A Loop Shaping Method for Stabilising a Riderless Bicycle*

Tom Andersson, Niklas Persson, Anas Fattouh, and Martin C. Ekström

Abstract— **Several control methods have been proposed to stabilise riderless bicycles but they do not have sufficient simplicity for practical applications. This paper proposes a practical approach to model an instrumented bicycle as a combination of connected systems. Using this model, a PID controller is designed by a loop shaping method to stabilise the instrumented riderless bicycle. The initial results show that the bicycle can be stabilised when running on a roller. The work presented in this paper shows that it is possible to self stabilise a riderless bicycle using cascade PI/PID controllers.**

I. Introduction

A bicycle is a very well-known and popular means of transportation. The bicycle is often configured to have two inline wheels which make the system inherently unstable and difficult to control. The rider must train how to balance the bicycle. Balancing a bicycle is based on the simple concept, steering into the fall direction, but it is hard to be applied automatically as many features of the bicycle should be considered such as the trail, the gyroscopic torque, the mass distribution, and the forward velocity.

Two common models which represent, to some extent, the dynamics of a bicycle are the Whipple model and the point-mass model. The Whipple model, developed by Francis John Welsh Whipple in 1899, is the first analytic model that in a correct way describes the dynamics of a bicycle [1]. Linearised equations were derived from the Whipple model by Meijaard et al. [2] and they have been used in many implementations regarding control of a riderless bicycle, such as in the work by Baquero-Suarez et al. and Shafiekhani et al. [3], [4]. A less complex dynamic model of a bicycle is the point mass model, as described by Liembeer and Sharp [5], where a set of assumptions allows the bicycle to be modelled as an inverted pendulum, with its mass as a point mass.

Several control methods were proposed to stabilise the riderless bicycle using the point mass model [6], [7]. However, the proportional integral derivative (PID) is still attractive from an industrial point of view [8] and more investigation is needed to come up with an effective tuning method for PIDs.

In this paper, the mass-point model is considered to model the riderless bicycle. A robust PID controller design method is proposed to come up with the simplifications on the considered model. It is then used to stabilise an instrumented bicycle.

* Research supported by VINNOVA
T. Andersson, N. Persson, A. Fattouh, and M.C. Ekström are with Mälardalen University, Västerås, Sweden `name.surname@mdh.se`
The first two authors contributed to the work equally.
978-1-7281-3605-9/19/$31.00 ©2019 IEEE

The remaining of this paper is structured as follows. Section II gives a brief survey of related works. The simulation model of the bicycle is explained in Section III. The proposed robust PID designed method is developed in Section IV. Section V is devoted to the application of the robust PID control design method on an instrumented riderless bicycle and the obtained results. Finally, concluding remarks and future works are given in Section VI.

II. Related Works

Different types of controllers have been proposed in the literature to control and balance riderless bicycles. From the more traditional ones, such as the Proportional-Integral-Derivative (PID) controllers, to more complex ones which rely on various Artificial Intelligence (AI) controllers.

Tanaka and Murakami [9] presented one of the first riderless bicycles which managed to keep its balance while riding on a roller. To control the bicycles steering axis, and by extension the balance, a PD controller was utilised along with disturbance observers. From the results, it is possible to see how the steering angle follows the lean angle which makes the bicycle balanced. A PD controller was also implemented in the work by Suebsomran [10], where a bicycle, equipped with a reaction wheel was kept stable. The controller regulates the angle of the reaction wheel to produce a necessary torque for the bicycle to be able to balance. The PD controller manages to stabilise the bicycle, however, only in a simulated environment. In the work by Wang *et al*. a cascade controller for balance and directional control of a bicycle-type two-wheeled vehicle is presented [11]. In the inner loop, a PD controller was used for balancing the vehicle by sensing the lean angle and output a steering torque to the plant. The outer loop, which composed of the directional control, also relied on a PD controller, but by sensing the yaw angle it outputs a reference lean angle which was fed to the inner loop. Experiments were made in simulation and on a real bicycle-type two-wheeled vehicle. Both the balance and the directional controller showed promising results. However, the platform used for testing was small and was some sort of hybrid between a kid bicycle model and small scooter. A regular sized bicycle have a greater height and mass compared to small bicycle, and also the center of gravity is generally at a greater distance from the ground which makes the regular sized bicycles more sensitive to disturbance and harder to control. Because of the size and the structure of the platform, the presented results cannot be generalised to a regular sized bicycle.

Since many systems, including bicycles, are simplified when modelled, there are some uncertainties present. A robust controller is specially designed to tune a system to have the desired behaviour, even with some uncertainties in the models [12]. Many of the proposed controllers in the related works are only evaluated in simulation and often experiments conducted on a real bicycle is missing. For example, in the work by Chen and Dao [13], a Sliding-Mode Controller (SMC) was proposed to track the bicycles roll angle. The benefits of using SMC to control a bicycle is that the uncertainties in the Whipple model's velocity can be compensated. The results showed that the bicycle was stable in 15km/h in a simulation environment.

Anjumol and Jisha [14] proposed an LQR controller to control a second-degree bicycle model. The results obtained from the controller was compared with a posture controlled proposed by Tanaka and Murakami [15]. The result from the comparison shows that an LQR controller performs better. However, the posture controller uses a PD controller with a disturbance observer which was better for disturbance rejection. An adaptive self-tuning regulator is proposed by Al-Buraiki and Ferik [16]. The controller uses an estimation stage and a construction stage for the input signal, the first stage uses a weighted recursive least squares approach to estimate the models state-space, which is used in the second stage to construct the input signal. Additionally, in the second stage, an LQR controller is adapted for the on-line estimation. The proposed controller was only evaluated in simulation, where it successfully managed to balance the bicycle.

An AI-controller is used in the work by Sharma, where a fuzzy controller is presented to balance a bicycle [6]. The developed controller takes two inputs, the lean angle, and the steering angle, and outputs a correction lean. Thus, the controller relies on direct regulation of the lean angle which would require a reaction wheel, an inverted pendulum or something similar. In this paper, regulation of the steering angle and the rear wheel speed is used to keep the bicycle stable. Sharmas' controller manages to stabilise the bicycle in a simulated environment but was never used in any real-life experiments.

Real-life experiments are conducted in the work of Shafiekhani et al. [4] where an adaptive critic-based neuro-fuzzy controller was developed. A comparison was made between the neuro-fuzzy controller and a Fuzzy Inference System (FIS) controller, and it was concluded that the neuro-fuzzy offers more accurate performance in term of tracking the lean angle in both simulation and reality. Additionally, Abdolmalaki [17] presented a control system that relies on a FIS in combination with a PID controller to balance a bicycle. Results from experiments showed that the bicycle was able to balance along with a straight line, however with oscillations of $\pm 5°$. The oscillations were also present when a sinusoidal trajectory was followed, despite this, the proposed control was still able to keep the bicycle from falling over in real-life experiments.

In 2018, Baquero-Suarez et al. [3] presented promising results where a regular sized bicycle, of a male model, equipped with sensors and actuators, manage to balance itself. It was able to follow a path using steering torque and forward velocity as the only control outputs. The results showed that the proposed system can balance even under small accelerations, however, the control structure is complex.

III. Modelling of the Riderless Bicycle

The riderless bicycle consists of three main parts, the bicycle, the steering unit, and the moving unit. The point-mass model is used in this paper to model the bicycle where a direct relationship between the lean angle and the steering angle are described. The steering motor is internally controlled to get the desired second-order system dynamic. The rear wheel motor is controlled such that the bicycle can move in a constant speed and it is not considered in this paper.

A. The Point-Mass Model

The point-mass model is one of the more basic analytic bicycle models, it's a simple second-order linear model with a set of simplifications. The model assumes that both the front and rear wheel along with the front frame is mass-less, giving them inertia of zero. However, their masses are lumped together forming a point-mass, hence the models' name. To simplify the bicycle model further, it is assumed that the forward speed is constant and the heading angle λ and the trail distance are zero. Consider the bicycle shown in Fig. 1, with x-axis in the forward direction of the bicycle and the z-axis in the vertical direction. The two points, P1 and P2 are the contact points between the ground and the rear wheel and front wheel respectively. P3 is the point where the steering axis and the horizontal plane intersects with each other. The parameters a and h describe the distance from the rear wheel to the centre of gravity (CoG) in the x- and z-axis. λ is the head angle, c is the trail distance, and the b is the wheel base [18].

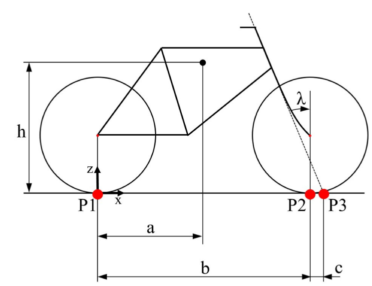

Fig. 1. The a and h corresponds to the position of the CoG. The wheelbase is given by b, λ describes the head angle, and c is the trail.

Following the procedure described in [18], we get the following transfer function from steer angle δ to lean angle φ

$$
\begin{aligned}
G_{\varphi\delta}(s) =& \frac{v(Ds + mVh)}{b(Js^2 - mgh)} \\
=& \frac{vD}{bJ} \frac{s + \frac{mVh}{D}}{s^2 - \frac{mgh}{J}} \approx \frac{av}{bh} \frac{s + \frac{v}{a}}{s^2 - \frac{g}{h}}. \qquad (1)
\end{aligned}
$$

As can be seen from the transfer function, it will behave differently for different velocities. In this paper, a constant forward velocity of 14km/h and the bicycles physical parameters presented in Table I are considered.

TABLE I

INSTRUMENTED BICYCLE PARAMETERS.

Design parameters			
Parameter	**Symbol**	**Unit**	**Value**
CoG with respect to O (x)	a	$[m]$	0.486
CoG with respect to O (z)	h	$[m]$	0.519
Gravity	g	$[m/s^2]$	9.820
Wheel base	b	$[m]$	1.080

B. Steering Response Matching

The internal structure of the position controller is composed of one PD steering angle controller followed by a speed and a current PI controller as shown in Fig. 7. Instead of modelling the three closed loop controllers, handlebar mass, friction and the motor characteristics, a step matching response method is applied where the recorded step response from the instrumented bicycle is matched with a timed delayed second-degree transfer function. To record the step response from the instrumented bicycle, it is held in an upright position with the wheels on the ground and a step of 3 degrees are commanded. The input to the transfer function is the desired steering angle and the output is the actual steering angle.

$$P(s) = e^{-d \cdot s} \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \quad (2)$$

Matching the response in Fig. 2 gives a transfer function with damping factor $\zeta = 0.6$, $\omega_n = 33.9$, and the time delay $d = 0.015$.

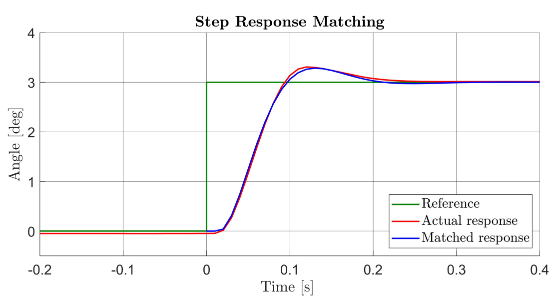

Fig. 2. Recorded response along with the matched transfer function.

By coupling the system (2) and (1), i.e expanding the point-mass model with the dynamics captured from the steering system, the instrumental bicycle is modelled. Converted into discrete time, using zero-order hold as the discretisation method, with a sampling time of 0.01 seconds, the complete model is given by the following transfer function. The input to the system is the desired steering angle and the output is the current lean angle.

$$P(z) = z^{-2} \times \frac{0.000461z^3 + 0.00198z^2 - 0.00186z - 0.000324}{z^4 - 3.574z^3 + 4.813z^2 - 2.905z + 0.6658} \quad (3)$$

IV. Balance Controller Design

In this section, the design problem of a PID controller is formulated as an optimisation problem that can be solved using an appropriate algorithm. The goal of the balance controller is to track a lean angle, by outputting a desired steering angle to the steering position controller. Consider the feedback control system in Fig. 3, where $G(z)$ is the plant, $K(z)$ is the controller, $r(t)$ is the reference signal, $y(t)$ is the system output signal, $u(t)$ is the control signal, $d(t)$ is the disturbance signal, and $n(t)$ is the noise signal.

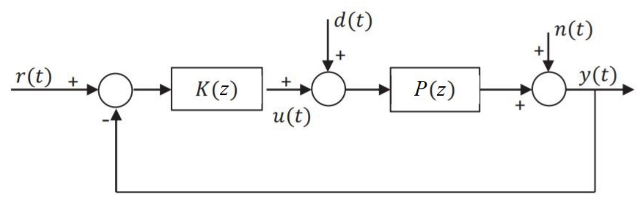

Fig. 3. A standard feedback control system.

The objectives of the feedback control loop are to ensure the stability of the closed loop system, good tracking performance, robustness against the plant uncertainties, and rejection of the disturbances affecting the system. These objectives can be formulated as constrained on the frequency response of the loop transfer function as follows [19]:

- The open loop frequency response should cross the 0 dB once with a constraint on the phase margin (to ensure the stability and the performance of the closed loop system).
- The gain of the open loop frequency response should be high below the desired bandwidth (to ensure the rejection of the disturbances).
- The gain of the open loop frequency response should be low above the desired bandwidth (to ensure the robustness against the plant uncertainties).

A controller that meets the above constraints can be designed by optimising the following objective function [20]:

$$\begin{aligned} J =& \omega_1(\omega_b - \omega_t)^2 \\ &+ \omega_2 \sum_{\omega > \omega_b} 20log|K(j\omega)P(j\omega)| \\ &- \omega_3 \sum_{\omega <= \omega_b} 20log|K(j\omega)P(j\omega)| \end{aligned} \quad (4)$$

where ω_t is the target bandwidth, ω_i, $i = 1, 2, 3$, are weights, and ω_b is the bandwidth of the loop transfer function defined by:

$$\omega_b = \inf_{\omega} |K(j\omega)P(j\omega)| \leq 1 \quad (5)$$

The weights ω_i, $i = 1, 2, 3$ have to be selected properly in order to approximately satisfy the design requirements. The following weights are suggested by [20]

$$\omega_1 = \frac{1}{\omega_t^2}, \omega_2 = \omega_3 = \frac{1}{2000} \tag{6}$$

The above optimisation problem is solved for the bicycle model in eq. 3 using *fmincon*, which is a nonlinear programming solver in MATLAB. The solution yields the following PID controller:

$$K(z) = K_P(1 + K_I T_s \frac{1}{z-1} + \frac{K_D}{T_s}\frac{(z-1)}{z}) \tag{7}$$

where $K_P = 2.5117$, $K_P = 1.5431$, $K_D = 0.075$, and $T_s = 0.01s$.

Fig. 4 shows the sensitivity function of the designed system. It shows that the largest value of the sensitivity function is 2.02 which is in the range of the recommended values and the system will have a good rejection of the disturbance in lean angle.

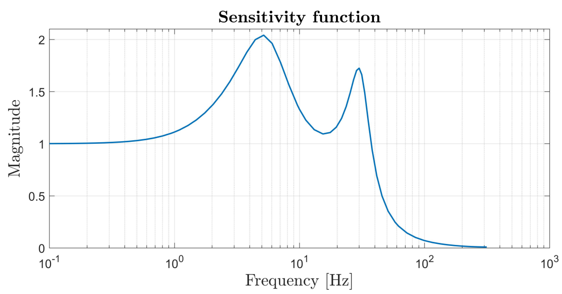

Fig. 4. Sensitivity function of the designed system.

V. Simulation Results

The complete system is modelled in Simulink along with the noise of the lean angle sensor which has a measured variance of 0.0001 and a standard deviation of 0.01. In the Simulink model, the balance controller has a sampling time of 100 Hz, the transfer function which represents the steering system executes in 600 Hz and the point-mass model is implemented in continuous time. The result of running the complete Simulink model with a disturbance on the lean angle is shown in Fig. 5. The simulation shows that the bicycle keeps balancing after disturbing the lean angle.

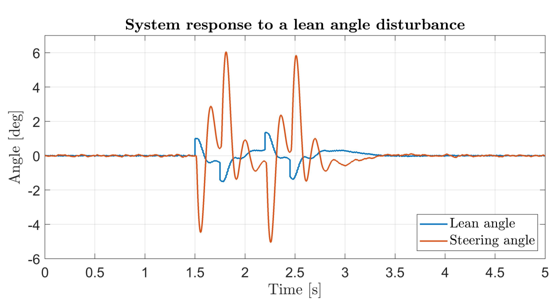

Fig. 5. The steering and lean angles from simulation.

VI. Application to the Instrumented Bicycle

The bicycle is a modified electrical bicycle of a regular sized male model with a motor in the rear wheel. The bicycle is equipped with a brushed DC motor for controlling the steering angle which is measured with an encoder. To control the steering motor a Junus motor controller is utilised. A VN-100 is used for measuring the lean angle of the bicycle and is mounted underneath the bottom bracket shell. To be able to send remote commands to the bicycle, a receiver is mounted on the bicycle. As the main processing unit a National Instruments roboRIO is utilised and the software is written using LabVIEW, the instrumented bicycle is shown in Fig. 6.

Fig. 6. The instrumented bicycle with its rear wheel motor inside the green square. The blue box indicates the position of the VN-100, used for measuring the lean angle. Inside the yellow box is the electrical speed controller. National Instruments roboRIO is mounted on the bicycle and is highlighted by the purple box. To be able to send remote commands to the bicycle, the receiver inside the red box is utilised. The motor used to control the steering angle is highlighted by the orange rectangle.

A. Control structure

The control structure on the roboRIO is devised of an inner and an outer loop forming a cascade controller as shown in Fig. 7

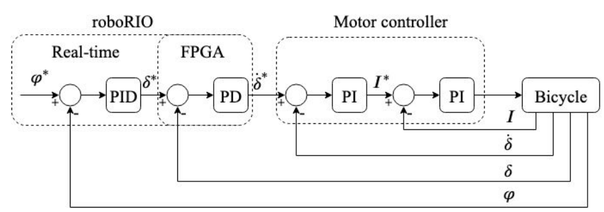

Fig. 7. The complete control structure where $\varphi^*, \delta^*, \dot{\delta}^*, I^*$ represents the desired values and $\varphi, \delta, \dot{\delta}, I$ are the measured values from the bicycle. The balancing controller and steering angle controller are implemented on the roboRIO while the velocity and current controller reside on the motor controller.

The steering angle controller tracks the position of the handlebar and the outer loop tracks the desired lean angle of the bicycle. The steering angle controller uses a PD controller running on the roboRIO FPGA target with a loop speed of 600Hz, while the outer balancing PID controller is implemented on the real-time OS running with a loop frequency of 100Hz. The output of the steering angle controller is a

PWM signal which is fed to the motor controller. Inside the motor controller, the PWM signal is mapped to the desired motor velocity and compared to the current motor velocity. The error is inputted to a PI controller and the resulting output is forwarded as the desired current to the last control loop which regulates the voltage going to the motor by using another PI controller.

Using software which accompanied the motor controller, the current controller is auto-tuned. The velocity controller is manually tuned using the same software and their respective control gains can be seen in Table II along with their saturation limits. The two controllers which reside on the roboRIO use the values shown in Table III.

TABLE II
THE GAINS USED FOR THE TWO PI CONTROLLERS WHICH RESILE ON THE MOTOR CONTROLLER.

Junus controllers			
Velocity controller		**Current controller**	
C_p	70	V_p	700
C_i	81	V_i	200
Limit	4.12 A	Limit	7000 RPM
Speed	4 kHz	Speed	20 kHz

TABLE III
CONTROLLER GAINS FOR THE PD CONTROLLER IMPLEMENTED ON THE FPGA TARGET OF THE ROBORIO AND THE PID CONTROLLER WHICH EXECUTES ON THE REAL-TIME TARGET.

roboRIO controllers		
	Steering angle controller	**Balance controller**
$\mathbf{K_p}$	0.10	2.5117
$\mathbf{K_i}$	0	1.5431
$\mathbf{K_d}$	0.04	0.0750
Output range	±50	±45
Speed	600 Hz	100 Hz
Filter coefficient	0.80	-

To control the forward velocity of the bicycle, a PI controller is implemented on the FPGA target of the roboRIO. To measure the speed of the rear wheel, 12 magnets are mounted on the rear wheel, and a Hall sensor measures the pulses from the magnets and converts it to speed. The output from the PI controller is fed to the rear wheel motor through an electrical speed controller.

B. Experimental setups

To evaluate the proposed controller, the bicycle is placed on a bicycle roller with the front wheel pointing forward and is kept in an upright position by a human. Riding a bicycle on a roller is similar to riding a bicycle in an outdoor environment [21].

An experiment begins with a small calibration phase where the steering angle is initialised to zero and the IMU is powered on. During the calibration phase, it is important that the front wheel is pointing forward. Using a small offset, the IMU angle is also initialised to approximately zero. Next, the control loop is deployed and the rear wheel is powered on and accelerates to approximately 14km/h. When the bicycle rides at a nearly constant velocity the human is releasing the bicycle and the control loop is fully in-charge of keeping the bicycle stable, as shown in Fig. 8.

Fig. 8. Self stabilising bicycle on a roller running with a forward speed of 14 km/h. The width of the rollers is 37 cm.

Two experiments are conducted, in the first one the bicycle is simply riding on the bicycle roller and the signals are logged until a human has to interfere with the bicycle. To further evaluate the controller and its robustness, lean angle disturbance is injected in the lean angle measurements. The amplitude is 1 degree and its present for 0.25 seconds, and the disturbance is injected two times. Another possibility is to induce disturbance manually by introducing some lateral forces on the bicycle. However, by inducing the disturbance directly in the measurements, the experiment can easily be reproduced if necessary.

C. Results

In Fig. 9 and Fig. 10 the result from the bicycle experiment is presented. The signals are only logged while the bicycle self-stabilising, as soon as a human interacts with the bicycle the logging is turned off. Fig. 9 plots the lean and steering angle of the bicycle as well as the desired steering angle which is outputted from the balance controller. The forward speed of the bicycle is given in Fig. 10. In the experiment, the gains of the PID controller are taken from the tuning process done in simulation without any modifications and can be found in Table III. The results from the disturbance rejection experiment are shown in Fig. 11.

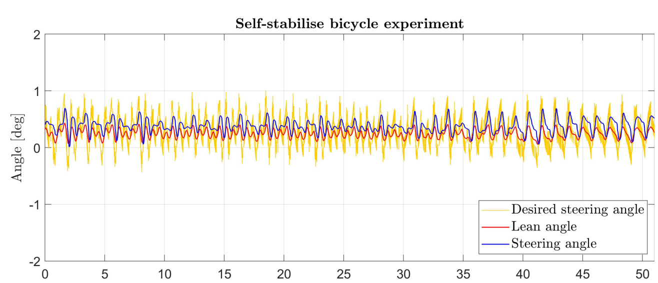

Fig. 9. The result from the bicycle experiment where the bicycle is placed on a roller and a commanded forward speed of 14km/h is used. The signals are logged when the bicycle is self-stabilising on the roller. The PID controller, employed for balancing the bicycle, is first tuned in simulation and then implemented on a roboRIO.

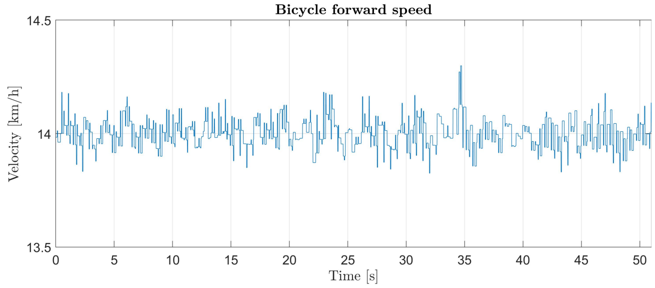

Fig. 10. The forward speed of the bicycle during the experiment. To measure the speed of the bicycle, 12 magnets are mounted on the rear wheel and its pulses are measured with a Hall sensor. By calculating the time between the pulses and knowing the radius of the rear wheel it is possible to calculate the speed of the bicycle.

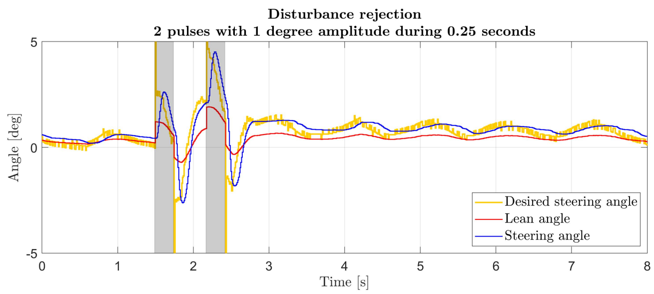

Fig. 11. A disturbance is injected at approximately 1.5 seconds and another one at 2.2 seconds. Each disturbance period is present for 0.25 seconds and each period highlighted by a grey rectangle in the figure. The amplitude of the disturbance, which is injected in the lean angle measurements, is 1 degree.

VII. Conclusion & Future Work

The work presented in this paper showed that it is possible to self stabilise a riderless bicycle using a cascade PID controller. By modelling a dynamic bicycle model, step response matching, and sensor noise a controller can be tune in simulation and adopted on an instrumented bicycle. The roller experiment showed that the instrumented bicycle is self-stabilising for 51 seconds or approximately 200 meters. Additionally, from the disturbance experiment, it is possible to see that the proposed controller can reject disturbance on the lean angle and still maintain to balance the bicycle. Both experiments are cancelled when a human has to interfere with the bicycle, this is due to the width limitations of the bicycle roller and the lack of trajectory tracking which makes bicycle drift to the sides. This could also be the result of asymmetric mass distribution of the bicycle or that the testing ground is not entirely flat. Additionally, in reality, the lean and steering angle might not be initialised to the absolute zero position. This needs to be investigated further, and an outer loop for trajectory tracking should be implemented. Furthermore, to avoid the limitations of the roller, future experiments should be conducted on a flat surface without the confined limitations of the roller.

Acknowledgement

This work is part of an ongoing research project between Mälardalen University, Chalmers University, Volvo Cars, AstaZero, and Cycleurope supported by Vinnova.

References

[1] F. Whipple, *Stability of the Motion of a Bicycle*. Quarterly Journal of Pure and Applied Mathematics 30, 1899.

[2] J. Meijaard, J. M. Papadopoulos, A. Ruina, and A. Schwab, "Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 463, no. 2084, pp. 1955–1982, 2007.

[3] M. Baquero-Suárez, J. Cortés-Romero, J. Arcos-Legarda, and H. Coral-Enriquez, "A robust two-stage active disturbance rejection control for the stabilization of a riderless bicycle," *Multibody System Dynamics*, no. 45, pp. 1–29, 2018.

[4] A. Shafiekhani, M. Mahjoob, and M. Akraminia, "Design and implementation of an adaptive critic-based neuro-fuzzy controller on an unmanned bicycle," *Mechatronics*, vol. 28, pp. 115 – 123, 2015.

[5] D. J. N. Limebeer and R. S. Sharp, "Bicycles, motorcycles, and models," *IEEE Control Systems Magazine*, vol. 26, no. 5, pp. 34–61, Oct 2006.

[6] H. D. Sharma and N. UmaShankar, "A Fuzzy Controller Design for an Autonomous Bicycle System," *2006 IEEE International Conference on Engineering of Intelligent Systems*, pp. 1–6, 2006.

[7] N. H. Getz and J. E. Marsden, "Control for an autonomous bicycle," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 2, May 1995, pp. 1397–1402 vol.2.

[8] A. O'Dwyer, *Handbook of PI and PID controller tuning rules*. Imperial College Press, London, UK, 2009.

[9] Y. Tanaka and T. Murakami, "Self sustaining bicycle robot with steering controller," *Advanced Motion Control, 2004. AMC '04. The 8th IEEE International Workshop on*, no. 2, pp. 193–197, 2004.

[10] A. Suebsomran, "Dynamic compensation and control of a bicycle robot," *2014 International Electrical Engineering Congress, iEECON 2014*, pp. 1–4, 2014.

[11] L. X. Wang, J. M. Eklund, and V. Bhalla, "Simulation & road test results on balance and directional control of an autonomous bicycle," *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering: Vision for a Greener Future, CCECE 2012*, pp. 1–5, 2012.

[12] K. Halbaoui, D. Boukhetala, and F. Boudjema, "Introduction to Robust Control Techniques," *Introduction to Robust Control*, no. May 2014, p. 23, 2007.

[13] C. K. Chen and T. K. Dao, "A study of bicycle dynamics via system identification," *3CA 2010 - 2010 International Symposium on Computer, Communication, Control and Automation*, vol. 2, no. March, pp. 204–207, 2010.

[14] M. A. Anjumol and V. R. Jisha, "Optimal stabilization and straight line tracking of an electric bicycle," in *2014 International Conference on Power Signals Control and Computations (EPSCICON)*, Jan 2014, pp. 1–6.

[15] Y. Tanaka and T. Murakami, "A study on straight-line tracking and posture control in electric bicycle," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 1, pp. 159–168, Jan 2009.

[16] O. Ai-Buraiki and S. El Ferik, "Adaptive control of autonomous bicycle kinematics," *International Conference on Control, Automation and Systems*, no. 2, pp. 783–787, 2013.

[17] R. Y. Abdolmalaki, "Implementation of fuzzy inference engine for equilibrium and roll-angle tracking of riderless bicycle," *CoRR*, vol. abs/1709.09014, 2017. [Online]. Available: http://arxiv.org/abs/1709.09014

[18] K. J. Astrom, R. E. Klein, and A. Lennartsson, "Bicycle dynamics and control: adapted bicycles for education and research," *IEEE Control Systems Magazine*, vol. 25, no. 4, pp. 26–47, Aug 2005.

[19] I. D. Landau and A. Karimi, "Robust digital control using pole placement with sensitivity function shaping method," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 8, no. 2, pp. 191–210, 1998.

[20] D. Bruijnen, R. van de Molengraft, and M. Steinbuch, "Optimization aided loop shaping for motion systems," in *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*. IEEE, 2006, pp. 255–260.

[21] S. M. Cain, J. A. Ashton-Miller, and N. C. Perkins, "On the skill of balancing while riding a bicycle," *PLoS one*, vol. 11, no. 2, p. e0149340, 2016.