CHALMERS
UNIVERSITY OF TECHNOLOGY

Modelling and Path Tracking Control of an Autonomous Bicycle

Master's thesis in Systems, Control and Mechatronics

CÉCILE SAVOYE

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

Master's thesis 2019:EENX30

Modelling and Path tracking Control of an Autonomous Bicycle

CÉCILE SAVOYE

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

Modelling and Path Tracking Control of an Autonomous Bicycle
CÉCILE SAVOYE

Supervisor: Jonas Sjöberg, Department of Electrical Engineering
Examiner: Jonas Sjöberg, Department of Electrical Engineering

Master's Thesis 2019:EENX30
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 was used during this project.

Typeset in LaTeX
Gothenburg, Sweden 2019

Modelling and Path Tracking Control of an Autonomous Bicycle
CÉCILE SAVOYE
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Chalmers Mechatronics group is currently developing a self-driving bicycle. The purpose of this new type of vehicle is to help in testing autonomous cars, where it is meant to replace a human biker.

This Master's thesis project focuses on the design and performances of the balancing and path tracking control loops.

The description of the balancing control development methodology enables a methodical investigation plan to improve its performances. The bicycle's model is corrected by improving the definition of the mass distribution. It results in a successful balance of the bicycle during test drives.

The position estimation is computed by a Kalman filter combining the measurements of velocity and position. The implementation of this filter in simulation demonstrates the accuracy of the position estimation which leads to a robust control of trajectory.

Indicators and test cases are defined to measure the performance of the path tracking control in simulation. The purpose is to easily assess the severity of the test cases and to compare the performances predicted in the simulation with the ones which will be obtained during the test drives.

In conclusion, the contributions of this Master's thesis on the model's accuracy investigation and position feedback loop allowed to improve the balancing of the bicycle, and to implement a robust path tracking control. Performances of balancing control have been assessed in test drives. As for path tracking control, performances have been estimated in simulation and the test cases are ready for the real test drives.

Keywords: simulation, model, control, balancing, path tracking, performances.

Acknowledgements

I would like to thank the persons who contributed to the achievement of this Master Thesis and the redaction of this report.

First of all, I would like to thank my Master's Thesis supervisor, Prof. Jonas Sjöberg, Head of the Mechatronics Research Group at Chalmers University. He dedicated a lot of time for me as I was joining the project, provided me with continuous advice and support for the whole duration of my thesis. He also taught me how to structure and present my research work which I really appreciated. I am grateful I could participate in the development of such an interesting and complex project, and I thank him for it.

I really appreciated the help that I received from Maxime Feingesicht, Postdoctoral Researcher in the Mechatonics Research group. As he joined the project in May 2019, the dynamic of the project improved. His work in the debugging of the code was an important step forward and allowed us to perform the tests to validate the work that had been done so far. It was really rewarding to work with such an experienced person. I thank him for his many advice.

I particularly want to thank Yixiao Wang, Phd student in the Mechatonics Research group. I had the chance to work by his side for the whole duration of my Master's Thesis which gave me the opportunity to learn a lot from his work. As we were investigating different matters, communicating with each other helped me keep an overview of the bicycle project. I would like to thank him for his availability, patience and encouragements at any moment of the project.

Finally, I would like to underline my family's support while I was joining this research project abroad, and also the investment of my parents in the proofreading of this report.

Cécile Savoye, Gothenburg, June 2019

Contents

List of Figures

List of Symbols and Abbreviations

Angles Describing Movement and Position

φ	Roll angle
$\dot{\varphi}$	Roll rate
$\ddot{\varphi}$	Roll acceleration
δ	Steering angle
$\dot{\delta}$	Steering rate
δ_{eff}	Effective steering angle
β	Angle of the Centre of Mass velocity with respect to the longitudinal axis of the bicycle
ψ	Yaw angle

Geometrical Parameters

a	Horizontal distance from the centre of the rear wheel to the Centre of Mass
b	Distance between the rear and front wheel points of contact on ground
c	Fork trail
d	Horizontal distance from the centre of the rear wheel to the Inertial Measurement Unit
h	Height of the bicycle's Centre of Mass
h_{IMU}	Height of the Inertial Measurement Unit
λ	Fork angle

Others

g	Gravitational acceleration
CM	Centre of Mass
IMU	Inertial Measurement Unit
LQR	Linear Quadratic Regulator
GPS	Global Positioning System

1
Introduction

1.1 Technical Background

Studies about autonomous bicycles can be classified according to their various purposes and methodologies.

Purpose
The purposes of these studies are either to aid children to ride a bicycle [2], to discharge the rider of driving the bicycle (as in autonomous cars for example) [3] or to develop a solution for autonomous car test drives, as the one conducted by Chalmers University [1] and Mälardalen University [4].

In the context of autonomous vehicles, the design of safe systems is very complex. A great part of this complexity is due to the many variables composing the traffic, such as pedestrians, bicycles, cars, and their individual behaviour.

In order to ensure safe testing of this new kind of vehicle, the autonomous bicycle is meant to replace the human biker. To do so, the design of this bicycle has to show a similar behaviour to the one of an ordinary bicycle, to be able to balance itself and to track a given path.

Methodology
The work of N. Tamaldin and al. on the design of self-balancing bicycles [5] and the work of Kiattisin Kanjanawanishkul on controller design for autonomous bicycles [6] classify balancing methodologies for autonomous bicycles in four categories: Control Moment Gyroscope, Mass Balancing, Steering Control and Reaction Wheel. Universities of Cornell [3] and Tsinghua [7] chose the Steering Control methodology for their project and succeeded in balancing their bicycle and performing turns with it. Therefore, Chalmers' project is also based on this methodology and aims for a positive result as well.

Concerning the physical modelling of the bicycle, the work of J. Åström and al. [2] as well as the work of U. Erdinç [1] use a point of mass model. As explained in Section 1.3, this master thesis investigates a more realistic modelling of the bicycle, taking into account the distribution of mass.

1.2 Project Background

Chalmers Mechatronics group is currently developing an autonomous bicycle. This autonomous bicycle project has been running since autumn 2017 and is part of the collaboration between Chalmers University, Mälardalen University, Volvo Cars, AstaZero and Cycleurope. As Mälardalen University is also building a self-driving bicycle, this enables sharing problems and progress between both groups.

At Chalmers University, earlier contributions addressing the mechanical and electronic features were made by several students. The development and the implementation of the control algorithms on the autonomous bicycle are still in progress and need some improvements. Details of the previous work can be found in [1].

1.3 Contributions of this Master's Thesis

The bicycle's geometrical parameters, the actuators and sensors enabling its automation, the simulation and test environments are introduced in Chapter 2.

The first contribution of this Master's Thesis is the improvement of the balancing of the bicycle, by setting up an investigation plan based on the balancing control development methodology. The model of the bicycle is redefined so that it takes the distribution of mass into account. This more realistic model, together with the appropriate tuning of the Linear Quadratic Regulator and the correction of the code, leads to a better balancing of the bicycle during test drives. This contribution is presented in Chapter 3.

The second contribution is the design of a realistic path tracking control. Position measurements are included and combined to velocity measurements in a Kalman filter to obtain an accurate position estimation in the control feedback loop. The impact of the model's accuracy is also assessed for path tracking control. This contribution is detailed in Chapter 4.

The third contribution is the development of a frame to assess the path tracking performances. It includes the computation of indicators comparing the bicycle's path to the reference. On the other hand, it defines test cases based on which predicted behaviour in simulation and test drive results can be compared. This contribution is described in Chapter 5

2

Overview of the Autonomous Bicycle

This Chapter gives an overview of the autonomous bicycle: the bicycle's model, the actuators and sensors integrated in the bicycle, the simulation environment describing the control strategy and the test environment in which the test drives are performed.

2.1 Bicycle

2.1.1 Geometrical Parameters

The bicycle is described by its geometrical parameters in order to create a representative model in simulation. In Figure 2.1, the roll angle φ is defined. It is monitored to ensure the balancing of the bicycle.

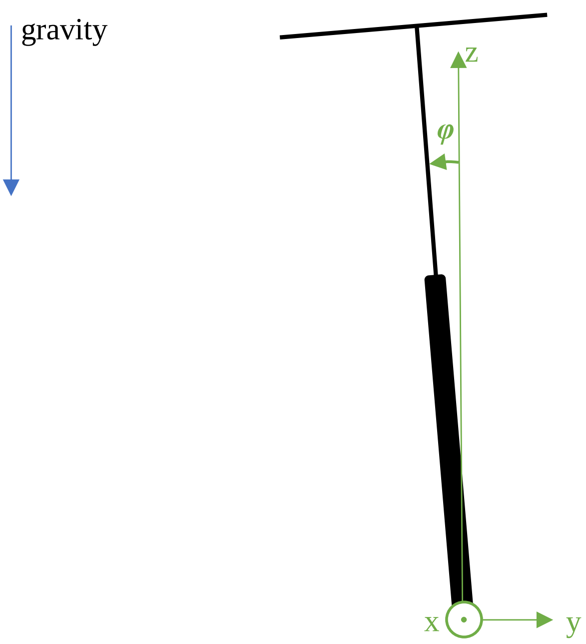

Figure 2.1: Front view of the bicycle – definition of the roll angle. [1]

In Figure 2.2, the main geometrical parameters of the bicycle are represented:

- h_{IMU} : height of the Inertial Measurement Unit (IMU)
- h : height of the bicycle's Centre of Mass (CM)
- d : horizontal distance from the centre of the rear wheel to the Inertial Measurement Unit
- a : horizontal distance from the centre of the rear wheel to the Centre of Mass
- b : distance between the rear and front wheel points of contact on ground
- c : fork trail
- λ : fork angle

These parameters are used in the equations of the model, which are presented in Section 3.1.

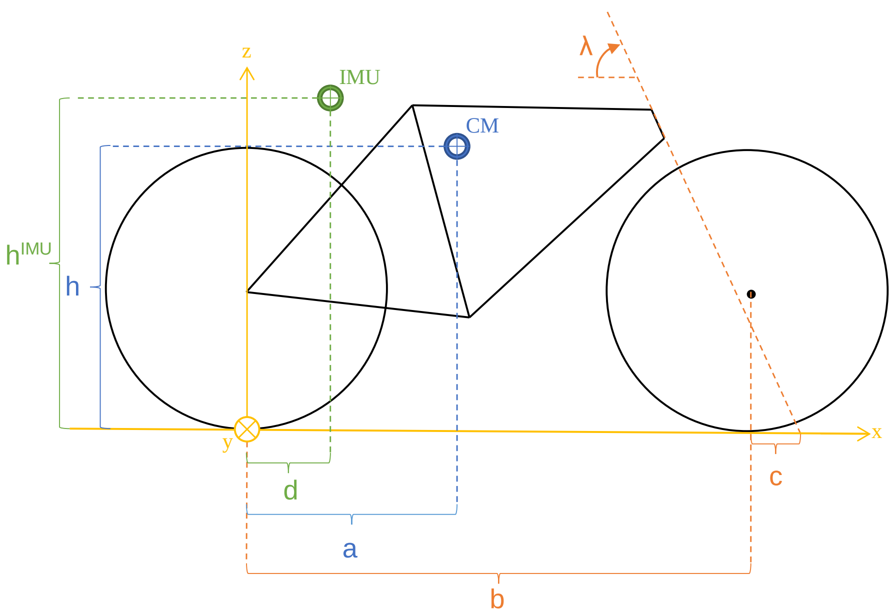

Figure 2.2: Side view of the bicycle - Definition of the geometrical parameters. [1]

In Figure 2.3, the bicycle is shown in the global coordinates system which is used to compute the trajectory and perform the path tracking control. Several angles are also defined:

- δ : steering angle of the handlebar
- β : angle of the Centre of Mass velocity with respect to the longitudinal axis of the bicycle
- ψ : yaw angle

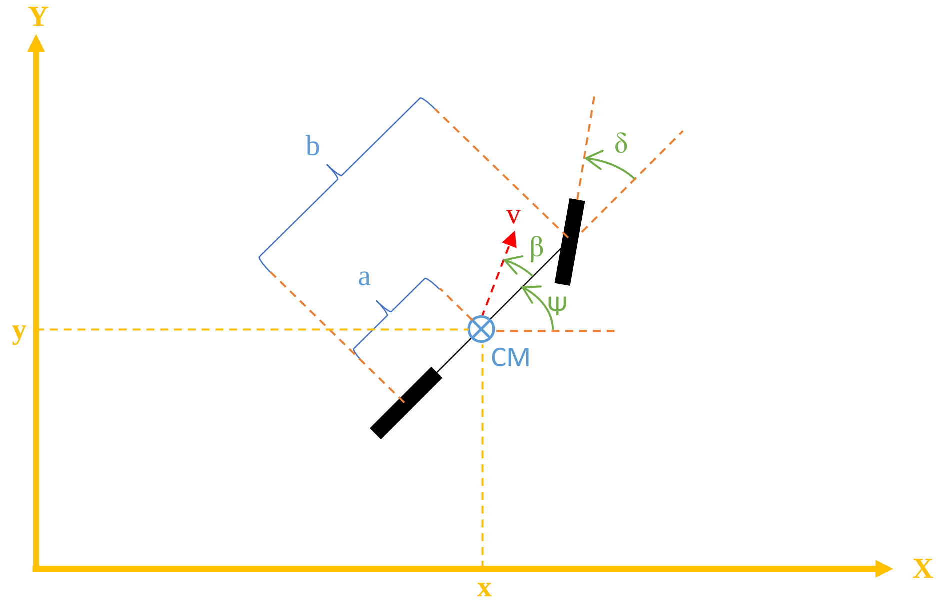

Figure 2.3: Top view of the bicycle - Definition of the global coordinates system and steering angle. [1]

2.1.2 Actuators and Sensors

Different actuators and sensors are implemented on the bicycle to control it and to observe its response. Figure 2.4 shows all actuators and sensors as well as their position on the bicycle. Figure 2.5 defines the inputs and outputs of the bicycle given by the actuators and sensors. The links between both Figures are the following:

- The forward velocity actuator is the built-in motor of the electric bicycle. It applies the reference velocity v^{ref} to the bicycle.
- The steering motor is responsible for the steering rate $\dot{\delta}^{ref}$ applied to the handlebar.
- The encoder measures the steering angle δ.
- The IMU (Inertial Measurement Unit) is a combination of an accelerometer and a gyroscope so it records the linear accelerations and angular rates in the three directions. These measurements are used to obtain the roll rate $\dot{\varphi}$ and the roll angle φ estimations.
- The Hall effect sensor is combined with neodymium magnets to estimate the bicycle's velocity.

More detailed information about the actuators and sensors implemented on the bicycle can be found in [1] as well as their validation.

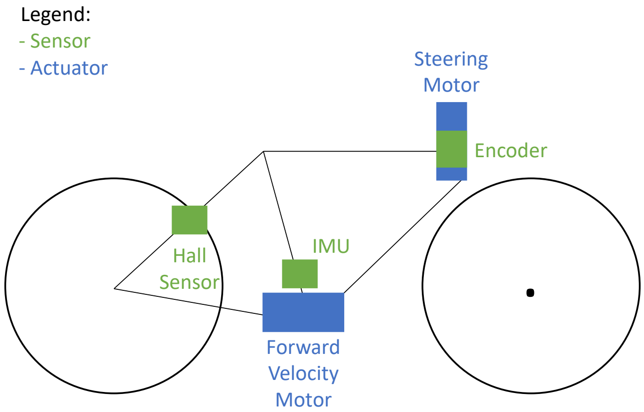

Figure 2.4: Actuators and Sensors implemented on the bicycle.

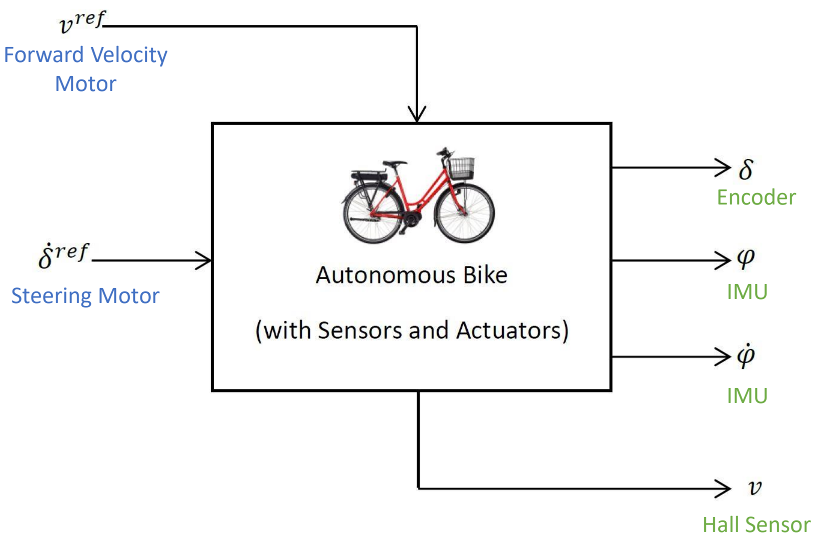

Figure 2.5: Bicycle's system combining actuators and sensors - Inputs from actuators, outputs from sensors.

2.2 Simulation Environment

In order to tune the bicycle controllers and assess their performances, a simulation environment describing the autonomous bicycle system is implemented in Simulink. Its structure is detailed in Figure 2.6.

Two reference blocks (in blue) are used as the inputs of the system, for velocity and position. Velocity reference is independent of the control loops while position reference is updated during path tracking simulations.

Two controller blocks (in purple) are defined in the simulation environment, the bicycle's control being divided into: balancing control and path tracking control. Balancing control uses a Linear Quadratic Regulator (LQR) while trajectory control is performed by a cascade of PID controllers. These two control loops are reviewed and improved in Chapters 3 and 4 respectively. Concerning the implementation of the controllers, the details can be found in [1].

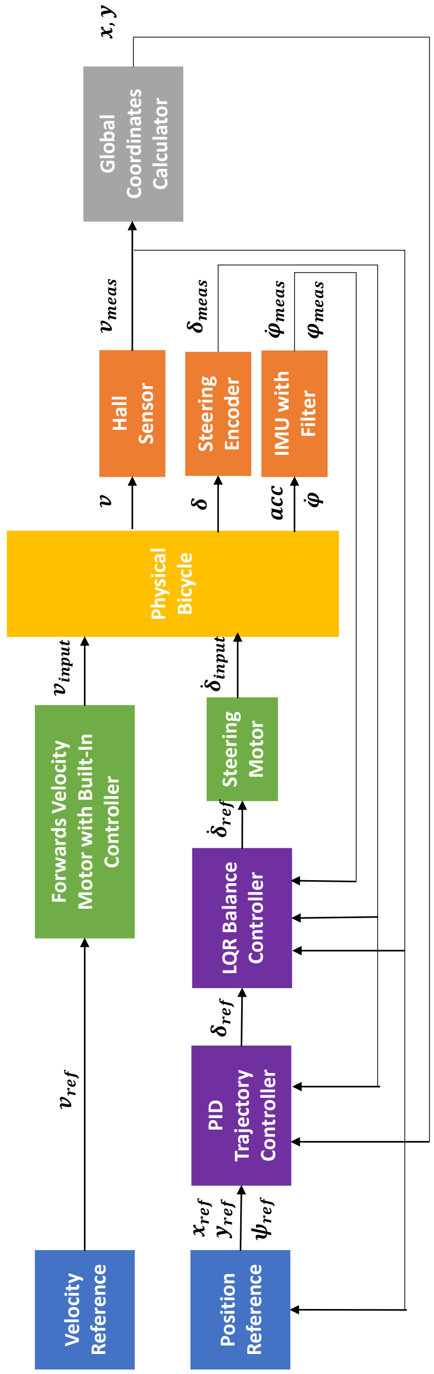

Figure 2.6: Simulation environment of the bicycle - Definition of the control loops.

Actuators and sensors described in Section 2.1.2 are modelled in the simulation environment. Actuator blocks (in green) include the physical limitations of the motors (minimal and maximal velocity, dead band). Sensor readings (in orange) are modelled by adding noise to the true values calculated in the physical bicycle model.

The physical bicycle model (in yellow) represents the dynamics of the autonomous bicycle using the geometrical parameters defined in Section 2.1.1. It is used to determine the behaviour of the bicycle by calculating the states estimates and the actual velocity. In reality, these values cannot be accessed. Only the outputs of the sensors are available.

The global coordinates calculator block (in grey) transposes the measured velocity and steering angle into the global coordinates system described in Figure 2.3. The resulting parameters can then be used in the control loop for trajectory control.

Geometrical parameters, actuators limitations and settings of the simulations are defined in a MATLAB file. It also enables the display of the results in different plots and in a 3D animation (see Appendix A.1).

2.3 Test Environment

Tests performed on the bicycle were done both indoors and outdoors. A cycling roller was used so that balancing control could be tested indoors. Nevertheless, test drives were still done outdoors as the environments present substantial differences, such as resistance on the wheels and space limitation.

The surfacing of the rolls is smoother compared to the ground on which the outdoors tests are conducted. Therefore, the resistance acting on the wheels during the test drives are different in the two cases. The initialisation of the test drives is done by hand to help the forwards velocity motor to counteract ground resistance.

Indoors test environment is limited on two points compared to outdoors drive tests: the steering angle amplitude and the lateral deviation. Both of these limitations are due to the use of a roller and its geometry. There is no similar problem outdoors.

Comparing to simulation, there is a difference with the real-life test drives linked to the ground resistance which is not taken into account for now.

3

Adjustment of the Balancing of the Bicycle: Improvement of the Model

Instabilities of the balancing control are observed during test drives: the steering angle presents increasing oscillations and the bicycle falls after a few seconds. Therefore, the balancing control has to be improved.

In order to investigate the instabilities' origins and to correct the balancing control efficiently, a methodical plan is established based on the development methodology of the balancing control (see Figure 3.1). First, a model is developed to represent the autonomous bicycle and to describe its dynamics. Then, based on this model, the controller is designed. Finally, the controller algorithm is implemented on the real bicycle and tests can be performed.

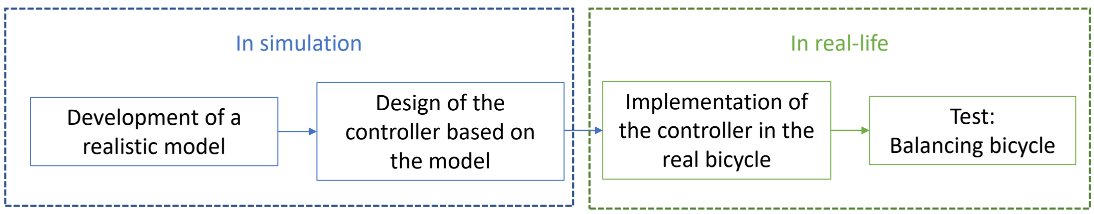

Figure 3.1: Balancing control development methodology.

From this methodology, the balancing instabilities encountered during the test drives could be explained by three main causes (see Figure 3.2):

- The model developed in simulation to represent the bicycle is not realistic enough for the controller to balance the real bicycle.
- The controller design is not robust enough to stabilise the real plant.
- The implementation on the bicycle contains errors.

This contribution details the investigation on the first cause: the physical model's accuracy. The second and third causes were handled by the other team members and are not described here. However, for the sake of completeness, it should be noted that the final results described in Section 3.2 incorporates the effects of the three corrections.

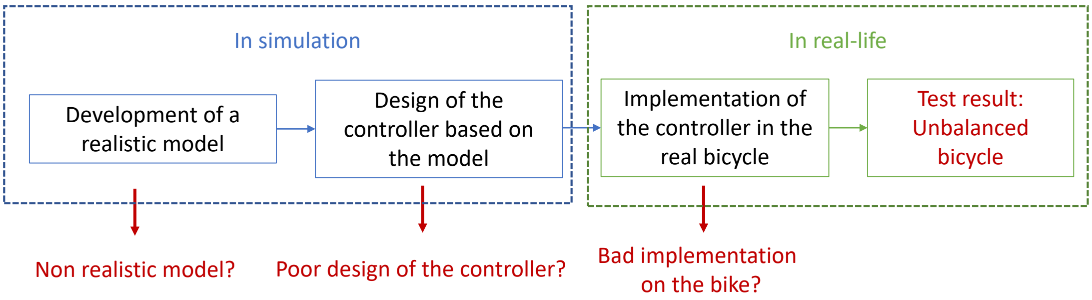

Figure 3.2: Balancing control development methodology - Possible causes of instability.

3.1 Improvement of the Physical Model

The physical model used to describe the autonomous bicycle in simulation defines the system as a point of mass. In reality, the bicycle is made of various components (frame, wheels, handlebar, ...). As a consequence, the moment of inertia of the bicycle depends on all of these components' mass and centre of mass position. A realistic computation of the resulting moment of inertia is important as it constitutes the first step in the definition of the system's state space model, which itself also influences the controller design (see Figure 3.3).

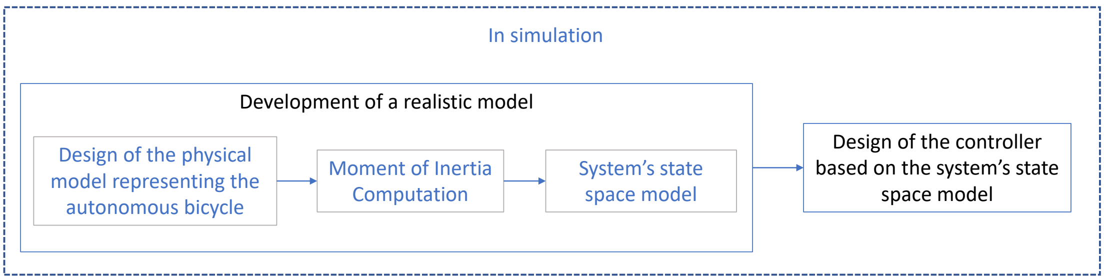

Figure 3.3: Balancing control – Development of a realistic model.

In order to keep a simple physical model and therefore a simple state space representation, the autonomous bicycle can be divided into two bodies: the bicycle and the electronics box. This box contains the electronic boards, motor controllers, battery and safety circuit that are needed for the automation of the bicycle. It is attached to the bicycle's frame above the rear wheel and weights approximately 12 kg.

The box representing approximately a quarter of the weight of the bicycle and its position being offset compared to the bicycle centre of mass, the concern is that it would make the "bicycle-box" system naturally less stable than the bicycle alone. Therefore, it could represent an additional constraint to take into account in the design of the balancing control. Proving this correlation is precisely the objective of this Section.

First, the influence of the physical model is illustrated by comparing different representations of the bicycle's model and their effect on the moment of inertia value:

- one point of mass model
- one volume model
- two volumes model

Then, the state space representation is computed for the two bodies model. Finally, the simulation results for the point of mass model are compared to the two bodies ones. The most accurate model, described by the two volumes, appears to present bigger variation of the roll and steering angles for the same controller settings.

3.1.1 Comparison of the Models and their Influence on the Computation of the Moment of Inertia

The objective of this Section is to improve the current model to make it more representative of the reality and to be able to highlight the influence of the mass distribution on the moment of inertia.

By describing the autonomous bicycle as a unique point of mass, the following approximations are made:

- The system is closer to a body composed by numerous points of mass, rather than a unique point of mass.
- The system involves numerous components. The bicycle and the electronics box, by their mass and position, are the components that potentially influence the moment of inertia the most, leading to the consideration of a two bodies model.

3.1.1.1 Moment of Inertia Evaluation: from a Point of Mass Model to a Volume Model

The first approximation which is studied is the volume model compared to the point of mass.

The point of mass system is defined in Figure 3.4. It is shifted upwards by a length h from the axis origin, representing the height of the bicycle's centre of mass measured from the ground.

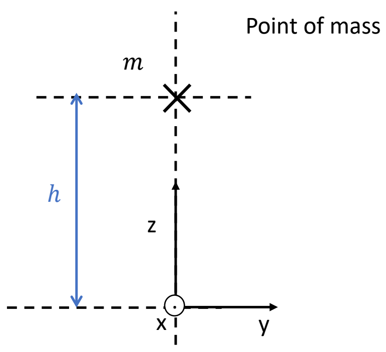

Figure 3.4: Definition of the system as a point of mass.

The volume model in the other hand is defined by its geometrical parameters: b is the height along the axis z, c is the width along the axis y and h is the height of its Centre of Mass (see Figure 3.5).

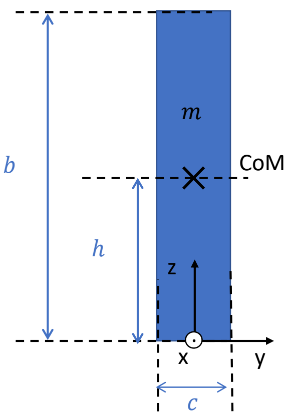

Figure 3.5: Definition of the system as a volume of mass.

To compute the moment of inertia around the axis x, the Parallel Axis Theorem is used. It states that the moment of inertia around an axis is equal to the inertia at the centre of mass added to the inertia contribution due to the distance between the centre of mass and the considered axis (3.1) [8]:

$$J_{xx} = J_{CoM} + m\,h^2 \tag{3.1}$$

Where J_{xx} is the moment of inertia around the axis x, J_{CoM} is the moment of inertia around the centre of mass, m is the mass of the body and h is the distance between the body's centre of mass and the axis x.

For a point of mass, the inertia around the centre of mass is null as all the mass is already located in that point. Thus, the equation of the inertia around the x axis can be re-written (3.2).

$$J_{xx} = m\,h^2 \tag{3.2}$$

By defining the system as a volume, the inertia at the centre of mass must be considered, and the moment of inertia takes the volume's geometry into account:

$$J_{xx\,1\,volume} = J_{CoM} + m\,h^2 = \frac{m}{12}\,(c^2 + b^2) + m\,h^2 \tag{3.3}$$

The comparison of the moments of inertia of both models (3.2) and (3.3) reveals that the point of mass approximation does not take into account a term linked to the geometry of the considered volume. We can already consider that the volume of mass model is more realistic.

3.1.1.2 Moment of Inertia Evaluation: from a One Volume Model to a Two Volumes Model

So far, the influences of two main components (the bicycle and the electronics box) have not been taken into account separately. As this model describes the geometry of the bicycle more accurately, we can expect that a two volumes model would bring even more realistic results. The purpose of this Section is to evaluate its impact on the moment of inertia computation.

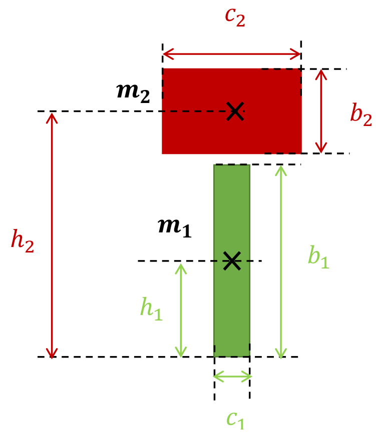

Figure 3.6: Definition of the system as two volumes.

The system described with two volumes is shown in Figure 3.6. As in the previous section, the inertia calculus is based on the expression of the Parallel Axis Theorem (3.1) as there is an offset between the centres of mass and the axis around which the moment of inertia is calculated. The equation of the inertia around x for this model becomes:

$$J_{xx\,2\,volumes} = J_{CoM1} + m_1\,{h_1}^2 + J_{CoM2} + m_2\,{h_2}^2 \tag{3.4}$$

Where J_{CoMi} is the moment of inertia around the centre of mass of volume i, h_i is the distance between the volume i's centre of mass and the axis x, m_i is the mass of the volume i.

The moment at the centre of mass for a volume is defined by its geometrical parameters as in the previous Section. Thus, (3.4) becomes:

$$J_{xx\,2\,volumes} = \frac{m_1}{12}\,({c_1}^2 + {b_1}^2) + m_1{h_1}^2 + \frac{m_2}{12}\,({c_2}^2 + {b_2}^2) + m_2\,{h_2}^2 \tag{3.5}$$

To verify the accuracy of the moment of inertia equation for a two volumes model, the equivalence between single volume and two volumes models is checked in case of a homogeneous mass distribution in Appendix A.2.

The expression of the moment of inertia becomes more complex. It now depends on the geometry of two bodies. The description of the bicycle could get even more accurate by considering more volumes but a compromise has to be reached between model's accuracy and computation's complexity.

3.1.1.3 Comparison of All Models based on a Numerical Example

To get an appreciation of the two volumes model change compared to the point of mass and single volume models, a numerical example is simulated. To do so, the mass, geometry and centre of mass positions of the bicycle and of the box are estimated:

- The bicycle and box are weighted to obtain their mass.
- The position of the box's centre of mass is estimated as the centre point of its volume.
- The position of the bicycle's centre of mass is computed following the method described in [9].

The different values are defined in Figure 3.7.

For the point of mass and one volume models, the parameters are computed as follows:

$$m = m_1 + m_2; \quad b = \frac{b_1\, m_1 + b_2\, m_2}{m_1 + m_2}; \quad c = \frac{c_1\, m_1 + c_2\, m_2}{m_1 + m_2}; \quad h = \frac{h_1\, m_1 + h_2\, m_2}{m_1 + m_2}$$

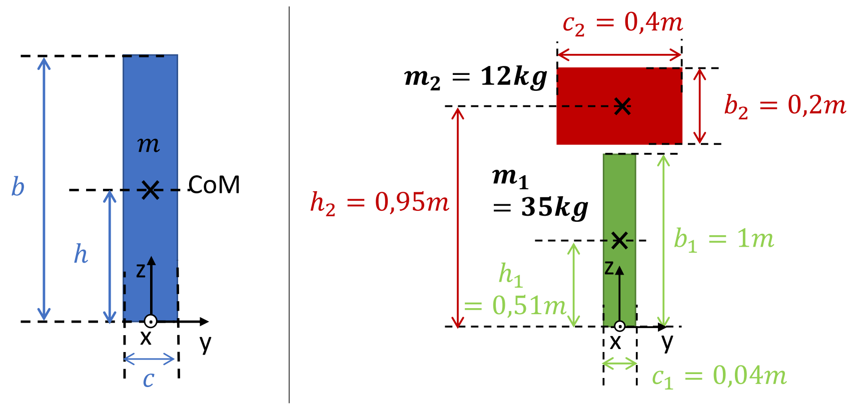

Figure 3.7: Comparison of the models - Numerical example.

The moments of inertia are computed for each model:

$$J_{xx\,point\,of\,mass} = m\, h^2 = 17,46\,\mathrm{kg.m}^2 \tag{3.6}$$

$$J_{xx\,1vol} = \frac{m}{12}\,(c^2 + b^2) + m\, h^2 = 20,01\,\mathrm{kg.m}^2 \tag{3.7}$$

$$J_{xx\,2vol} = \frac{m_1}{12}\,({c_1}^2 + {b_1}^2) + m_1\, {h_1}^2 + \frac{m_2}{12}\,({c_2}^2 + {b_2}^2) + m_2\, {h_2}^2 = 21,94\,\mathrm{kg.m}^2 \tag{3.8}$$

The difference between the values of the moments of inertia for each model is significant, specifically comparing the point of mass model and the two volumes model. Relatively, the inertia of the two bodies model is approximately 15% higher than the point of mass model.

As a conclusion, the two volumes physical model describes reality better and its influence on the moment of inertia value has just been demonstrated.

3.1.2 Two Volumes Model State Space Representation

As demonstrated in the previous Section, the definition of the physical model influences the value of the moment of inertia. Therefore, it also influences the state space representation. By defining the state space representation for the two volumes model, simulations can be run with a model closer to reality.

This section details the state space representation development for the two volumes model, extrapolating from the one used for the point of mass model in [1].

The starting point is the **angular momentum** of the system, using the moments of inertia:

$$L_x = J_{xx}\, w_x + J_{xy}\, w_y + J_{xz}\, w_z \tag{3.9}$$

With L_x is the angular momentum around the axis x, J_{xi} is the moments of inertia around the axis x and w_i is the angular velocities.

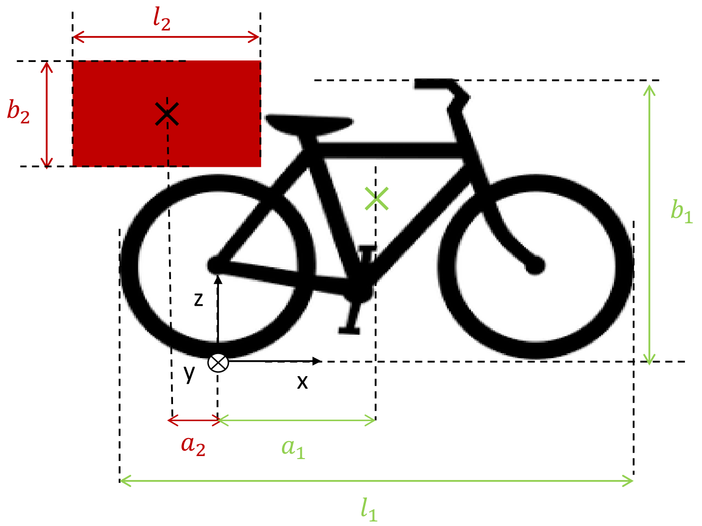

Figure 3.8: Side view of the bicycle and definition of the two volumes geometry.

The angular velocities are defined by (see definition of the angles in Chapter 2):

$$w_x = \dot{\varphi} \tag{3.10}$$

$$w_y = 0 \tag{3.11}$$

$$w_z = \dot{\psi} = \frac{v \tan \delta_{eff}}{b} \tag{3.12}$$

From Figure 3.8, the Parallel Axis Theorem is used to express the moment of inertia (3.13) and product of inertia (3.14):

$$J_{xx} = \frac{m_1}{12}\,({c_1}^2 + {b_1}^2) + m_1\, {h_1}^2 + \frac{m_2}{12}\,({c_2}^2 + {b_2}^2) + m_2\, {h_2}^2 \tag{3.13}$$

$$\begin{aligned} J_{xz} &= J_{xz\,CoM1} - m_1\,a_1\,h_1 + J_{xz\,CoM2} + m_2\,a_2\,h_2 \\ &= -m_1\,a_1\,h_1 + m_2\,a_2\,h_2 \end{aligned} \tag{3.14}$$

Values of $J_{xz\,CoMi}$ can be neglected as the product of inertia for cuboids are approximately null.

To ease the computations and the reading, the following simplifications are made, considering the steering angle δ value stays relatively small:

$$\delta_{eff} = \delta \sin \lambda \tag{3.15}$$

$$\tan \delta \approx \delta \tag{3.16}$$

Where δ_{eff} is the effective steering angle. This angle must be defined as the front wheel does not touch the ground vertically under the steering bar. Therefore, the fork angle induces a difference between the steering angle on the handlebar and the effective steering angle on the wheel.

Finally, the expression of (3.9) becomes:

$$\begin{aligned} L_x = \Big[\frac{m_1}{12}\,({c_1}^2 + {b_1}^2) + m_1{h_1}^2 + \frac{m_2}{12}\,({c_2}^2 + {b_2}^2) + m_2{h_2}^2\Big]\,\dot{\varphi} \\ - \Big[m_1 a_1 h_1 - m_2 a_2 h_2\Big]\,\frac{v\,\delta \sin\lambda}{b} \end{aligned} \tag{3.17}$$

The definition of the **derivative of the system's angular momentum** described in [1] is the second step:

$$\begin{aligned} \frac{dL_x}{dt} = (m_1 h_1 + m_2 h_2)\,g\,\sin\varphi + (m_1 h_1 + m_2 h_2)\,\frac{v^2 \tan\delta_{eff}}{b} \\ - (m_1 a_1 + m_2 a_2)\,\frac{c\,g\,\delta_{eff}}{b} \end{aligned} \tag{3.18}$$

With the different terms:

- Torque generated by gravity
- Torque due to centrifugal force
- Torque due to the geometry of the front fork. For $c \neq 0$, the centre of mass is shifted when the fork is turned

To ease the computation and the reading, simplifications from (3.15), (3.16) as well as (3.19) are applied.

$$\sin\varphi = \varphi \tag{3.19}$$

By replacing these expressions in (3.18), the derivative of the system's angular momentum becomes:

$$\begin{aligned} \frac{dL_x}{dt} &= (m_1 h_1 + m_2 h_2)\,g\,\varphi + (m_1 h_1 + m_2 h_2)\,\frac{v^2 \delta \sin\lambda}{b} - (m_1 a_1 + m_2 a_2)\,\frac{c\,g\,\delta \sin\lambda}{b} \\ &= (m_1 h_1 + m_2 h_2)\,g\,\varphi + \frac{\delta \sin\lambda}{b}\,\Big[v^2\,(m_1 h_1 + m_2 h_2) - c\,g(m_1 a_1 + m_2 a_2)\Big] \end{aligned} \tag{3.20}$$

The third step is the **derivation of (3.17)** which leads to the following expression:

$$\frac{dL_x}{dt} = \left[\frac{m_1}{12}\left({c_1}^2 + {b_1}^2\right) + m_1\,{h_1}^2 + \frac{m_2}{12}\left({c_2}^2 + {b_2}^2\right) + m_2\,{h_2}^2\right]\ddot{\varphi} \\ - \left[\,m_1\,a_1\,h_1 - m_2\,a_2\,h_2\right]\frac{\sin\lambda}{b}\left(\dot{v}\,\delta + v\,\dot{\delta}\right) \quad (3.21)$$

For readability reasons further in the document, the orange and purple expressions are defined as follows:

$$A = \left[\frac{m_1}{12}\left({c_1}^2 + {b_1}^2\right) + m_1\,{h_1}^2 + \frac{m_2}{12}\left({c_2}^2 + {b_2}^2\right) + m_2\,{h_2}^2\right] \quad (3.22)$$

$$B = \left[m_1\,a_1\,h_1 - m_2\,a_2\,h_2\right] \quad (3.23)$$

By definition, the expressions of the **(3.20) and (3.21) are equal**:

$$A\,\ddot{\varphi} - B\,\frac{\sin\lambda}{b}\left(\dot{v}\,\delta + v\,\dot{\delta}\right) = (m_1h_1 + m_2h_2)\,g\,\varphi + \frac{\delta\sin\lambda}{b}\left[v^2\,(m_1h_1 + m_2h_2) \\ - c\,g(m_1a_1 + m_2a_2)\right] \quad (3.24)$$

$$A\,\ddot{\varphi} = (m_1h_1 + m_2h_2)\,g\,\varphi + \frac{\delta\sin\lambda}{b}\left[B\,\dot{v} + v^2\,(m_1h_1 + m_2h_2) - c\,g(m_1a_1 + m_2a_2)\right] \\ + B\,\frac{\sin\lambda}{b}\,v\,\dot{\delta} \quad (3.25)$$

For the same readability reasons, the light blue expression is defined as:

$$D = \left[B\,\dot{v} + v^2\,(m_1h_1 + m_2h_2) - c\,g(m_1a_1 + m_2a_2)\right] \quad (3.26)$$

And considering $\dot{v} \approx 0$, the **state space form** is given by:

$$\begin{bmatrix}\dot{\varphi}\\ \dot{\delta}\\ \ddot{\varphi}\end{bmatrix} = \begin{bmatrix}0 & 0 & 1\\ 0 & 0 & 0\\ \frac{(m_1h_1+m_2h_2)\mathbf{g}}{A} & \frac{\sin\lambda\,D}{\mathbf{b}\,A} & 0\end{bmatrix}\begin{bmatrix}\varphi\\ \delta\\ \dot{\varphi}\end{bmatrix} + \begin{bmatrix}0\\ 1\\ \frac{B\sin\lambda\,\mathbf{v}}{\mathbf{b}\,A}\end{bmatrix}\dot{\delta} \quad (3.27)$$

For comparison, the state space representation for a point mass model is given by:

$$\begin{bmatrix}\dot{\varphi}\\ \dot{\delta}\\ \ddot{\varphi}\end{bmatrix} = \begin{bmatrix}0 & 0 & 1\\ 0 & 0 & 0\\ \frac{\mathbf{g}}{h} & \frac{\sin\lambda\,(h\,v^2 - g\,a\,c)}{\mathbf{b}\,h^2} & 0\end{bmatrix}\begin{bmatrix}\varphi\\ \delta\\ \dot{\varphi}\end{bmatrix} + \begin{bmatrix}0\\ 1\\ \frac{a\sin\lambda\,\mathbf{v}}{\mathbf{b}\,h}\end{bmatrix}\dot{\delta} \quad (3.28)$$

The 2 volumes model state space representation takes the mass distribution into account (both masses, centres of mass position, geometries of the bodies), which was not the case in the previous form.

3.1.3 Simulation Results and Comparison of the Models

The two volumes model is implemented in Simulink in order to be tested and compared to the point of mass model.

The balancing simulation results of both models are shown in Figures 3.9 (Point of mass model) and 3.10 (Two volumes model), in the same conditions:

- Velocity of 3 m/s
- Ideal initial conditions: $\varphi_0 = 2°$, $\delta_0 = 1°$, $\dot{\varphi}_0 = 0°/\mathrm{s}$
- Adjusted parameters of the balancing controller from controller settings investigation (see Figure 3.2): $Q = \mathrm{diag}([100\,100\,10])$ and $R = 42$

Observations
Both Figures predict that the bicycle should remain stable for the whole duration of the simulation. For the two volumes model, the oscillations of the roll and steering angles present higher amplitude and lower frequency than for the point of mass model.

Conclusion
The amplitude and frequency differences observed between the roll and steering angles of both models can be explained by the fact that the two volumes model considers a higher momentum of inertia of the bicycle. This behaviour leads to longer periods of time where the bicycle reaches the maximal roll rate values and a bigger action on the steering angle in order to counteract this phenomenon.

The difference of behaviour of both models shows that the two volumes model is preferred to describe the dynamics of the bicycle for balancing control as the point of mass gives too optimistic results.

The LQR controller parameters being designed for a point of mass model and the two volumes model representing the real plant more accurately, this means that bigger oscillations will be observed when using these settings on the real bicycle. Therefore, the model's inaccuracy partially explains balancing instabilities of the test drives.

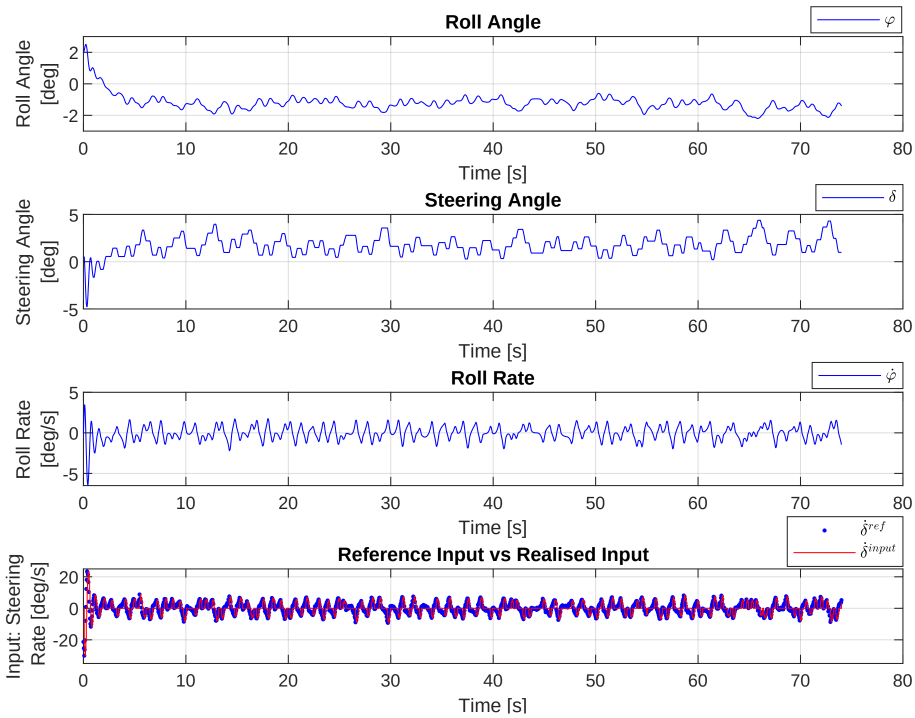

Figure 3.9: Balancing simulation of the point of mass model - Smaller amplitude and higher frequency variations of the roll angle than for the two volumes model.

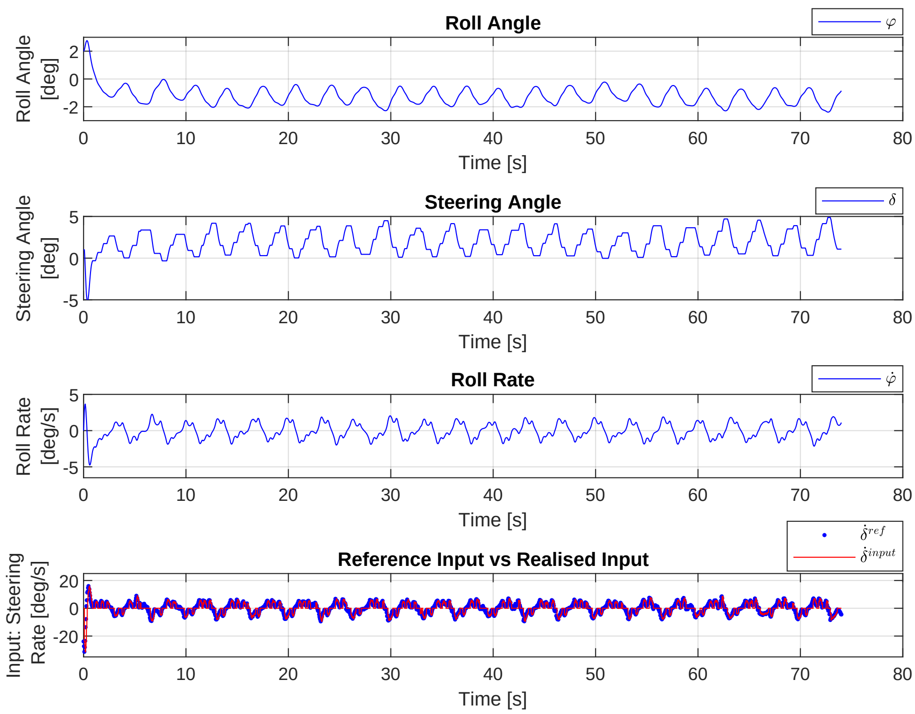

Figure 3.10: Balancing simulation of the two volumes model - Higher amplitude and lower frequency variations of the roll angle than for the point of mass model showing the effect of the higher moment of inertia taken into account.

3.2 Balancing Control: Test Drive Results

After implementing the correction of the model, the tuning of the controller setting and the improvement of the implementation on the autonomous bicycle (following the investigation plan of Figure 3.2), test drives are performed to assess the balancing capabilities of the bicycle. This section shows one of the results leading to a successful balancing of the bicycle.

The conditions in which the test was performed are the following:

- Outdoors test environment
- Velocity $= 3,9\,\mathrm{m/s}$
- Initialisation by hand during the 5 first seconds. Then, the controller takes over.
- Adjusted LQR settings: $Q = \mathrm{diag}([100, 100, 10])$, $R = 42$

Observations

Figure 3.11 shows the steering and roll angles obtained during the test drive. Before the second 5, the bicycle is balanced by hand while it is gaining speed. The steering and roll angles are respectively 3° and -1° when then controller is activated.

Afterwards, the oscillations' amplitude diminishes for both angles, illustrating the balancing control action. When the roll angle gets more significant around the 16s mark, the steering counteracts directly and the oscillations decrease once again. The test drive was stopped manually as the test environment is limited in space.

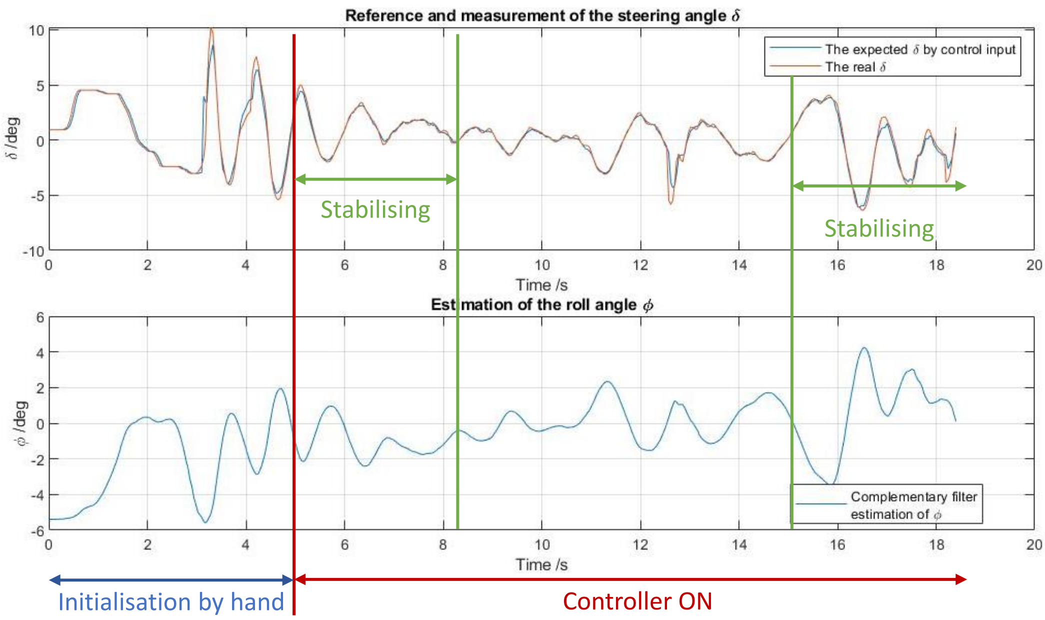

Figure 3.11: Balancing test drive results at $3,9\,\mathrm{m/s}$ - As the balancing control is activated, the steering angle action stabilises the bicycle within 3 seconds as highlighted in green for two examples.

Conclusion

The measured steering and roll angles present higher variations than the simulations but the bicycle remained balanced for the whole test duration. In conclusion, the adjustments applied on the balancing control following the investigation plan led to a successful balance control of the bicycle in real life test.

4

Design of a Realistic Path Tracking Control with Position Measurement Feedback Loop

Path tracking is another important control implementation on the autonomous bicycle. It is implemented in the simulation environment (not on the bicycle yet), where the position of the bicycle is estimated by integrating the velocity (see Figure 2.6). The principal upgrade discussed in this Chapter is the addition of a position measurement system combined to a filter, in order to improve the position estimation.

First, the computation of position estimation is discussed : the integration of measure velocity leads to poor estimation results, getting worse over time. Then, a direction position measurement system is added and the Kalman filter is chosen on the basis of a simple example. After being tuned, the filter is integrated in the autonomous bicycle simulation environment where the feedback loop and the path tracking control are adapted appropriately. Finally, the implementation of the path tracking on the bicycle is tested in simulation and the impact of the model's accuracy is investigated.

4.1 Computation of the Position Estimation

The position is estimated by integrating the velocity in simulation. The main problem about this method is that it cannot be implemented on the real bicycle without issues. As the measurement of the velocity contains errors, their integration will generate an increasing deviation of the position estimation in comparison to the true position (see Figure 4.1).

Figure 4.1 illustrates this issue on the basis of a simple one dimension example. The velocity is set to 3 m/s for the 10 first seconds. Then, it is decreased back to 0. The true position is plotted in green and the position estimation obtained by integrating the measured velocity is shown in blue. The measured velocity is biased of 0.05 m/s, representing the possible errors of the measurement. As explained in the introduction of this Section, the integration of the measurement errors generates the deviation of this position estimation with time.

To avoid this problem, a direct position measurement system should be added on the autonomous bicycle. However, the error generated on the position estimation of this type of measurement system is usually significant (see Figure 4.1).

In conclusion, integration of velocity is accurate in high frequencies but not in low frequencies and direct position measurement is accurate in low frequencies but not in high frequencies. Therefore, a filter has to be implemented.

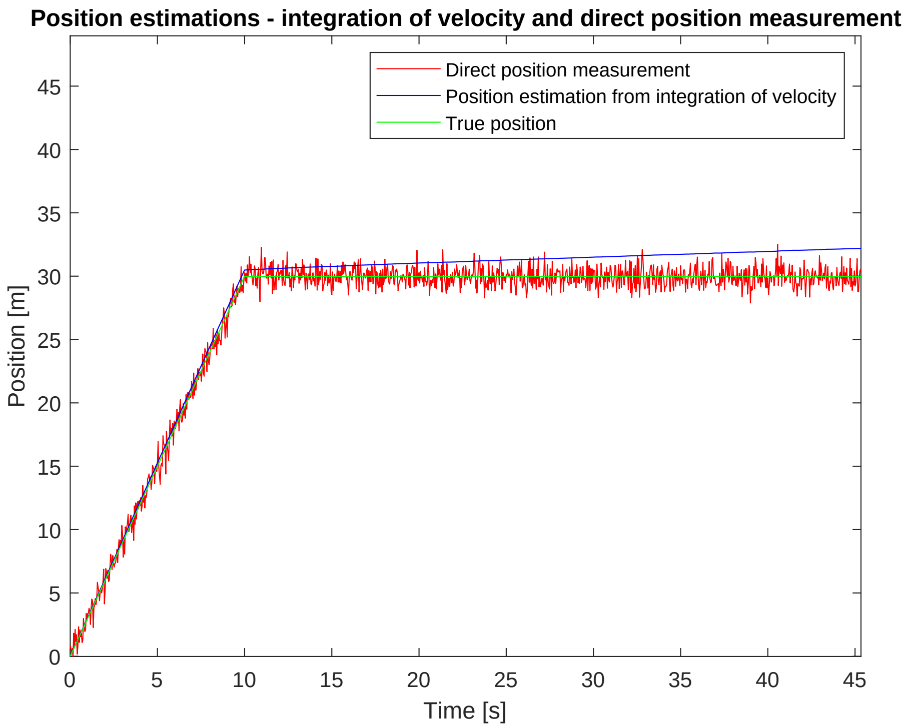

Figure 4.1: Comparison of the position estimations with regards to the true position - Velocity integration curve shows increasing error - Direct position measurement curve shows high frequency noise.

4.2 Processing of the Sensor's Signals

As discussed in the previous Section, a filter has to be implemented as integration of the velocity is not reliable and direct position measurements are usually quite noisy.

In order to choose the adequate filter, a set of three filters is tested on the basis of the simple example described in the previous Section:

- Low Pass filter
- Complementary filter
- Kalman filter

4.2.1 Low pass filter

This first filter simulated is the low pass filter. It is applied to the direct position measurement to obtain a smoother estimation of the position by attenuating high frequencies of the measurements.

The simulation results are shown on Figure 4.2. True position is plotted in light green (following the same path as in Figure 4.1), direct position measurement is plotted in red and low pass filter estimation is plotted in blue. The position estimation is clearly delayed compared to the true position and direct measurement.

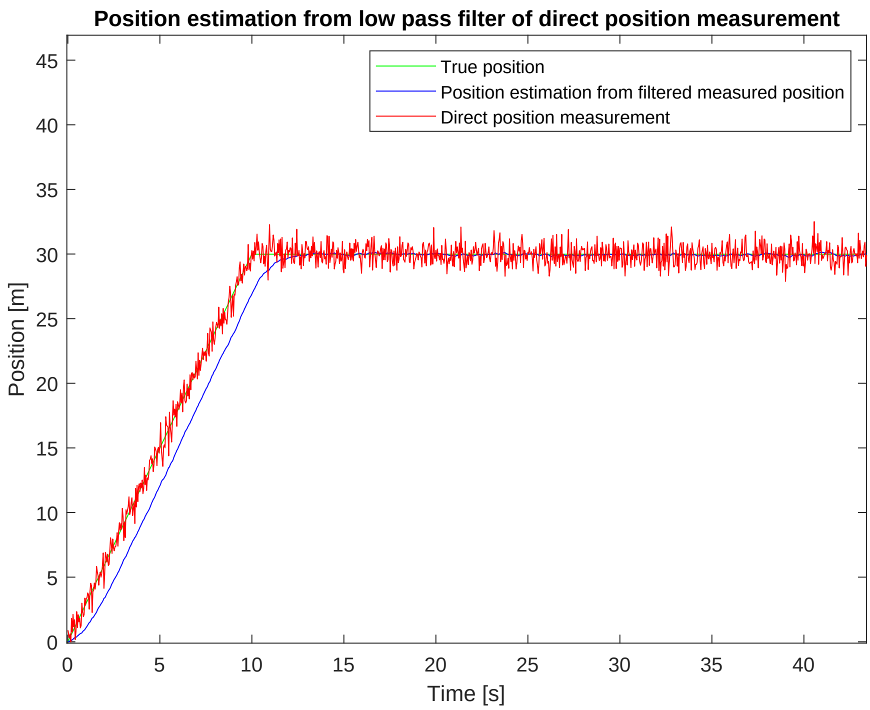

Figure 4.2: Position estimation from filtered position measurement (low pass filter cut-off frequency $f_c = 0,16\,\mathrm{Hz}$) - Significant delay of the position estimation compared to the true position.

For the autonomous bicycle application, the position can change fast. In order to track the trajectory properly, the controller has to get an accurate estimation in real time. Therefore, a low pass filter is not suitable for path tracking control on the autonomous bicycle.

4.2.2 Complementary filter

In order to combine the direct position measurement, and the integrated velocity measurement, the complementary filter is presented here as an easy solution. A gain is defined to set the influence of the estimations on the output signal of the filter [10]:

$$\hat{x} = C\,\hat{x}_{pos\,meas} + (1 - C)\,\hat{x}_{int\,meas\,vel} \tag{4.1}$$

Where $\hat{x}$ is the filtered position estimation, C is the complementary filter parameter $(0.5 < C < 1)$, $\hat{x}_{pos\,meas}$ is the direct position measurement and $\hat{x}_{int\,meas\,vel}$ is the integration of measured velocity.

The gain used in this simulation is set to 0.6, meaning that the filter gives more importance to the position measurement than to the velocity integration.

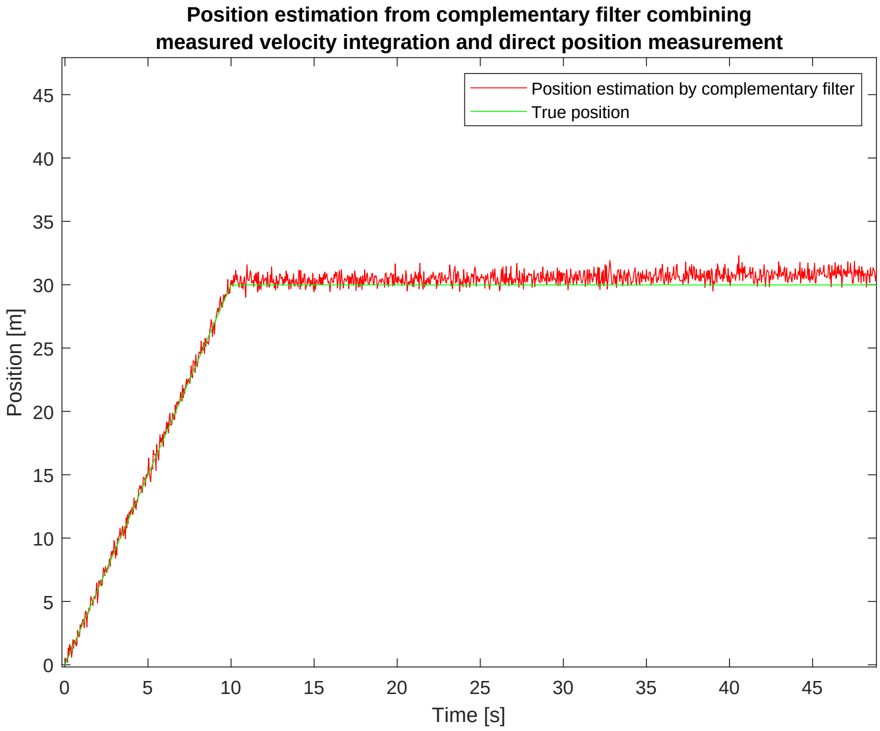

Figure 4.3: Position estimation from complementary filter, combining measured velocity integration and position measurement for C=0.6 - Important noise and deviation with time of the estimation compared to the true position.

The simulation results for this filter implementation can be observed on Figure 4.3. Disadvantages discussed previously can still be observed on the complementary filter response: noisy position measurement and deviation of the integrated velocity estimation. Modifying the gain only reduces one disadvantage while increasing the other. Therefore, this filter is not suitable for the path tracking of the bicycle and another filter should be investigated.

4.2.3 Kalman filter

When the states of a system can only be estimated by an observer, the states of a system can only be determined indirectly, or their direct measurement is affected by an important noise, the recommended choice is the Kalman filter [11]. This filter presents a more complex theoretical background. Therefore, the first Section summarises the essential equations defining the principle of the Kalman filter as presented in [12]. Then, it is implemented on the simple example used until now.

4.2.3.1 Kalman Filter Theory

The general state space representation of a system has the following expression:

$$x_{k+1} = A\,x_k + B\,u_k + w_k \tag{4.2}$$

$$y_k = C\,x_k + D\,u_k + z_k \tag{4.3}$$

Where x_k is the system's state vector at time k, u_k is the system's input vector at time k, w_k is the process noise at time k, A is the state matrix, B is the input matrix, y_k is the output vector at time k, z_k is the measurement noise at time k, C is the output matrix and D is the feed-through matrix.

For the example developed in this Section, the state space representation defined by (4.2) and (4.3) becomes:

$$\begin{bmatrix} s_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & Ts \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_k \\ v_k \end{bmatrix} + \begin{bmatrix} w_k^s \\ w_k^v \end{bmatrix} \tag{4.4}$$

$$\begin{bmatrix} y_k^s \\ y_k^v \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_k \\ v_k \end{bmatrix} + \begin{bmatrix} z_k^s \\ z_k^v \end{bmatrix} \tag{4.5}$$

Where s_k is the position state at the time k, v_k is the velocity state at the time k, w_k^x is the process noise on the state x at time k, z_k^x is the measurement noise on the state x at time k and T_s is the sample time.

The Kalman filter uses the model to predict the **a priori estimates** of the states, $\hat{s}_{k+1}^-$ and $\hat{v}_{k+1}^-$:

$$\begin{bmatrix} \hat{s}_{k+1}^- \\ \hat{v}_{k+1}^- \end{bmatrix} = \begin{bmatrix} 1 & Ts \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{s}_k^+ \\ \hat{v}_k^+ \end{bmatrix} \tag{4.6}$$

$$P_k^- = A\,P_{k-1}^+ A^T + Q \tag{4.7}$$

Where P_k^- is the a priori estimate noise covariance and Q is the process noise covariance.

Then, it updates the states' estimation by combining the measurements to the a priori estimations. As a result, the **a posteriori estimates** $\hat{s}_k^+$ and $\hat{v}_k^+$ are obtained:

$$\begin{aligned} \begin{bmatrix} \hat{s}_k^+ \\ \hat{v}_k^+ \end{bmatrix} &= \begin{bmatrix} \hat{s}_k^- \\ \hat{v}_k^- \end{bmatrix} + \begin{bmatrix} K_k^s \\ K_k^v \end{bmatrix} \left(\begin{bmatrix} y_k^s \\ y_k^v \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{s}_k^- \\ \hat{v}_k^- \end{bmatrix} \right) \\ &= \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} K_k^s \\ K_k^v \end{bmatrix} \right) \begin{bmatrix} \hat{s}_k^- \\ \hat{v}_k^- \end{bmatrix} + \begin{bmatrix} K_k^s \\ K_k^v \end{bmatrix} \begin{bmatrix} y_k^s \\ y_k^v \end{bmatrix} \end{aligned} \tag{4.8}$$

$$P_k^+ = (I - K_k\,C)\,P_k^- \tag{4.9}$$

$$K_k = P_k^-\,C^T (C\,P_k^-\,C^T + R)^{-1} \tag{4.10}$$

Where $\hat{x}_k^+$ is the update estimation of the state x at the time k, $\hat{x}_k^-$ is the predicted estimation of the state x at the time k, K_k^x is the Kalman's gain for the state x at time k, P_k^+ is the a posteriori estimate noise covariance and R is the measurement noise covariance.

The Kalman's gains K_k^x are calculated in order to find the optimal a posteriori estimates with the minimum covariance. In the simulation environment, a Kalman filter block can be used: it determines the optimal gains and computes the position estimation obtained by (4.8). As shown in (4.10), the process and measurement noise covariances (Q and R respectively) are needed to determine Kalman's gain. Thus, both covariances have to be defined.

The **measurement noise covariance** R is directly linked to the accuracy of the measurement instruments. In our case, the velocity is measured by a Hall sensor which the standard deviation is evaluated at $\pm 0.056\,\mathrm{m/s}$ in [1]. Therefore, the covariance can be obtained as follows:

$$R_{velocity} = E[z_2\, z_2^T] = \sigma_{velocity}^2 = (0,056)^2 = 0,0031 \tag{4.11}$$

As there is no position measurement system on the bicycle for now, this example considers using a Global Positioning System (GPS) at first and the simulation results will indicate if a more precise instrument is needed. In [13], the average error of GPS is approximated at $0,715\,\mathrm{m}$ (in 95% of the cases for the month of May 2016). Thus, this is the supposed accuracy used for the direct measurement system:

$$R_{position} = E[z_1\, z_1^T] = \sigma_{position}^2 = (0,715)^2 = 0,51 \tag{4.12}$$

(4.11) and (4.12) are combined:

$$R = \begin{bmatrix} 0,51 & 0 \\ 0 & 0,0031 \end{bmatrix} \tag{4.13}$$

For the **process noise covariance** Q, it can be tuned in order to make the filter output match the wanted results [14]. After several iterations, the following matrix is obtained:

$$Q = \begin{bmatrix} 0,0001 & 0 \\ 0 & 0,1 \end{bmatrix} \tag{4.14}$$

4.2.3.2 Comparison of Position Estimation Results with Integrated Velocity and Direct Position Measurement Estimations

The system's state space representation and the noise covariance matrices being determined, the filter is implemented in Simulink. As in (4.4), there is no input to the system as shown in Figure 4.4.

Figure 4.5 shows the simulation results for the simple example where the estimations can be compared to the true position:

- Integration of measured velocity (in blue)
- Direct position measurement (in red)
- Kalman filter (in black)

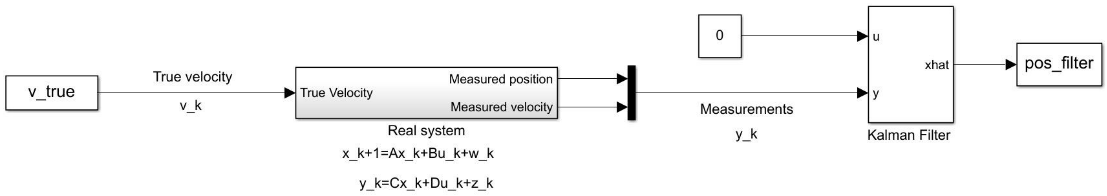

Figure 4.4: Kalman filter - Simple example implementation in Simulink.

The filtered signal is close to the true position for the whole duration of the simulation. There is no deviation observed for the first 10 seconds. Afterwards, the filtered position estimation presents a small offset.

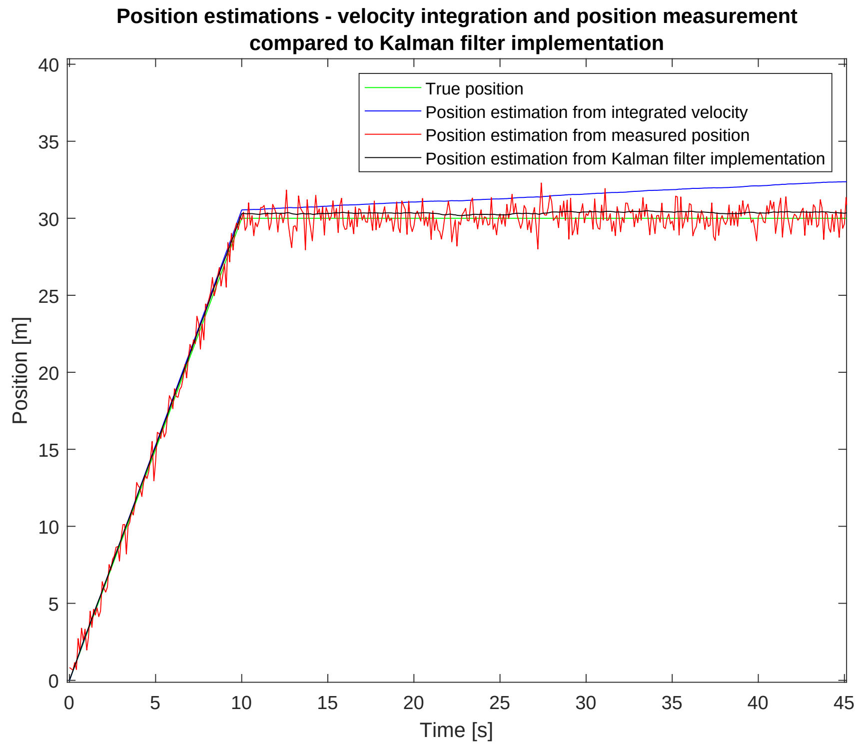

Figure 4.5: Kalman filter position estimation compared to measured velocity integration and direct position measurement - Kalman filter is much closer to the true position than the integrated velocity and does not show high frequency noise like direct position measurement.

As each estimation methodology generates different errors with regards to the true path, Figure 4.6 compares the absolute error between the estimations and the true position. First, the most significant error is the one generated by the direct position measurement. Then, the deviation of the estimation from the integration of measured velocity becomes more important, reaching $2,5\,\mathrm{m}$ after 47 seconds of simulation. With the Kalman filter, the error compared to the true position remains under $0.5\,\mathrm{m}$.

The comparison between the position estimations shows the positive effect of using the Kalman filter.

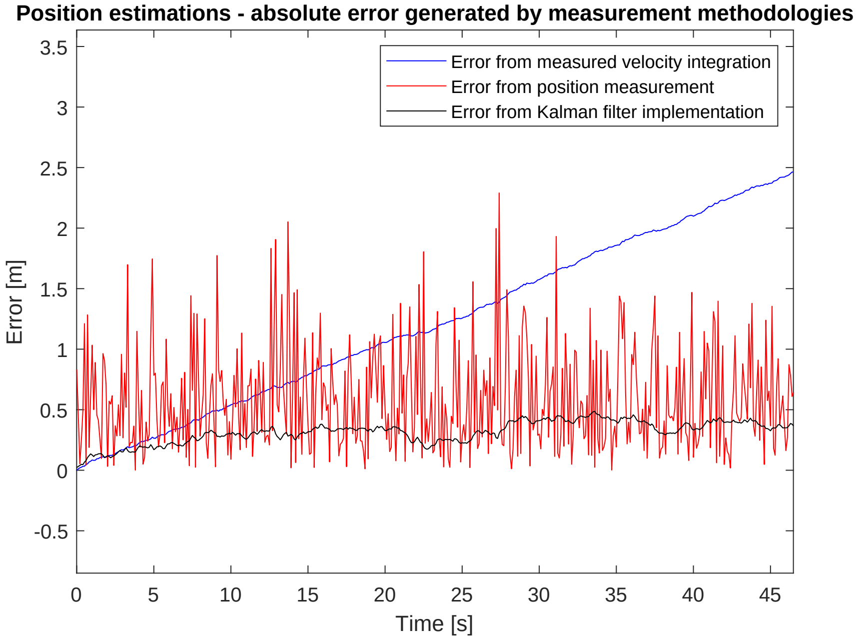

Figure 4.6: Absolute error between estimations and true position - Kalman filter estimation shows the lowest error in average.

4.3 Implementation of the Kalman Filter in the Feedback loop

In this Section, the position measurement system is implemented in the bicycle's simulation environment implying some changes on the position feedback loop design (see Figure 4.7).

In order to implement the position measurement and Kalman filter in the feedback loop, the simplified system developed in Section 4.2.3.1 is slightly modified. In the autonomous bicycle's simulation environment, the position and the velocity are given in two different directions x and y. Therefore, the model represented by (4.4) and (4.5) becomes:

$$\begin{bmatrix} s^x_{k+1} \\ s^y_{k+1} \\ v^x_{k+1} \\ v^y_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & Ts & 0 \\ 0 & 1 & 0 & Ts \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s^x_k \\ s^y_k \\ v^x_k \\ v^y_k \end{bmatrix} + \begin{bmatrix} w^{sx}_k \\ w^{sy}_k \\ w^{vx}_k \\ w^{vy}_k \end{bmatrix} \tag{4.15}$$

$$\begin{bmatrix} y^{sx}_k \\ y^{sy}_k \\ y^{vx}_k \\ y^{vy}_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s^x_k \\ s^y_k \\ v^x_k \\ v^y_k \end{bmatrix} + \begin{bmatrix} z^{sx}_k \\ z^{sy}_k \\ z^{vx}_k \\ z^{vy}_k \end{bmatrix} \tag{4.16}$$

Where s_k^x and s_k^y are the positions in directions x and y respectively and, v_k^x and v_k^y are the velocities in directions x and y respectively.

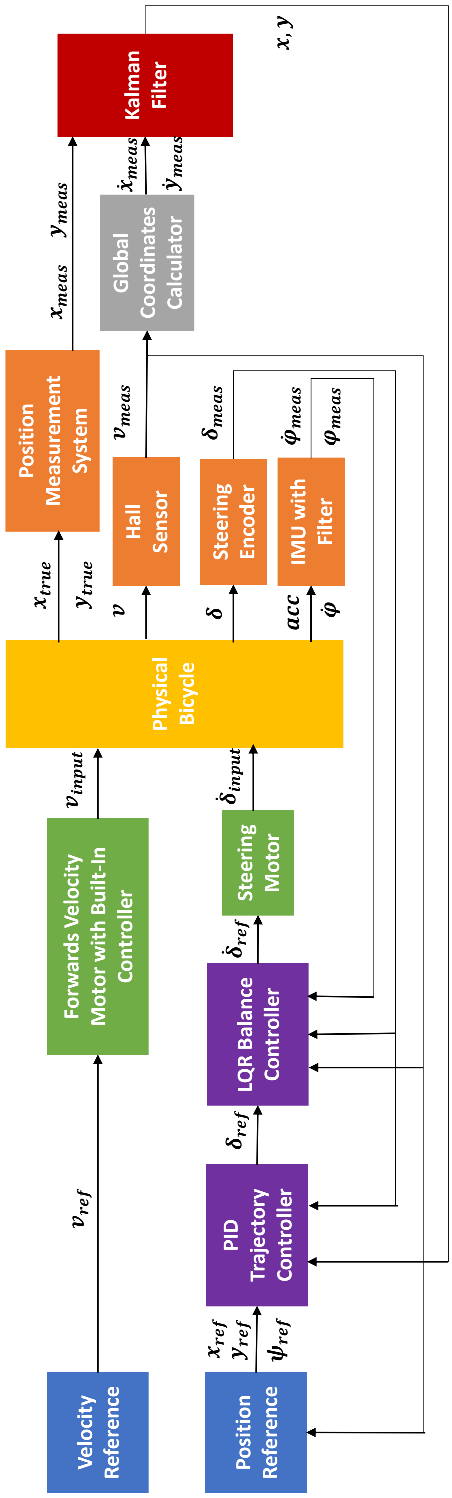

Figure 4.7: Modified simulation environment – Implementation of a Position Measurement System and a Kalman Filter in the path tracking loop.

The expressions of the a priori and a posteriori estimations involve more terms but the principle remains the same as in the previous Section.

4.3.1 Validation of the Kalman Filter in the Feedback loop: Simulation Results

The trajectory controller input in the simulation environment (Figure 2.6) is modified, the feedback loop being upgraded with the position measurement and the Kalman filter (Figure 4.7). A first closed loop simulation is performed on a straight path. The new implementation in the simulation environment is then assessed on a circle path and illustrates the Kalman filter's accuracy with regards to the actual path compared to the other estimations.

The improved feedback loop considers velocity and position measurements with their associated noise covariance. Both measurements are combined in a Kalman filter to estimate the bicycle's position.

Conditions

- Velocity: $v = 3\,\mathrm{m/s}$
- Ideal initial conditions: $\varphi_0 = 0°$, $\delta_0 = 0°$, $\dot{\varphi}_0 = 0°/\mathrm{s}$
- Point of mass model
- LQR settings based on the point of mass model: $Q = \mathrm{diag}([100, 100, 10])$, $R = 42$

Observations

Figures 4.8 and 4.9 show the simulated bicycle path compared to the reference path:

- Reference path tracked by the bicycle (in blue).
- Variation of direct position measurement due to its noise (in green).
- Deviation with time of the measured velocity integration (in purple).
- Kalman filter estimation based on which the trajectory is controlled (in black).
- Actual position of the bicycle (in red).

The estimation from the Kalman filter is close to the actual path. The direct position measurement shows a noisy estimation while the variations of the actual path, the Kalman estimation and the velocity integration are similar. Yet, the position estimation from the velocity integration deviates from the actual position towards the end of the path.

To make the comparison easier, Figure 4.10 illustrates the errors generated on the actual path by considering the integration of velocity and the Kalman filter estimations. As the position is described in two directions, these errors are computed as Euclidean distance which is defined as follows:

$$pos_{error} = \sqrt{(x_{est} - x_{bicycle})^2 + (y_{est} - y_{bicycle})^2} \tag{4.17}$$

Where pos_{error} is the position error, x_{est} and y_{est} are the coordinates of the estimated position and, $x_{bicycle}$ and $y_{bicycle}$ are the actual coordinates of the bicycle.

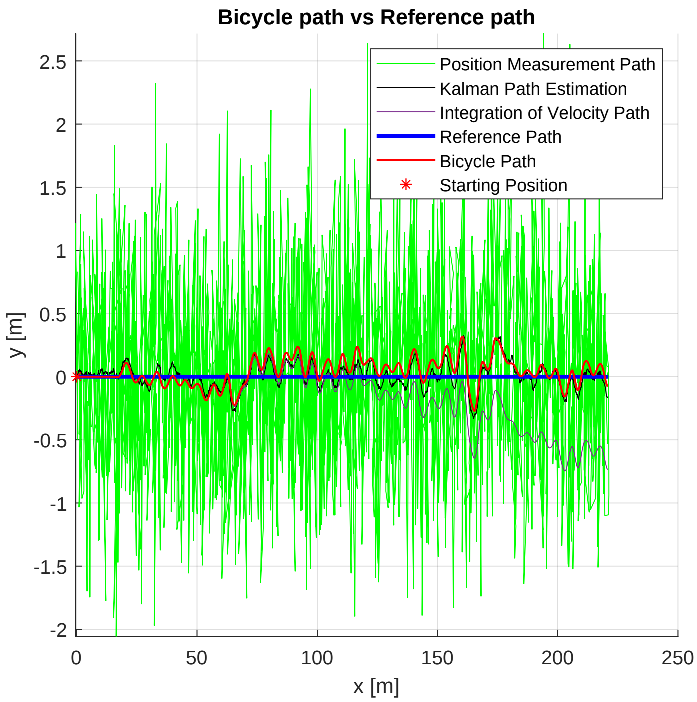

Figure 4.8: Simulation of straight path tracking - Kalman path estimation is the closest to the actual bicycle path, while the noise of the direct position measurement and the deviation of the integrated velocity are highlighted.

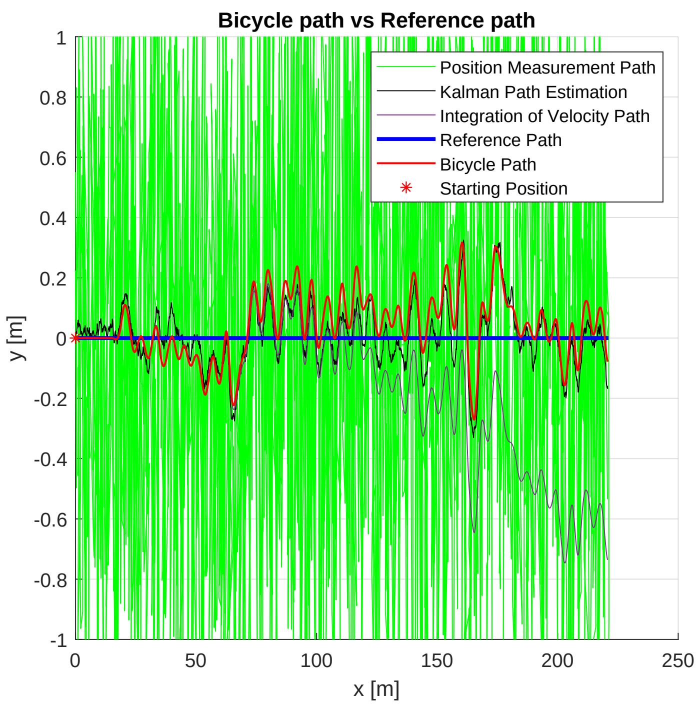

Figure 4.9: Simulation of straight path tracking zoomed in - Kalman path estimation is the closest to the actual bicycle path, while the noise of the direct position measurement and the deviation of the integrated velocity are highlighted.

Kalman estimation and integrated velocity both present relatively small errors in the first 35 seconds. Then the deviation of the velocity integration is noticeable and increases until the end of the simulation. Overall, the Kalman filter estimation generates an error of less than 0.25 m compared to the actual position.

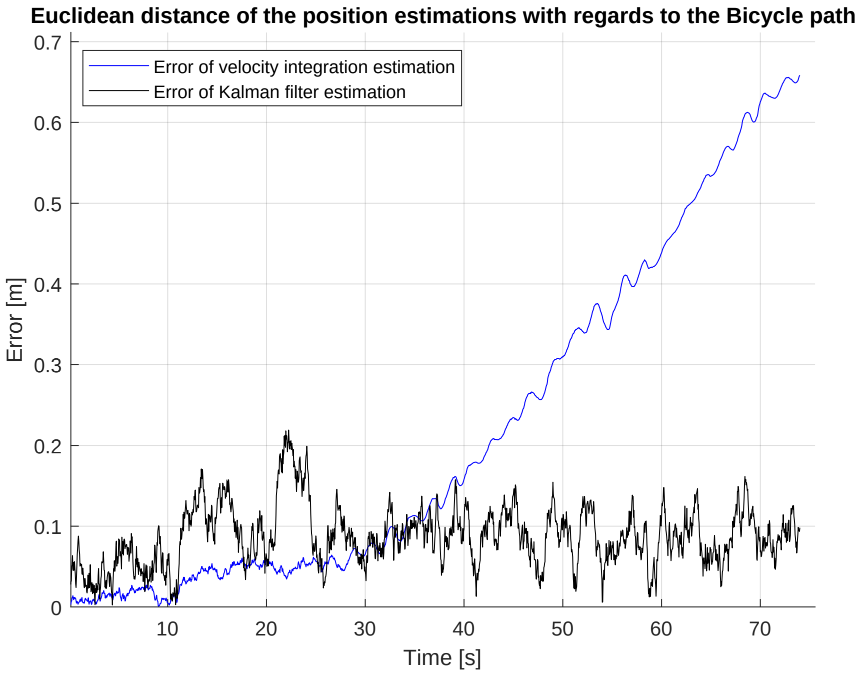

Figure 4.10: Comparison of the Euclidean distance error for straight path tracking - Error of the Kalman filter estimation is lower than 0.25 m for the whole duration of the simulation. The integrated velocity estimation error is forever increasing and is lower than for the Kalman filter estimation in the first 40 seconds.

Figures 4.11 and 4.12 show the sames results for a circular path. The Kalman filter shows again an excellent performance, the estimation error remaining under 0.3 m.

Conclusion

With a measured velocity and a GPS based position measurement, simulation results show excellent results of the Kalman Filter estimation of the bicycle's position. This allows the trajectory controller to track a reference path with an accuracy in the range of less than 0.3 m.

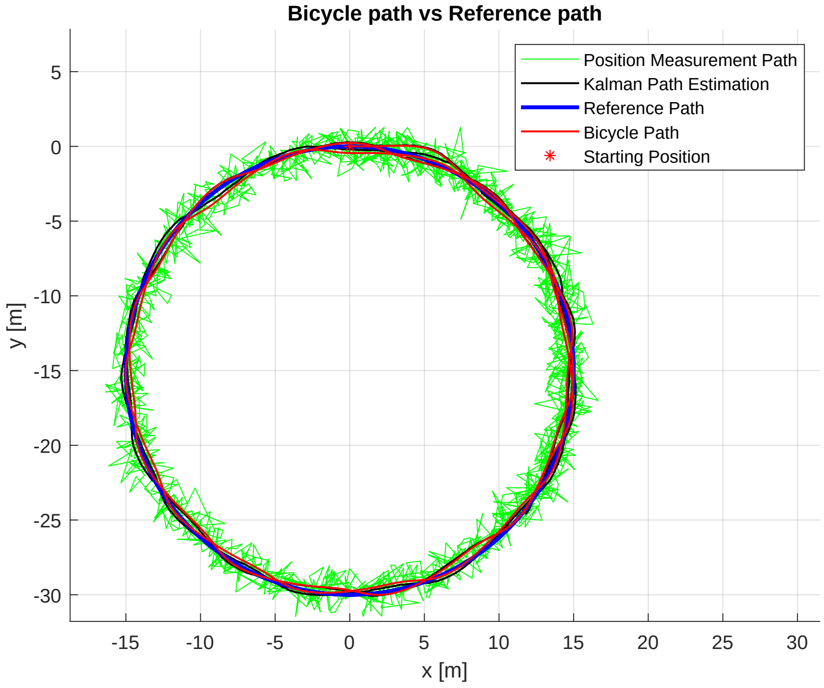

Figure 4.11: Simulation of circle path tracking - Based on the Kalman filter estimation, the bicycle is able to track the reference path closely.

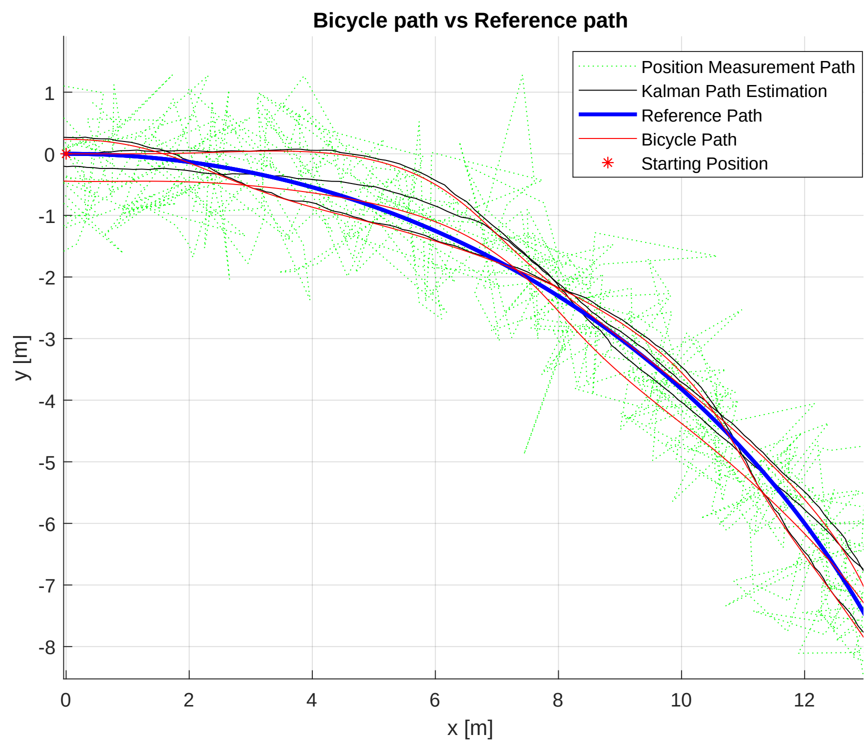

Figure 4.12: Simulation of circle path tracking zoomed in - Based on the Kalman filter estimation, the bicycle is able to track the reference path closely.

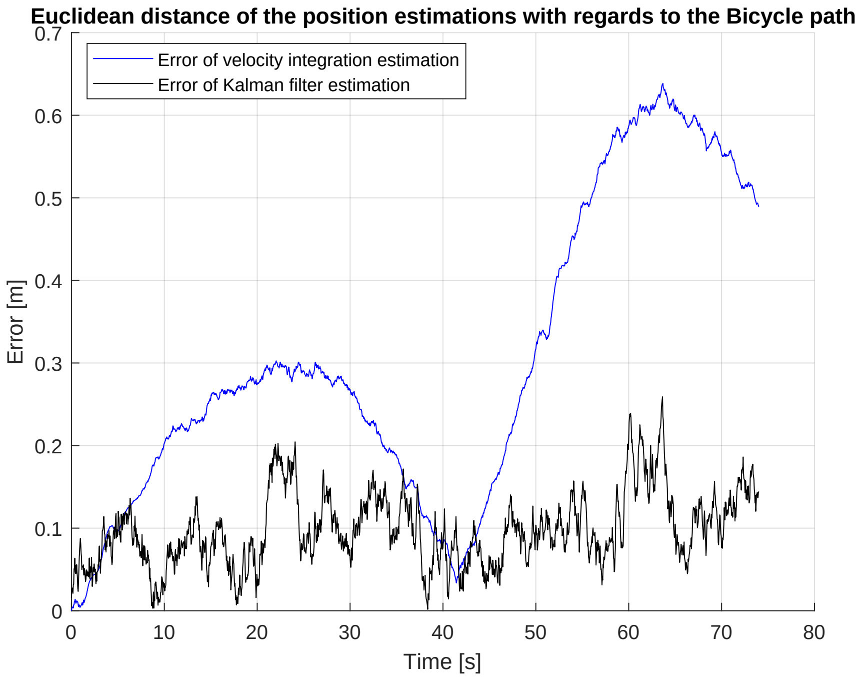

Figure 4.13: Comparison of the Euclidean distance error for circle path tracking - Kalman filter estimation error is lower than for integrated velocity for the whole duration of the simulation and stays under 0.3 m.

4.4 Impact of the Model on the Path Tracking Capabilities

In Section 3.1, a more realistic model was studied and generated significant changes in the results of the balancing simulation. This section studies the impact of using the two volumes model in path tracking simulations.

The simulation conditions are the following:

- Velocity: $v = 3\,\mathrm{m/s}$
- Ideal initial conditions: $\varphi_0 = 0°$, $\delta_0 = 0°$, $\dot{\varphi}_0 = 0°$
- Parameters of the LQR for balancing control:
  - $Q = \mathrm{diag}([100\,100\,10])$ and $R = 42$ for the point of mass model
  - $Q = \mathrm{diag}([70\,50\,10])$ and $R = 42$ for the two volumes model

  Remark: the LQR is based on the point of mass model. The parameters are different so that the two volumes model can balance properly for the whole duration of the simulation.

Observations

Both models balance and track the reference path successfully with different accuracy. Figure 4.14 shows small deviations of the bicycle of less than 0.2 m on a straight path when simulating the point of mass model. For the two volumes model,

Figure 4.15 illustrates much bigger variations of approximately 0.8 m.

Conclusion
This simulation demonstrates the key influence of the moment of inertia on the behaviour of the bicycle. Comparing both models, the point of mass is clearly optimistic. This confirms the need for a better model (e.g. the two volumes model) in order to predict the behaviour of the bicycle more realistically.

In consequence, to get a closer tracking of the reference path, the LQR should be re-designed based on the two volumes model. The trajectory controllers might have to be tuned to improve the tracking performance of the bicycle.

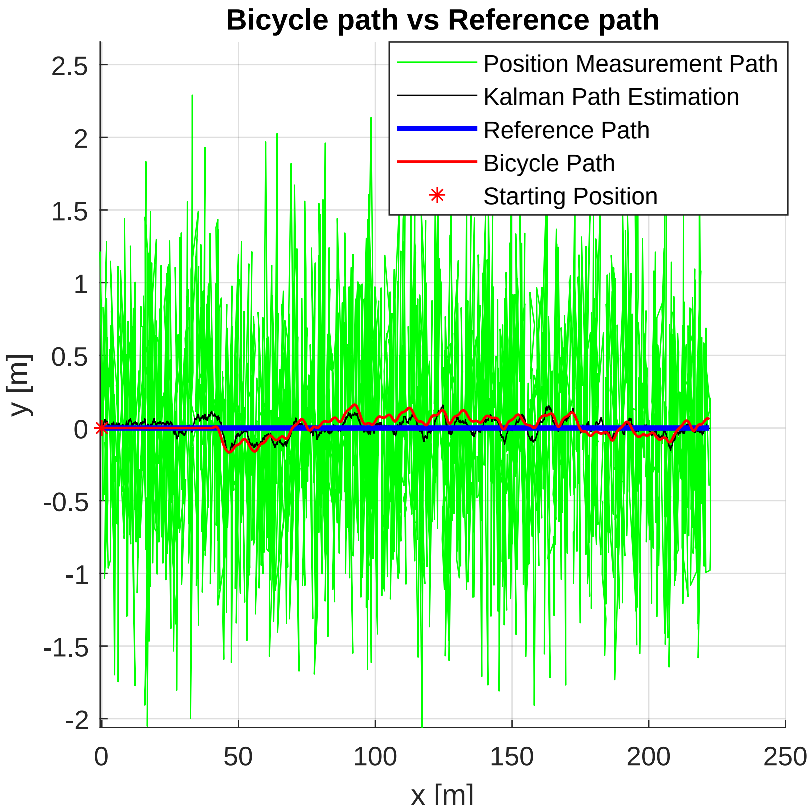

Figure 4.14: Impact of the model on path tracking capabilities - Point of mass model - The deviation of the bicycle path is in a range of 0.2 m from the reference path.

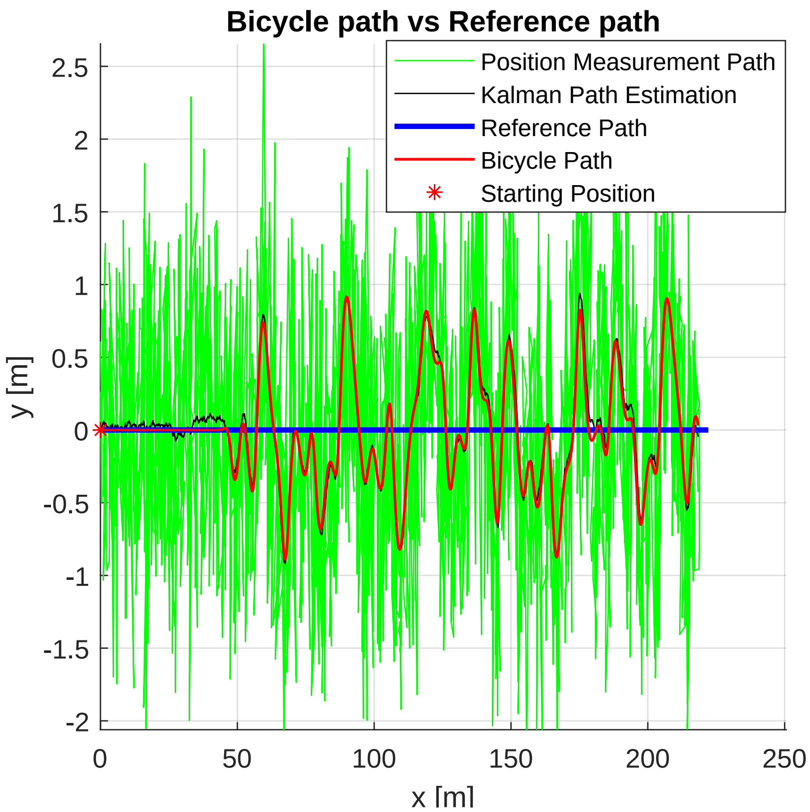

Figure 4.15: Impact of the model on path tracking capabilities - Two volumes model - The deviation of the bicycle path is in a range of 1 m from the reference path compared to 0.2 m with the point of mass model. This illustrates the influence of the higher moment of inertia of this more realistic model.

5 Assessment of the Bicycle Path Tracking Performance

In order to assess the path tracking performances of the autonomous bicycle, this Chapter develops path tracking indicators and test cases.

Path tracking indicators are defined to compare performances between several tests (to assess their severity), or between simulations and test drives (to evaluate the accuracy of the simulation's predictions).

Test cases are designed in order to test the bicycle path tracking capabilities in different situations, from basic (straight path) to more elaborated tracks (successive changes of direction).

First, the path tracking indicators are defined. Then, several test cases are presented and compared, showing the increasing difficulty of execution and variation of the path tracking indicators.

5.1 Path Tracking Performance Indicators

Path Tracking Performance Indicators constitute a useful instrument to measure the capability of the bicycle to follow a reference path. They are used for the following purposes :

- For a given reference path, evaluate and compare the effects of different control settings on the path tracking performance of the bicycle. This allows proper tuning by simulation before running test drives.
- For a given bicycle configuration and controllers setting, evaluate (by simulation) or measure (during test drives) its capability to follow several typical trajectories. It identifies among those which ones are the most difficult to track.

Three different types of indicators have been defined:

- Maximal deviation (in meters) between real path and reference

$$Xmax_{ind} = \max(x_{bicycle}) \tag{5.1}$$

$$Ymax_{ind} = \max(y_{bicycle}) \tag{5.2}$$

Where $x_{bicycle}$ is the position of the bicycle in the x direction, $y_{bicycle}$ is the position of the bicycle in the y direction.

- Average absolute deviation of the real path compared to the reference

$$Xpath\,error_{ind} = \frac{\sum_{i=0}^{n} |x_{ref\,i} - x_{bicycle\,i}|}{n} \tag{5.3}$$

$$Ypath\,error_{ind} = \frac{\sum_{i=0}^{n} |y_{ref\,i} - y_{bicycle\,i}|}{n} \tag{5.4}$$

  Where n is the maximum number of sample time and i is the number of the time step.
- Euclidean distance similar to (4.17)

$$EuclDist_{ind} = \frac{\sum_{i=0}^{n} \left(\sqrt{(x_{ref\,i} - x_{bicycle\,i})^2 + (y_{ref\,i} - y_{bicycle\,i})^2} \right)}{n} \tag{5.5}$$

Then, another indicator assesses the velocity tracking efficiency:

$$v_{ind} = \frac{\sum_{i=0}^{n} |v_{ref\,i} - v_{bicycle\,i}|}{n} \tag{5.6}$$

Where v_{ind} is the indicator of velocity, v_{ref} is the reference velocity and $v_{bicycle}$ is the velocity of the bicycle.

5.2 Test Cases

The purpose of developing test cases is to assess the tracking capabilities of the bicycle on more complex paths and determine its limitations. After implementing the trajectory controller on the bicycle as described in Chapter 4, it will be possible to perform the test cases and the performance indicators will be computed in order to be compared to the simulation results of the same case.

An important difference between simulation and reality should be kept in mind: initialisation of the bicycle during test drives is done by hand. Therefore, the initial roll angle can vary within a few degrees range. Depending on this value, the bicycle will behave differently. The same problem occurs for the steering angle but usually in a smaller range.

In order to obtain the most accurate results with regards to the reality, the simulations showed in this Section are performed in the following conditions:

- Velocity: $v = 3\,\mathrm{m/s}$
- Initial conditions: $\varphi_0 = 2°$, $\delta_0 = 0,5°$, $\dot{\varphi}_0 = 0°/\mathrm{s}$
- Radius of the curves: $r = 15\,\mathrm{m}$
- Point of mass model
- LQR adapted to the point of mass model: $Q = \mathrm{diag}([100, 100, 10])$, $R = 42$

In the results of this Section, the simulations stop at different times, not because the bicycle fell but because the path was only defined until that point. All of the simulated paths illustrated in this Section were successful in balancing the bicycle. Only the performances of the path tracking are discussed.

5.2.1 Straight path

The straight path assesses the ability of the bicycle to track a simple path as a first test.

The bicycle path is shown in Figure 5.1. The first oscillation is very significant due to the non-zero initial conditions. At any moment, the bicycle drives within a $[-0.15; +0.15]$ m range around the reference for a path more than 200 m long.

In Figure 5.2, the states estimations of the bicycle can be observed. Apart for the higher initial oscillation, the roll and steering angles both oscillate between $[-1.5; +1.5]\,^\circ$ and $[-3; +4.5]\,^\circ$ respectively. The oscillations do not increase at any moment, demonstrating a good combination of balancing and path tracking controllers.

In order to compare the test cases severity and to compare the simulation results to the future test drives, the performance indicators are regrouped into the Table 5.1.

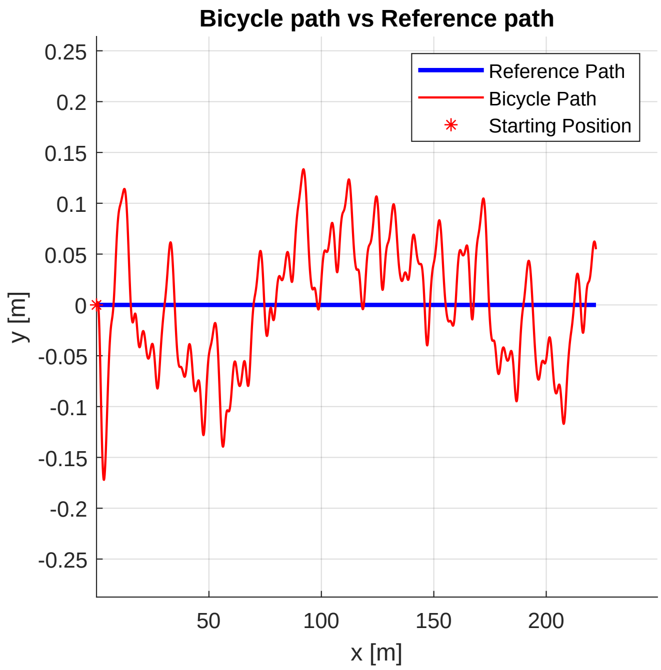

Figure 5.1: Test Case 1 - Straight path tracking simulation - High deviation at the beginning of the path due to the non-zero initial values of the roll and steering angles. Afterwards, the bicycle tracks the reference within a $[-0.15; +0.15]$ m range.

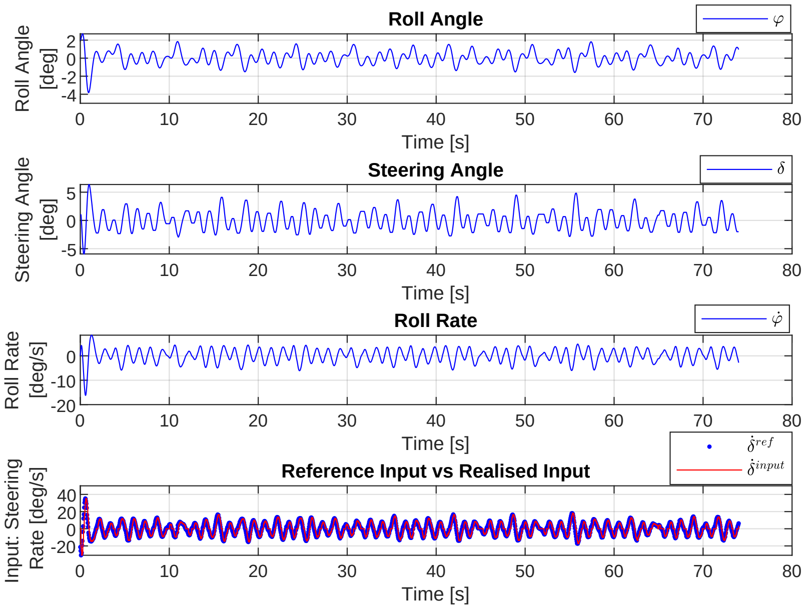

Figure 5.2: Test Case 1 -Straight path tracking simulated states - Initial variation of the states due to non-zero initial conditions.

Performance Indicator	Value
$Xmax_{ind}$	0.09 m
$Ymax_{ind}$	0.17 m
$Xpath\,error_{ind}$	0.03 m
$Ypath\,error_{ind}$	0.05 m
$EuclDist_{ind}$	0.06 m
v_{ind}	0.34 m/s

Table 5.1: Test Case 1 - Straight path tracking performance indicators - Typically, this table of indicators should be compared with other simulation results (e.g. with different settings) and test drives computations.

5.2.2 Change of Direction of 45°

This test case evaluates the ability of the bicycle to perform a change of direction. An important variation of the roll and steering angles are expected while performing the turn. Afterwards, the bicycle should be able to balance itself.

More specifically, the path defined for this second test case presents a 45° turn and the transition is smoothed out with a curve of 15 m radius (see Figure 5.3). Deviations between the bicycle path and the reference are observed during, and just after the turn. Then, the bicycle recovers and gets back on track. On the straight parts, the bicycle follows the reference closely.

The red box in Figure 5.4 highlights the states variation generated during the turn. The steering decreases in the negative values, meaning the bicycle is turning to the right (see Section 2.1.1 for the definition of the angles sign). This variation influences the roll angle which increases. The angles amplitudes reach 9,3° and -10,6° for roll and steering angles respectively. As the bicycle is overshooting to the right of the path, the steering angle increases and becomes positive in order to get back on track. The following oscillations stabilise the bicycle around the straight part.

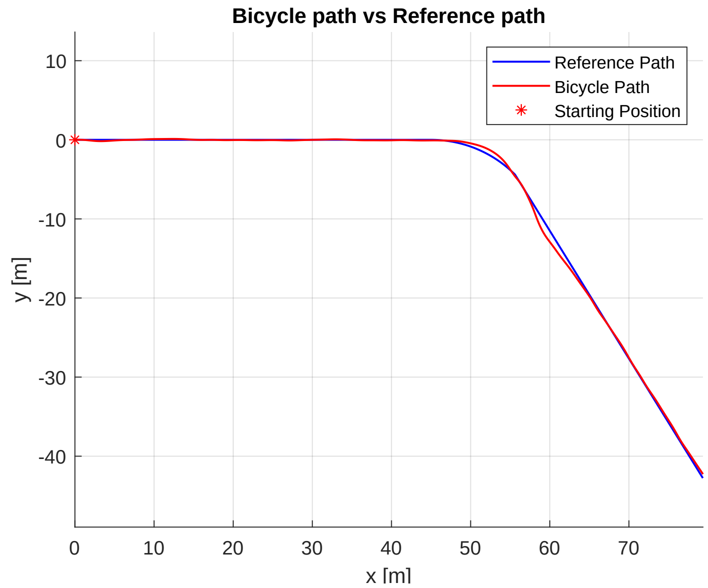

Figure 5.3: Test Case 2 - Change of Direction of 45° path tracking simulation - Deviation of the bicycle path around the curved section.

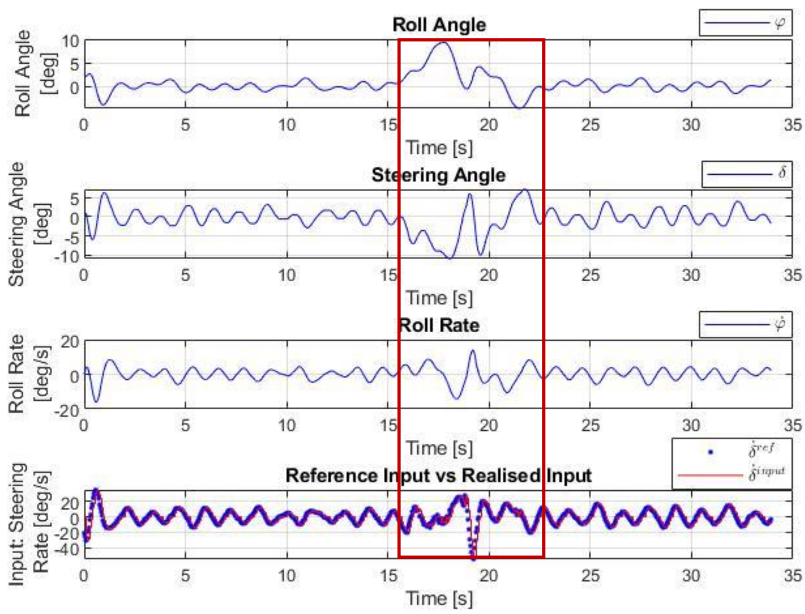

Figure 5.4: Test Case 2 - Change of Direction of 45° path tracking simulated states - Highlight on the change of direction.

As previously, the performance indicators are computed (see Table 5.2). Compared to the performance indicators of the first test case (see Table 5.1), all the indicators values are higher. Significant oscillations were observed due to the turning sequence leading to the increase of path related indicators. The velocity indicator remains fairly similar.

Performance Indicator	Value
$Xmax_{ind}$	0.91 m
$Ymax_{ind}$	0.63 m
$Xpath\,error_{ind}$	0.16 m
$Ypath\,error_{ind}$	0.20 m
$EuclDist_{ind}$	0.30 m
v_{ind}	0.35 m/s

Table 5.2: Test Case 2 - Change of Direction of 45° path tracking performance indicators - Typically, this table of indicators should be compared with other simulation results (e.g. with different settings) and test drives computations.

5.2.3 U-turn

The capability of performing a u-turn can be interesting during test drives. It is also a way of challenging the bicycle controllers for a longer turn than in the second test case. Thus, important variations of the roll and steering angles should be observed for a longer period of time. The bicycle should be able to stabilise after the turn. The difficulty of balancing and tracking is increased.

As for the second test case, the turn presents the biggest deviations of the bicycle path with regard to the reference at the beginning and at the end of the turn (see Figure 5.5). For the other sections of the path, the trajectory tracking seems robust but still presents small variations.

Figure 5.6 shows the system's states and input. At the 10s mark, the angles variation presents the same shape as for the Test Case 2. Comparing to Figure 5.4, the variations of roll and steering angles of the turning sequence are repeated five times in this third test case (see the red box in Figure 5.6) as the turn is much longer.

The overshooting of the bicycle trajectory compared to the reference path are highlighted in green on Figure 5.6. It corresponds to the beginning and the end of the curve in Figure 5.5. The amplitude of the angles are higher at these two moments.

Table 5.3 summarises the different performance indicators of this path. All the indicators linked to the path tracking abilities are better or equivalent to the indicators of the second test case. The x direction indicators are influenced by the straight parts being tracked towards positive and then negative values.

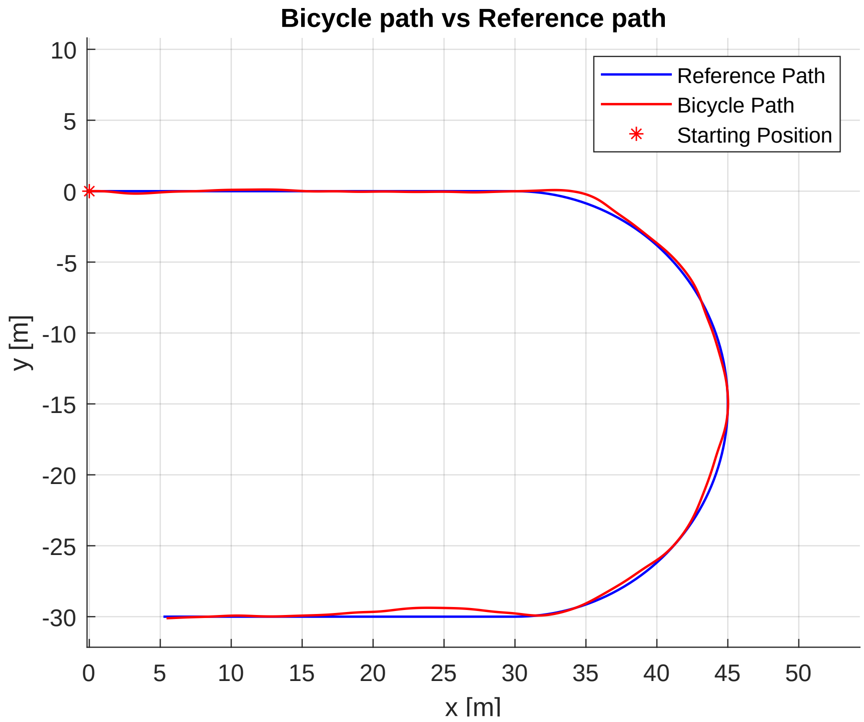

Figure 5.5: Test Case 3 - U-turn path tracking simulation - Highest deviation of the bicycle path observed at the path transitions.

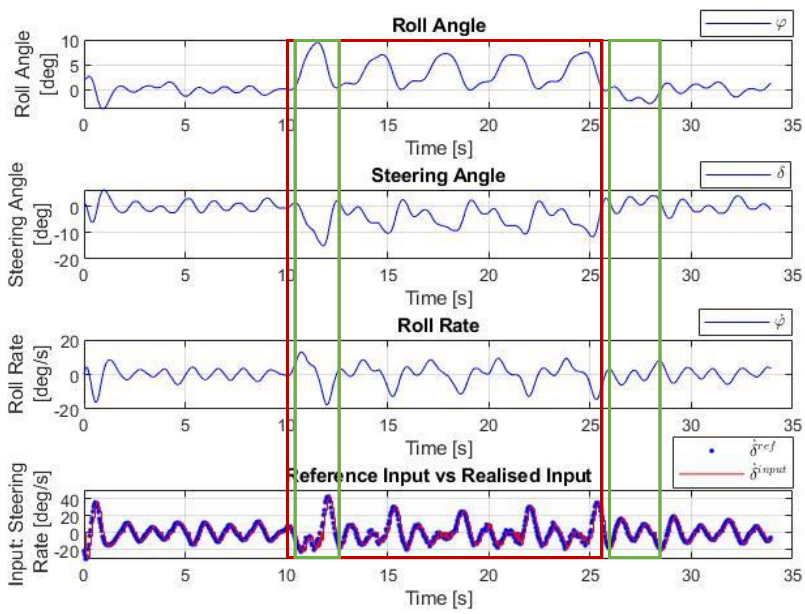

Figure 5.6: Test Case 3 - U-turn path tracking simulated states - Red box shows the control actions during the turning sequence - Green boxes highlight the path transitions (at the beginning and at the exit of the turn).

Performance Indicator	Value
$Xmax_{ind}$	0.31 m
$Ymax_{ind}$	0.64 m
$Xpath\,error_{ind}$	0.12 m
$Ypath\,error_{ind}$	0.23 m
$EuclDist_{ind}$	0.28 m
v_{ind}	0.35 m/s

Table 5.3: Test Case 3 - U-turn path tracking performance indicators - Typically, this table of indicators should be compared with other simulation results (e.g. with different settings) and test drives computations.

5.2.4 Double change of direction

This last test case assesses the ability to sequence two curves of opposite directions. The expected results should be similar to the ones computed for the Test Case 2 as this reference path is a combination of two 45° direction changes (see Figure 5.7). The transition between the two curves should be characterised by a change of sign for the roll and steering angles.

As in Test Cases 2 and 3, the bicycle path presents bigger errors with regards to the reference at the beginning and at the end of the curved section (see Figure 5.7).

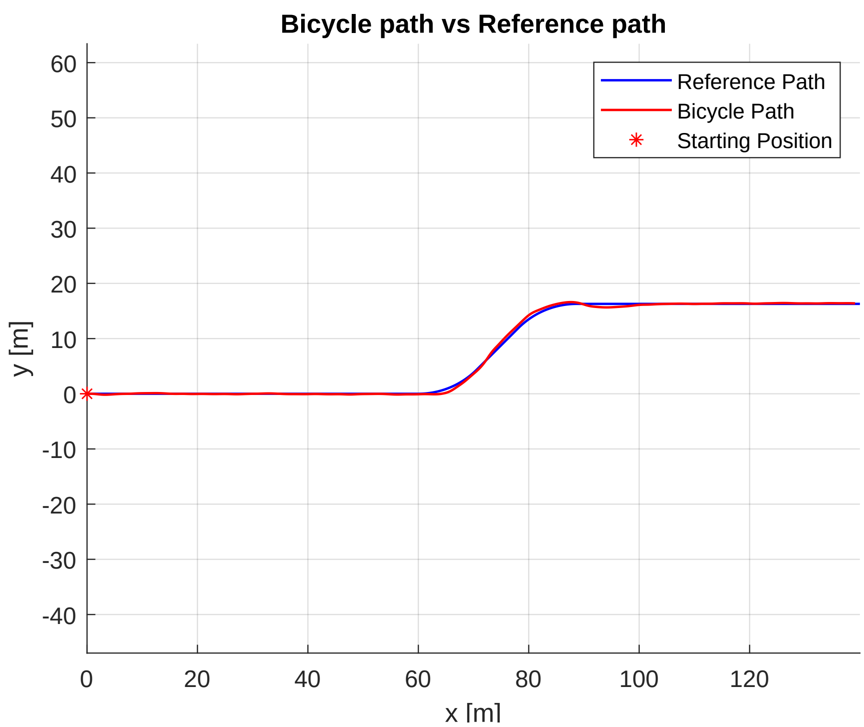

Figure 5.7: Test Case 4 - Double change of direction path tracking simulation - Highest deviation of the bicycle path observed at the path transitions.

As predicted, at the transition between both turns, the roll and steering angles present an inversion of the signs (see the red box Figure 5.8). The turns performed on each side led to similar variations of the angles with opposite signs. The amplitudes of the variations reach [-10;+10]° for the roll and [-11;+12.6]° for the steering.

The performance indicators shown in Table 5.4 indicate bigger errors with regards to the reference compared to the indicators of test case 2 for the x direction. This can be explained by the accumulation of the error for each turn. Overall, the other indicators are similar or worse than the ones of test case 2.

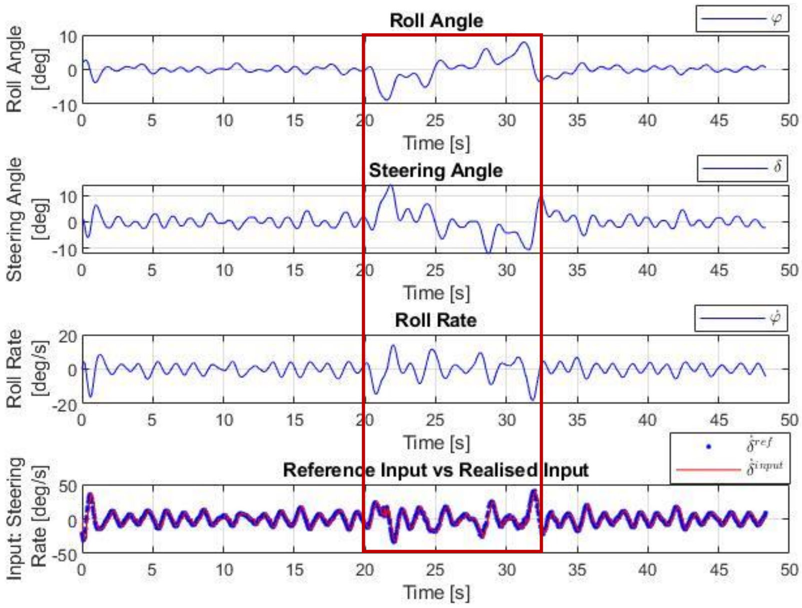

Figure 5.8: Test Case 4 - Double change of direction path tracking simulated states - Red box highlights the bigger control actions during the double change of direction sequence.

Performance Indicator	Value
$Xmax_{ind}$	0.91 m
$Ymax_{ind}$	0.70 m
$Xpath\,error_{ind}$	0.43 m
$Ypath\,error_{ind}$	0.14 m
$EuclDist_{ind}$	0.49 m
v_{ind}	0.29 m/s

Table 5.4: Test Case 4 - Double change of direction path tracking performance indicators - Typically, this table of indicators should be compared with other simulation results (e.g. with different settings) and test drives computations.

In conclusion, all the test cases were simulated in similar conditions as in reality by using non-zero initial conditions. They were all successful in keeping the balance of the bicycle and tracking several paths, from the most basic straight line to a more difficult sequence of turns.

6 Conclusion

In the frame of this Master's Thesis report:

- The high impact of the two volumes model has been demonstrated in balancing and path tracking simulations.
- The balancing control of the bicycle has been corrected in a systematic way leading the bicycle to successfully balance during the test drives.
- The path tracking control, including a direct position measurement and a Kalman filter, has been successfully simulated and presents promising performances.
- The test cases and path tracking performance indicators have been defined and simulated to assess the path tracking capabilities of the bicycle.

On the other hand, several opportunities for future work have been identified:

- Adapt the controllers on the basis of the two volumes model.
- Implement the path tracking feedback loop developed in this thesis.
- Assess the path tracking performances of the bicycle using the indicators and test cases defined in this thesis.

From a user perspective, the following topics could be addressed:

- Improve user friendliness: start and stop sequences, trajectory programming, user manual.
- Reshape the bicycle to keep radar signature equivalent to that of an ordinary bicycle.

On a more personal level, I am very pleased to have been given the chance of joining this very rewarding project, combining both theoretical and practical aspects. All the work performed, together with the rich relationships with my supervisor and all the team members, were for me a great opportunity to learn and to improve my skills. I really enjoyed living this experience in such a great team and I feel proud to have contributed to this challenging project.

Bibliography

[1] Erdinç Umur. Modelling, Validation and Control of an Autonomous Bicycle. Master's thesis, Department of Electrical Engineering Chalmers University of Technology, Gothenburg, Sweden, 2019.

[2] Richard E Klein Karl J Astrom and Anders Lennartsson. Bicycle Dynamics and Control: adapted bicycles for education and research. *IEEE Control Systems*, (25.4):26–47, 2005.

[3] Yu Meng Zhou Andy Ruina Arundathi M. Sharma, Shihao Wang. Towards a maximally-robust self-balancing robotic bicycle without reaction-moment gyroscopes nor reaction wheels. In *Symposium on the Dynamics and Control of Single Track Vehicles*, 2016.

[4] Design School of Innovation and Engineering. MDH students develop autonomous bicycle and an autonomous refuse-collecting robot. `www.mdh.se/mdh-studenter-utvecklar-sjalvkorande-cykel-och-autonom-sophamtningsrobot-1.117867?l=en_UK`, December 2018.

[5] M.F.B. Abdollah G. Omar M.I.F. Rosley N. Tamaldin, H.I.M. Yusof. Design self-balancing bicycle. *Proceedings of Mechanical Engineering Research Day 2017*, pages 160–161, 2017.

[6] Kiattisin Kanjanawanishkul. Lqr and mpc controller design and comparison for a stationary self-balancing bicycle robot with a reaction wheel. *Kybernetika*, 51(1):173 – 191, 2015.

[7] Y. Yongchao X. Kaiyuan C. Kai H. Jiarui, S. Sotirios. Auto-balancing Bicycle. 2015.

[8] A. R. Abdulghany. Generalization of parallel axis theorem for rotational inertia. *American Journal of Physics*, 85(791), 2017.

[9] G. Starr. Calculating Motorcycle Center of Mass. [online], `http://www.me.unm.edu/~starr/moto/cm.pdf`, 2015.

[10] Walter T. Higgins. A Comparison of Complementary and Kalman Filtering. *IEEE Transactions on Aerospace and Electronic Systems*, AES-11:321 – 325, May 1975.

[11] Simon Dan. Kalman Filtering. *Embedded Systems Programming*, 14(6):72–79, 2001.

[12] Greg Welch Bishop and Gary. An Introduction to the Kalman Filter. 2006.

[13] U. S. National Coordination Office. Gps.gov, gps accuracy. https://www.gps.gov/systems/gps/performance/accuracy/.

[14] Roger R Labbe Jr. Kalman and Bayesian Filters in Python. https://drive.google.com/file/d/0By_SW19c1BfhSVFzNHc0SjduNzg/view, 2018.

A

Appendices

A.1 Implementation of a 3D Animation for Simulation Test Visualisation

This Appendix describes the implementation of a 3D animation of the simulation results in order to visualise the behaviour of the bicycle.

The simulation is an important tool to assess the performances of controllers. In our case, two controllers have been implemented: one for balancing and another for path tracking. By running the simulation, the states can be observed, and the controllers can be tuned accordingly. Therefore, the most natural way to visualise the states results is by using a 3D animation of the bicycle.

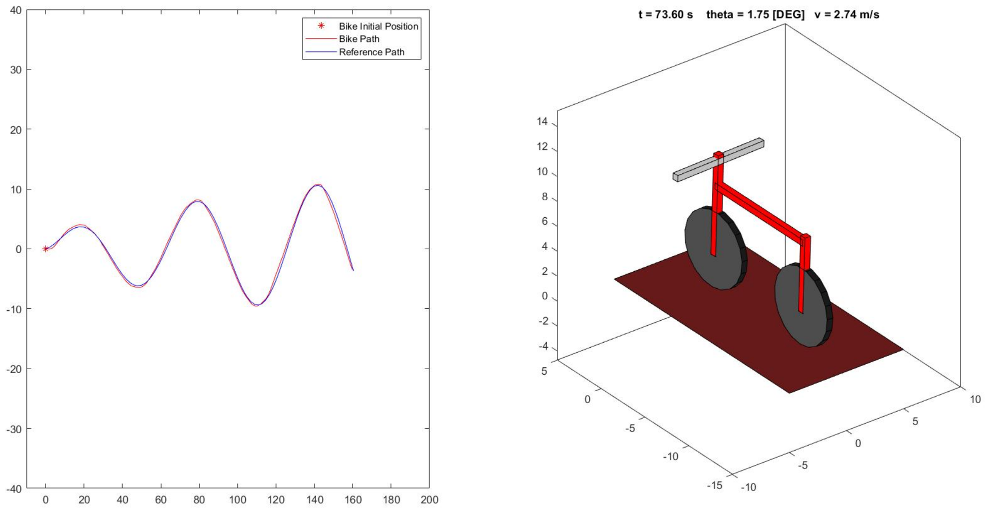

Figure A.1: Display of the 3D simulation test visualisation - Path drawn simultaneously to the bicycle's movements.

The goal of the implementation is to merge the 3D model designed by the students Michael Marne and Ivar Wikenstedt with the MATLAB/Simulink project of the autonomous bicycle.

After simulation, the bicycle animation display presents the 3D model moving according to the varying states of roll, roll rate and steering angle. In the same window, the path is drawn simultaneously (see Figure A.1).

The speed of the animation can be tuned in the MATLAB code by setting the number of points that are simulated and represented in the display at the same time.

A.2 Verification of the Equivalence Between the Volumes' Models in case of a Homogeneous Mass Distribution

This appendix describes the verification of the moment of inertia computation of the two volumes model. To do so, the expressions of the moment of inertia are compared for a single volume model and an homogeneous mass distribution of the two volumes model.The equivalent models used in the verification are described in FigureA.2.

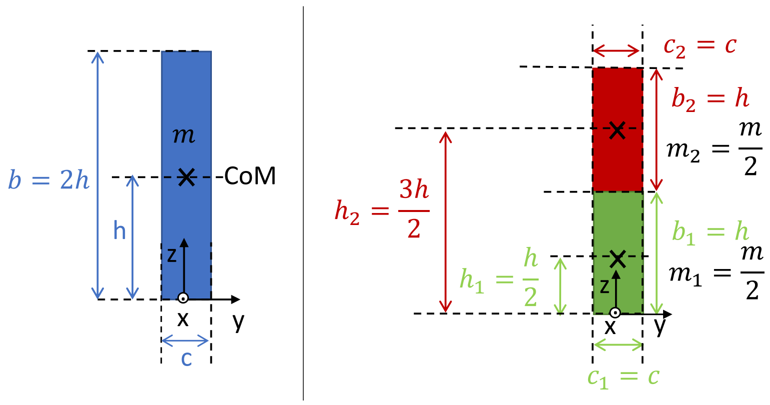

Figure A.2: Equivalent models for one and two volume' models in case of homogeneous mass distribution.

For the single volume model, the moment of inertia is:

$$
\begin{aligned}
J_{xx\,1vol\,verif} &= \frac{m}{12}\left(c^2 + (2h)^2\right) + m\,h^2 \\
&= \frac{mc^2}{12} + \frac{4mh^2}{12} + m\,h^2 \\
&= \frac{m}{12}\left(c^2 + 16h^2\right)
\end{aligned}
\tag{A.1}
$$

For the two volumes model, the moment of inertia becomes:

$$
\begin{aligned}
J_{xx\,2vol\,verif} &= \frac{m/2}{12}\left(c^2+h^2\right)+\frac{m}{2}\left(\frac{h}{2}\right)^2+\frac{m/2}{12}\left(c^2+h^2\right)+\frac{m}{2}\left(\frac{3h}{2}\right)^2 \\
&= \frac{m}{12}\,c^2+\frac{m}{8}\,h^2+\frac{m}{12}\,h^2+9\frac{m}{8}\,h^2 \\
&= \frac{m}{12}\left(c^2+16h^2\right)
\end{aligned}
\tag{A.2}
$$

Comparing the expressions of the Equations A.1 and A.2, the resulting moments of inertia are equal, meaning that the models are equivalent for a homogeneous mass distribution. Thus, the two volumes model is well defined.

In the opposite case, if the mass is not homogeneously distributed (as it is the case for the autonomous bicycle), the two volumes model is able to translate the influence of the mass distribution in the moment of inertia computation.