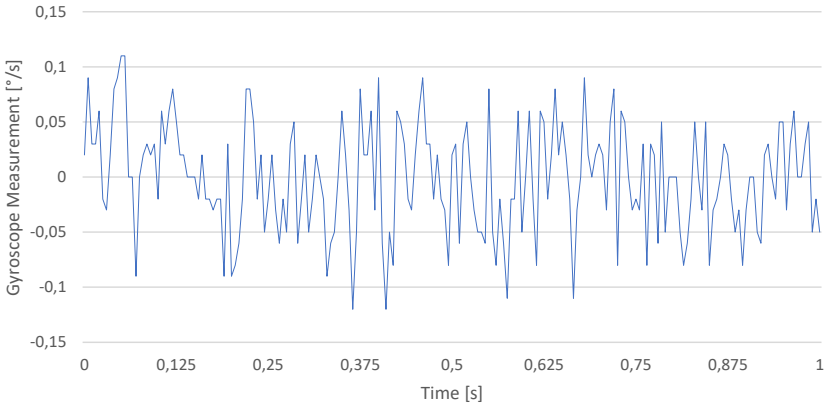


Figure A.14: Gyroscope readings for 1 second

Similarly, (lateral direction, y direction in Figure 2.3) accelerometer noise values are also measured and included to the simulation. Accelerometer noise for 1 second is shown in Figure (A.15).

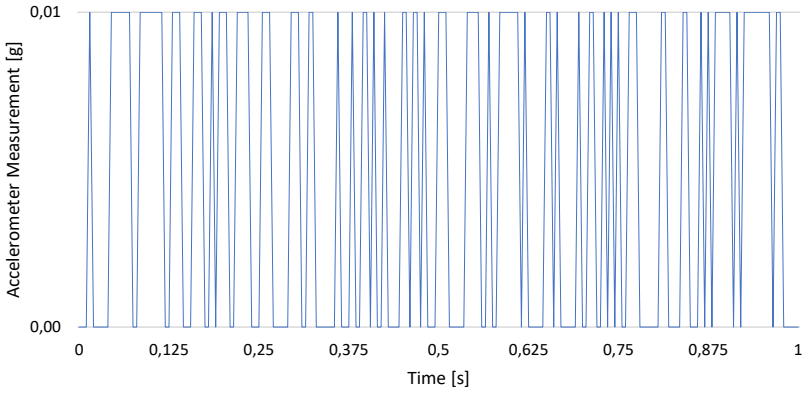


Figure A.15: Accelerometer readings for 1 second

Accelerometer noise is simpler, it is either 0 or 0.01 g. In fact, 0.01 g is same as the measurement sensitivity of the accelerometer. In the perpendicular direction (z direction in Figure 2.3), noise is measured as 0, meaning that noise in this direction is smaller than the measurement sensitivity of the accelerometer which is 0.01 g. Real noise values for both accelerometer and gyroscope for the real sensors are measured for 90 seconds and included to the simulations.

A.11 Coefficients of the Trajectory-Tracking PID Controllers

Proportional, integral and derivative coefficients of the PID controllers shown in Figure 4.5 are given below.

- Lateral position PID controller: P=0.1, I=0.01, D=0
- Direction PID controller: P=0.8, I=0.4, D=0
- Steering PID controller: P=1, I=0.1, D=0

All PID controllers are discrete PID controllers.