Modelling and control of an Autonomous Bicycle

Venkatraman Nagaraj*, Klas Lundgren†, Axel Davíð Ingólfsson‡ and Lucas Jansson§
All authors are master students at Systems, Control and Mechatronics,
Chalmers University of technology, Sweden
Email: *vennag@student.chalmers.se, †klaslu@student.chalmers.se, ‡inaxel@student.chalmers.se, §lucasj@student.chalmers.se

***Abstract*—The purpose of this project was to design a controller for a bicycle in order to keep it stable and preferably follow a set trajectory. A model of a bicycle was derived and linearized and an adaptive LQR controller which gains vary with varying velocity was designed to keep the bike stable. The model was based on the well known *Whipple model* which is a simple model that describes the dynamics of a bicycle. A powerful DC motor was used to control the angular velocity of the handlebar. Due to the high torque capacity of the motor, all dynamics affected by the bicycle on the angular acceleration of the handlebar were neglected and angular acceleration was used directly as an input. The controller manages to keep the bike stable and the bike is able to follow turns. However, no complete trajectory planning was implemented.**

***Index Terms*—Autonomous, bicycle, modelling, control, LQR**

I. Introduction

Detecting and tracking bicyclists in traffic is one of the challenges engineers are faced with when developing algorithms for collision avoidance in autonomous vehicles. Options to safely test vehicle's abilities to detect and act in response to threats of accidents involving cyclists are limited. One possible way to perform such tests in a safe environment is to build autonomous bicycles and use them to simulate human cyclists. This paper consists of an in depth description of the mathematical model of the bike's dynamics as well as how a controller is implemented on an actual prototype.

In this paper a model and controller will be implemented to the bicycle with steering of the handlebar as the only input to the system, considering bike forward velocity a constant. The goal is to have a self-stabilizing bicycle that uses steering in order to avoid falling. The dynamics of a bicycle are rather complex and therefore it is difficult to directly implement a controller however, a simple model could still be used to give an understanding of the dynamics, as described by Åström et al. [1]

Modeling and control of a bicycle is a previously explored field. Similar projects have been attempted by A. Sharma et al. [2], Y. Tanaka and T. Murakami [3] as well as well as by A.L. Schwab and N. Appleman [4]. Although the projects are similar in that they regard bicycle dynamics and control, the focuses are not the same for the three aforementioned. However they serve as a basis for what results have been produced previously and can therefore be comparative.

II. Hardware

The bicycle used for the project was a Skeppshult Cyk EL 7. The bike is equipped with an electric DC motor attached to the rear wheel which is designed to to provide torque assistance to the cyclist while pedaling.

Fig. 1. The bicycle used for the project was a Skeppshult Cyk EL 7. The bike in the picture has been equipped with a steering motor, sensors and other electronics.

A. Velocity

The velocity of the bicycle is measured with a hall effect sensor of type Honeywell 103SR13A-1. Five magnets were positioned evenly around the rear wheel frame at known distances and the elapsed time between passes of magnets is measured to calculate the forward velocity of the bike.

Electric bicycles are designed to only activate the wheel motor when a rider is pedaling to provide torque assistance. Being able to independently control the speed of the motor would have been ideal, however, all attempts to do so were unsuccessful.

Nevertheless the bike is equipped with a feature called *walk mode* which can operate the bicycle at a fixed velocity of $6\,\mathrm{km\,h^{-1}}$. It turned out to be possible to take advantage of that feature and programically set the bike's velocity to $6\,\mathrm{kh}$, with the downside of not being able to control the velocity further.

B. Steering

A powerful DC motor of the type *Maxon DCX 32* was attached to the handlebar with a timing belt through a gearbox. The gearbox is of the type *Maxon GPX 32* and has a reduction of 111:1. The resulting nominal torque to the handlebar was $11\,\mathrm{N\,m}$, stall torque $203\,\mathrm{N\,m}$ and the nominal angular velocity $7.8\,\mathrm{s^{-1}}$."

An incremental encoder of type Maxon HEDS 5540 with high resolution is used together with the steering motor to measure the position of the handlebar relative to its calibrated positon. The microcontroller of the type *Beaglebone Black* is equipped with a quadrature decoder that is able to read the shaft position without compromising overall performance of the controller. The result is that the handlebar angle can be measured at a resolution of $3500\,\mathrm{ticks/rad}$.

The angular velocity of the steering is controlled by sending a PWM signal from a real-time processor of type MSP430 Launchpad to a motor controller of type Jaguar MDLBDC24 which is connected to the motor. In order to communicate with the Launchpad, characters are sent from the BeagleBone through serial connection with each character corresponding to a predefined PWM value. Since serial communication is restricted to single characters, it was decided to use the entire ASCII set (which consists of 63 characters) where each character corresponds to a PWM value. As a result, there are a limited number of angular velocities available to the controller. Tests were then performed where the resulting angular velocity was measured for each PWM value and the measurements used to construct a linear function to convert desired angular velocities to PWM values. The result is a discrete angular velocity controller that can operate at velocities between $-7.9\,\mathrm{m\,s^{-1}}$ and $7.9\,\mathrm{m\,s^{-1}}$ with a resolution of $4\,\mathrm{steps/rad/s}$.

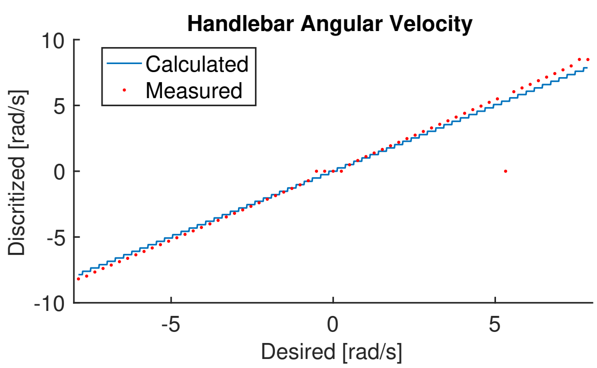

Fig. 2. The handlebar angular acceleration was discritized with a resolution of $4\,\mathrm{steps/rad/s}$ between $-7.9\,\mathrm{m\,s^{-1}}$ and $7.9\,\mathrm{m\,s^{-1}}$. The red dots are measured samples for different input angular velocities. This figure reveals a problem of controlling the angular velocity when it's close to zero.

C. Orientation estimation

The dynamics of a bicycle that are associated with stability are dependant on the roll angle and roll angular velocity of the bike. These states are estimated with the usage of an IMU (Inertial Measurement Unit) of the type *FXAS21002C*. IMUs are equipped with an accelerometer, gyroscope and a magnetometer which can be used separately or fused together in order to estimate orientation with respect to different degrees of freedom. The angular velocity can reliably be measured with the gyroscope since the gyroscope measures acceleration changes very accurately. The absolute roll angle is measured by passing measurements from the accelerometer and the gyroscope through a AR(1) type complementary filter. The AR(1) filter is used to balance between integrated measurements from the gyroscope and absolute acceleration measurements from the accelerometer. Using these measurements separately without a complementary filter would lead to a accumulated drift due to integration error in the case of the gyroscope or noisy estimations in the case of the accelerometer as acceleromters do not perform well when the gravitational vector is changing rapidly.

III. Modeling

Bicycles are highly dynamic and non-holonomic systems which makes the modeling and control more interesting to model less redundant systems. With the aim of balancing and steering, the bicycle is modelled considering the lateral forces responsible.

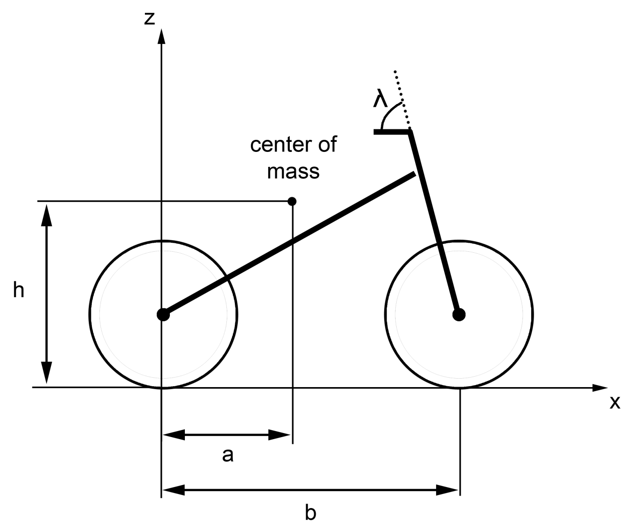

Fig. 3. A side view of the bike, showing parameters used in the model. The origin of the coordinate system is based at contact point where the rear wheel touches the ground.

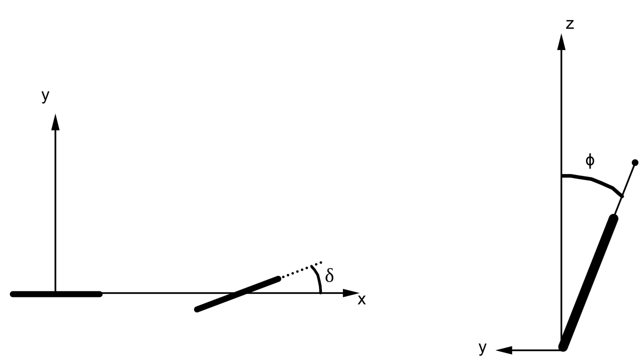

Fig. 4. Illustration of the angles used to describe the model. The left image is the bike seen form above, and the right image is seen from behind. The origin of the coordinate system is based at contact point where the rear wheel touches the ground.

The model of the bicycle used is the Whipple bicycle model [1] which describes the motion by use of four rigid bodies: a rear wheel R, a rear frame B, a front frame H consisting of the handlebar and fork assembly and a front wheel F [5]. The assumptions made are as follows;

- Head angle λ is $\pi/2$ and trail is zero.
- The model is made up of four rigid bodies.
- The bodies are connected to each other by zero friction joints.
- The bicycle is laterally symmetric, wheels have knife edges and no longitudinal slip while rolling.
- The bicycle moves on flat ground.

A bicycle has a self-stabilizing property as described by Åström [1]. However the motor mounted to the handlebar restricts the free steering, which is required for this property, and thus this property can be disregarded for this case.

A. Nonlinear Bicycle Model

A simple second order model (1) can be obtained from balances of forces on the system equation, with steering acceleration as control input.

$$J\frac{d^2\phi}{dt^2} - mgh\sin\phi = \frac{Dv_0}{b}\frac{d\delta}{dt} + \frac{mv_0^2h\delta}{b} \tag{1}$$

The terms in the right hand side of the equation(1) are torque generated while steering, due to inertial forces and the centrifugal forces respectively, in the left hand side $mgh\sin\phi$ is the torque due to gravity.
where m is the total mass of the bicycle, v_0 is the forward velocity, g is the gravity constant, h and a are the vertical and horizontal distances from the ground and the center of the rear wheel to the center of mass respectively and b is the distance between the rear and front wheel points of contact on ground.

Now, approximating moment of inertia and inertia product as, $J \approx mh^2$ and $D \approx mah$ in (2), we get

$$\ddot{\phi} = \frac{g}{h}sin\phi - \frac{av_0}{hb}\dot{\delta} - \frac{v_0^2}{h^2b}\delta \tag{2}$$

B. Second Order Linearized model

Linearizing this second order nonlinear model about some small tilt angles ϕ, results in(3).

$$\ddot{\phi} = \frac{g}{h}\phi - \frac{av_0}{hb}\dot{\delta} - \frac{v_0^2}{h^2b}\delta \tag{3}$$

The state vector used in order to describe the motions of the bikes is $\mathbf{x} = [\phi, \delta, \dot{\phi}, \dot{\delta}]^T$. Where ϕ is the roll angle, δ is the angle of the handlebar followed by their respective angular velocities. The dynamics are then described as a state space representation, with velocity v as a parameter.

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{g}{h} & -\frac{v^2}{hb} & 0 & -\frac{av}{hb} \\ 0 & 0 & 0 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}\mathbf{u}. \tag{4}$$

C. Linear Fourth-Order Model

The simple second order model has static momentum balance for front fork assembly and less information about mass distribution. The more detailed model which describes the mass distribution and geometry, is a fourth order model (5).

$$M\begin{bmatrix}\ddot{\phi} \\ \ddot{\delta}\end{bmatrix} = \begin{bmatrix}0 \\ T_\delta\end{bmatrix} - vC\begin{bmatrix}\dot{\phi} \\ \dot{\delta}\end{bmatrix} - [gK_0 + v^2K_2]\begin{bmatrix}\phi \\ \delta\end{bmatrix}, \tag{5}$$

where M is the symmetric mass matrix, C is the damping matrix, K_0,K_2 is the stiffness matrix, T_δ is the input steering torque,ϕ is the roll angle, δ is the angle of the handlebar, ϕ is the roll angular velocity, δ is the angular velocity of the handlebar.
This model can be used knowing the mass and inertia properties of the complete bicycle. The simple model is used here to control stability and turning.

IV. Controller

The linearized model described by equation 3 represents the nonlinear model (equation 2) quite well when the angles ϕ and δ are small. However, the nonliearities are more significant with respect to varying velocity which makes it unfeasible to use a linear controller with varying velocity. Since the velocity is a varying parameter but not a state which is affected by the system, it was decided to control the system with an adaptive LQR controller which gains vary with varying velocity.

A. LQR

The task is to control a bike with an LQR controller which basically only is a state feedback. An observer is not needed because all the states are observable. The advantage with LQR is that it takes up less computational power, than PID for example, because it is only a matrix multiplication by a gain that has been pre-calculated by minimizing a cost function with respect to the input, whilst a PID compute both derivative and integral. The gain is calculated with the state matrices from the linearized model. The weight matrix Q is chosen so that the input and state outputs are kept within the limits.

The disadvantage with LQR is that it is less robust than PID. The gain is calculated for a linearized model with fixed parameters which make the controller valid for just that object. This means that if this controller would be applied on a different bike it would lose its efficiency and possibly fail even if the bike is similar. With a PID controller, this method of failure would be less likely. The biggest challenge is if the system has a parameter which is changing with respect to time then the gain of the LQR also needs to be a function of time.

The cost function which is supposed to be minimized with respect to the input is

$$J = \frac{1}{2}\int_0^\infty \left(x^T(t)Qx(t) + u^T(t)Ru(t)\right) \mathrm{dt} \tag{6}$$

where Q is the weight matrix which penalizes the states and R is the input weight. The Q matrix has to penalize for steering angle a lot because the angle is limited to $\pm 30°$. The input is angular acceleration for the handlebar and it is important to limit that otherwise the motor takes damage.

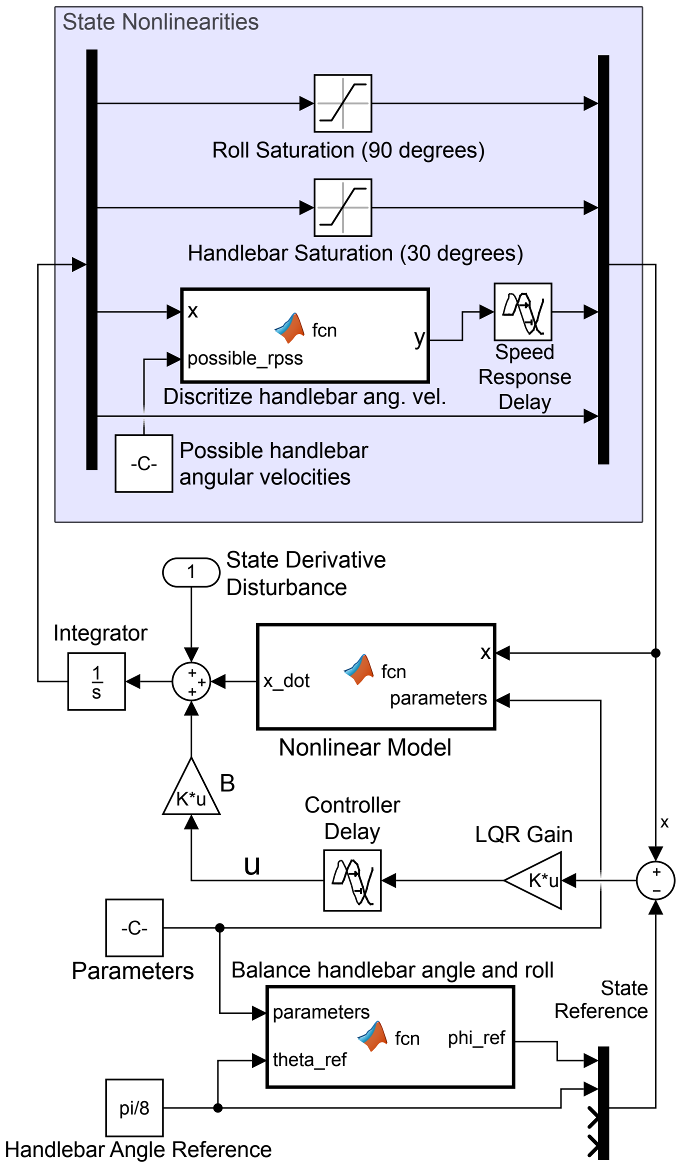

Fig. 5. Block diagram of the system with the implemented LQR feedback. This controlled model does not take varying velocity into account. The additive state derivative disturbance was used to test the controller when the bike was exposed to a torque disturbance around the axis of ϕ.

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad R = 1 \tag{7}$$

The gain is then found by using `lqrd(A,B,Q,R,Ts)` which is function that gives you a discrete controller from a continuous system where Ts is sampling time.

The velocity will mostly stay at a fixed speed, but variations can occur in a rough terrain and it will take a while to reach the desired speed from initially standing still. Having the velocity as a state will add unnecessary difficulties which will complicate the linearization. The easiest way to have a varying velocity is to keep it as a parameter but have a change gain depending on velocity $K(v)$.

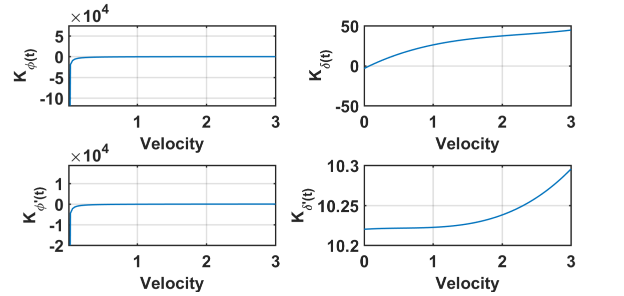

Fig. 6. The LQR gain described for the four states as a function of velocity from $v = 0$ m/s to $v = 3$ m/s.

The system has one set point reference and four states which gives a K with size 1x4. To find the adaptive gain the relationship between velocity and each one of the four scalars in K must be found and that is calculated by evaluating K for a range of different velocities. The relation is found by using the curve fitting tool box in MatLab which takes a set of data and fits a given type of function to the set. The type is given by the user ex. polynomial, power and sum of sines. It was found that the gain acting on the angles ($K(1)$ and $K(3)$) varies with respect to the function $a/(v^b) + c$ and entries 3 and 4 varies according to a third degree polynomial.The gain $K(v)$ is shown in figure 6.

$$K(v) = \frac{a}{x^b} + c \tag{8}$$

B. Algebraic relationships between state references.

When the bicycle heads straight forward the reference roll angle should be 0 in order for the bike to remain stable since the gravitational vector will be pointing downwards with respect to the bike. This is however not the case when the bicycle enters a turn since it will be exposed to a centripetal acceleration. In order for the bike to remain stable, the net acceleration acting perpendicular to the side of the bike should converge to zero. In order for this to hold, the relationship between the reference handlebar angle and the reference roll angle should be,

$$\tan(\phi_{ref}) = \frac{v^2}{gl}\tan(\delta_{ref}) \tag{9}$$

This conversion was added to the system modelled in Simulink and can be seen in figure 5. By setting a state reference to the handlebar angular acceleration, it is possible to steer the bicycle along a turn.

V. Simulation Results

The implemented LQR was then tested by adding an input to steering angle reference to 22.5° from standing still with the velocity set to $v = 1.6$ m/s. The response from the simulation is shown in figure 7. The rise time of the system is 0.88 s.

A. Robustness

To check the robustness of the controller simulations of the nonlinear system is executed with some errors of the parameters. The parameters which could affect the performance of

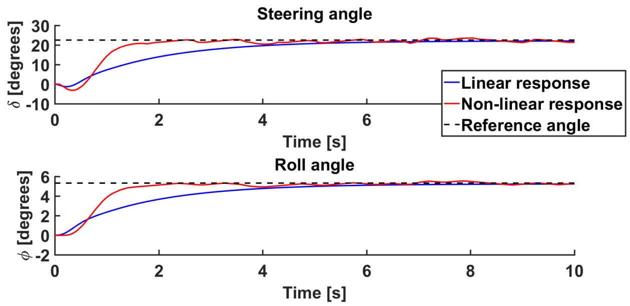

Fig. 7. Handlebar Angle reference is $22.5°$, with the center of mass of both the controller and model set to $(x, z) = (0.27, 0.53)$ m.

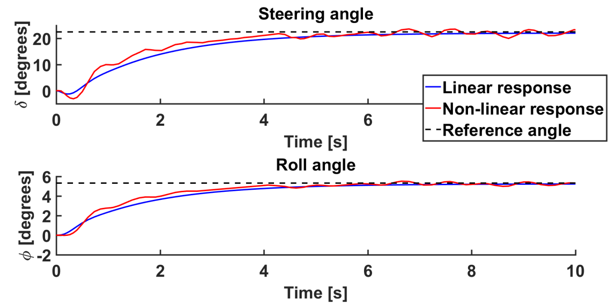

Fig. 9. Handlebar angle reference is $22.5°$. The center of mass of the model is $(x, z) = (0.42, 0.53)$ m and for the controller it is set to $(x, z) = (0.27, 0.53)$ m.

the controller on the actual bike were then changed for the dynamics model whilst still using the developed controller. Center of mass was then altered by changing the the parameters a and h. Then the same test as specified previous in the section for these different scenarios, the results simulations are shown below in figures V-A to figure 11. This is done to verify that the controller works even if the bike parameters differ from the simulated model.

To further analyze this firstly tests were made by independently varying the a parameter, which is the position of the center of mass on the x-axis. Initially This change was done to the system dynamics whilst keeping the controller for the initial parameters. It was found that a rise time of $t_r = 1.31$ s was still achieved for a bike which has the parameter $a = 0.32$ m whilst the controller was made with the parameter $a = 0.27$, as shown in figure V-A. Further increasing this parameter increases the rise time above $t_r > 1.5$ s. For $a = 0.34$ the rise time increases to $t_r = 2.22$ s.

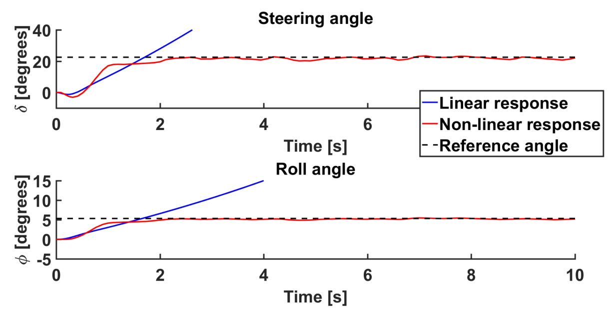

Fig. 10. Handlebar angle reference is $22.5°$. The center of mass of the model is $(x, z) = (0.27, 0.43)$ m and for the controller it is set to $(x, z) = (0.27, 0.53)$ m.

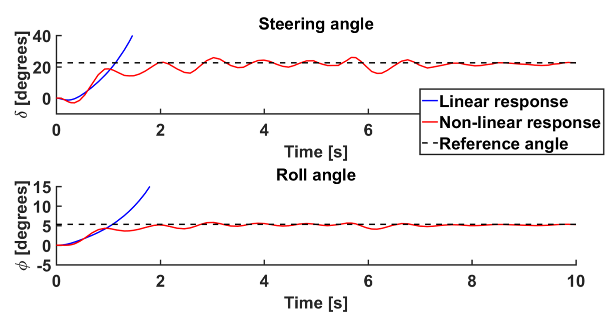

Fig. 11. Handlebar angle reference is $22.5°$. The center of mass of the model is $(x, z) = (0.27, 0.33)$ m and for the controller it is set to $(x, z) = (0.27, 0.53)$ m.

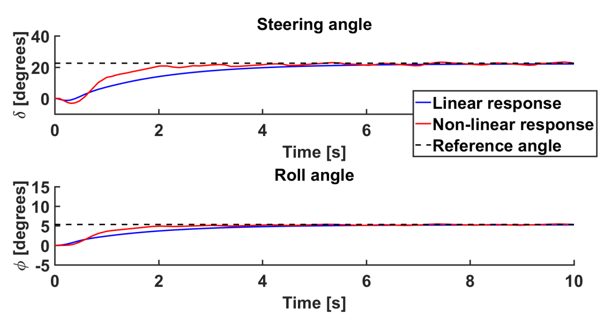

Fig. 8. Handlebar angle reference is $22.5°$. The center of mass of the model is $(x, z) = (0.32, 0.53)$ m and for the controller it is set to $(x, z) = (0.27, 0.53)$ m.

Similar tests were done with varying the h parameter. Increasing the parameter would in reality reflect an inverted pendulum with increased length. This results in a slower system and although the height is drastically increased the rise time is still kept below 1.5 s. Decreasing the height of the center of math is however a more crucial scenario. Setting $h = 0.43$ results in a rise time of $t_r = 1.47$ s, which is still acceptable with keeping $t_r < 1.5$ s.

VI. Conclusion

A model was derived describing local orientation of the bike with respect to the roll angle ϕ and steering angle δ. Simulations indicate that the controller can control the front wheel in such manner that it is possible to keep the bike stable for varying velocities between $0\,\mathrm{m\,s^{-1}}$ and $3\,\mathrm{m\,s^{-1}}$. This was not successfully tested on the an actual bicycle.

The controller was designed for a bicycle with a fixed height and length. Simulations show a rise time of 0.88 s. The controller also works for a similar bike where the center of mass can vary with ± 5 cm whilst maintaining a rise time of less than 1.5 s.

Acknowledgment

The authors would like to thank Boaz Ash for assisting in the project as well as building the prototype and thanks to Jonas Sjöberg for the continued support throughout the project.

References

[1] 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.

[2] Arundathi M Sharma et al. "Towards a maximally-robust self-balancing bicycle without reaction-moment gyroscopes or reaction wheels". In: (2016).

[3] Yasuhito Tanaka and Toshiyuki Murakami. "Self sustaining bicycle robot with steering controller". In: *Advanced Motion Control, 2004. AMC'04. The 8th IEEE International Workshop on*. IEEE. 2004, pp. 193–197.

[4] AL Schwab and Nick Appleman. "Dynamics and Control of a Steer-by-Wire Bicycle". In: (2013).

[5] Jaap P Meijaard et al. "Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review". In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 463. 2084. The Royal Society. 2007, pp. 1955–1982.