2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Gain Scheduled Controller Design for Balancing an Autonomous Bicycle

Shuai Wang[1,*], Leilei Cui[2,*], Jie Lai[1], Sicheng Yang[1], Xiangyu Chen[1], Yu Zheng[1], Zhengyou Zhang[1], *Fellow, IEEE*, and Zhong-Ping Jiang[2], *Fellow, IEEE*

***Abstract*— In this paper, the gain scheduling technique is applied to design a balance controller for an autonomous bicycle with an inertia wheel. Previously, two different balance controllers are needed depending on whether the bicycle is stationary or dynamic. The switch between the two different controllers may cause the instability of the autonomous bicycle. Our proposed gain scheduled controller can balance the autonomous bicycle in both stationary and dynamic cases. A physical system is built and experiments are carried out to demonstrate the effectiveness of the gain scheduled controller.**

I. INTRODUCTION

The control of bicycles has aroused interests of scientists and engineers for a long time [1][2][3]. Considering autonomous bicycles, the balance control is interesting because: the mathematical model of the bicycle is a time-varying system; autonomous bicycles are under-actuated systems [5]; autonomous bicycles are multiple-input multiple-output (MIMO) systems. It is difficult to decouple the effects of the multiple inputs.

When bicycles are moving forward, they can be balanced by the centrifugal force via changing the steering angle [1]. However, when the velocity of bicycles is low, the centrifugal force is too small to balance the bicycle. Thus, the slower the autonomous bicycle moves, the harder it is to balance the bicycle via changing the steering angle. Therefore, it is difficult to balance a stationary bicycle, and other additional components are required. For the autonomous bicycle as shown in Fig. 1, we use a torque-controlled motor to drive the inertia wheel, which was installed between the front wheel and the rear wheel. The inertia wheel is used to provide a reactive torque to assist the self-balancing of the autonomous bicycle when it moves in a low speed or even stops. Besides the torque-controlled motor, the velocity of the autonomous bicycle can be controlled by a servo motor installed at the center of the rear wheel, and the steering angular velocity can be controlled by a stepper motor. According to the existing papers, the operating modes of the autonomous bicycle can be categorized into the following modes.

- *Stationary mode*: the rear wheel of the bicycle stops moving, and the autonomous bicycle is balanced by additional components. Options of the additional components include a rotor mounted on the crossbar [6], a pendulum to balance the autonomous bicycle by tilting force [7], a gyroscope to balance the autonomous bicycle by gyroscopic effect [8][9][10], a high speed flywheel with a single DOF gimbal [11], and an inertia wheel [12][13][14] which is used to provide the balance torque.
- *Dynamic mode*: the rear wheel drives the autonomous bicycle to move at a fast speed, and the autonomous bicycle can be balanced by steering the handlebar only [15][16][17]. It is shown that the bicycle will fall onto the ground if the moving speed is less than a certain value which depends on the physical parameters of the bicycle itself [1][18]. It is almost impossible to keep balance under this dynamic mode once the velocity of the autonomous bicycle is lower than this physical limit.

Fig. 1. RoBicycle - the Autonomous Robot Bicycle System

In the aforementioned papers, most of them only take into consideration one of the operating modes. Therefore if we combine these controllers and apply them to our autonomous bicycle system, the controller needs to be switched between different modes when the autonomous bicycle status changes between being stationary and moving along some certain curve. However, the switch of controllers usually comes from engineering experience. The system stability cannot be guaranteed by theoretical proof. For different autonomous bicycles, due to different physical parameters, the switching time is different. Besides, the switching time and strategy should be determined by a lot of experiments, which are time consuming. In order to overcome the aforementioned limita-

*This work is supported in part by the National Science Foundation under Grant EPCN-1903781, and in part by a gift from the Tencent company. The first two authors contributed equally to this work.

[1]S. Wang, J. Lai, S. Yang, X. Chen, Y. Zheng and Z. Zhang are with Tencent Robotics X, Tencent Holdings, Shenzhen, Guangdong, China (e-mail: shawnshwang@tencent.com)

[2]L. Cui and Z. P. Jiang are with the Control and Networks Lab, Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201 USA (e-mail: l.cui@nyu.edu; zjiang@nyu.edu)

tions, firstly, we establish a linearized dynamic model, where both inertia wheel torque and the steering handlebar rotation are included as system inputs. Then, the gain scheduling technique is applied to design the balance controller based on the linearized dynamic model. The stability of the closed-loop autonomous bicycle is also analyzed.

Primary contributions of this paper are summarized as follows. A gain scheduled balance controller is designed without switching between different controllers. The stability of the closed-loop autonomous bicycle with the gain scheduled balance controller can be guaranteed under mild assumptions. Experiments demonstrate the effectiveness of the gain scheduled balance controller.

This paper is organized as follows. Section II presents the dynamic modeling of the autonomous bicycle, during which both the dynamics of the inertia wheel and the dynamics of the bicycle are considered. In Section III, the gain scheduling controller design method is introduced and applied to the autonomous bicycle. The stability of the closed-loop bicycle is also analyzed. In Section IV, experiments are conducted to demonstrate the effectiveness of the proposed controller. Finally, the conclusion and future works are described in Section V. Due to space limitations, proofs are not included.

II. MATHEMATICAL MODEL OF AN AUTONOMOUS ROBOT BICYCLE

The mathematical model presented in this section includes the effects of the inertial wheel and the steering handlebar on the balance of the bicycle. The trail effect is neglected. Parameters used in the model are depicted in Fig. 2.

In detail, θ is the roll angle of the robot, with $\theta = 0$ being the balancing point assuming the bicycle is not moving. The rotational angle of the inertia wheel is given by ϕ. The main bicycle frame consists of all components on the robot bicycle except for the inertia wheel. The mass of the main bicycle frame and the inertia wheel are given by m_1 and m_2 respectively. The distances from the ground to the center of mass of the main bicycle frame and the inertia wheel are given by L_1 and L_2 respectively. The center of mass of the whole bicycle is denoted by point P. The momentums of inertia of the main bicycle frame and the inertia wheel with respect to x-axis are given by I_1 and I_2 respectively. The bicycle's moving speed is represented by V. The distance between the center of the front wheel and the center of the rear wheel is given by L. The horizontal distance between the center of mass P and the rear wheel center is represented by d. The rotational angle and angular velocity of the front steering handlebar is represented by δ and $\dot{\delta}$, respectively. The gravity acceleration is represented by $g = 9.81 m/s^2$. In Fig. 2(b), when the front handlebar is steered with an angle δ, the radii of the tracks of the front wheel, the center of mass of the robot, and the rear wheel are denoted by R_1, R_2 and R_3, respectively. In practice, L is relatively small compared to the radius of the trajectory, so $R_1 \approx R_2 \approx R_3 \doteq R$. Define the curvature of the trajectory center as $\sigma = 1/R = \tan(\delta)/L$. The output torque of the motor driving the inertia wheel is represented by T_r.

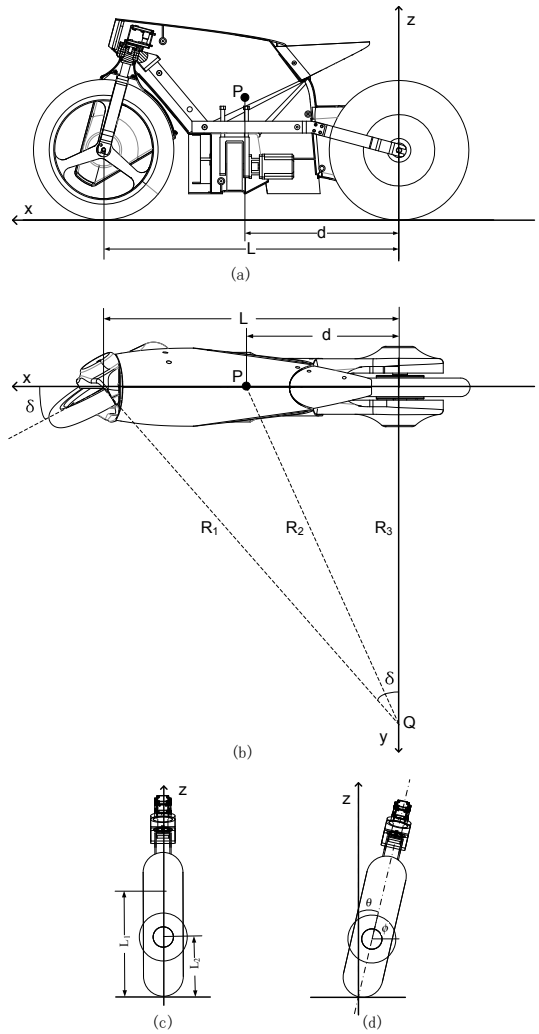

Fig. 2. Definition of parameters: (a) Front view; (b) Top view with with the steering angle δ; (c) Side view (seeing from the front wheel to the rear wheel) when the bicycle is straight up; (d) Side view when the bicycle tilts.

The moving speed of the robot V is independent of θ and ϕ during the movement.

$$\begin{aligned} T_{tran} &= \frac{1}{2}m_1(V_{x1}^2 + V_{y1}^2 + V_{z1}^2) + \frac{1}{2}m_2(V_{x2}^2 + V_{y2}^2 + V_{z2}^2) \\ &= \frac{1}{2}m_1[V^2 + (V\sigma d + L_1\dot{\theta}\cos\theta)^2 + (-L_1\dot{\theta}\sin\theta)^2] \\ &+ \frac{1}{2}m_2[V^2 + (V\sigma d + L_2\dot{\theta}\cos\theta)^2 + (-L_2\dot{\theta}\sin\theta)^2]. \end{aligned} \tag{1}$$

Similarly, the rotational kinetic energy can be derived by considering the bicycle frame and the inertia wheel separately

$$T_{rot} = \frac{1}{2}I_1\dot{\theta}^2 + \frac{1}{2}I_2(\dot{\theta} + \dot{\phi})^2. \tag{2}$$

The potential energy is given as

$$U = (m_1L_1 + m_2L_2)g(1 + \cos\theta). \tag{3}$$

Then according to the Euler-Lagrange Expression

$$\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathscr{L}}{\partial q_i} = \tau_i, \tag{4}$$

where $\mathscr{L} = T_{tran} + T_{rot} - U$, q_i's are generalized coordinates including θ and ϕ, $\dot{q}_i$'s are generalized velocities, and τ_i's

are generalized force with $\tau_\theta = (m_1L_1+m_2L_2)\cos\theta\sigma V^2$ and $\tau_\phi = T_r$. After substituting σ with $\tan(\delta)/L$ and linearizing $\sin\theta \approx \theta$, we have the dynamics model of the robot:

$$(m_1L_1^2+m_2L_2^2+I_1+I_2)\ddot{\theta}+I_2\ddot{\phi}-(m_1L_1+m_2L_2)g\theta = -\frac{V(m_1L_1+m_2L_2)d}{L}\dot{\delta}-\frac{(m_1L_1+m_2L_2)V^2}{L}\delta. \tag{5}$$

$$I_2(\ddot{\theta}+\ddot{\phi}) = T_r. \tag{6}$$

The motor driving the inertia wheel can be modeled [19] as:

$$\begin{aligned} V_m &= L_m\frac{di}{dt}+R_m i+K_e\omega_m, \\ T_m &= K_t i, \\ T_r &= N_g T_m, \end{aligned} \tag{7}$$

where V_m is the driving voltage, K_e is the motor back electromagnetic force, ω_m is the angular velocity of the motor, which equals to $\dot{\phi}$, L_m is the armature coil inductance, R_m is armature coil resistance, i is the armature current, T_m is the motor torque before the gear box, K_t is the motor torque constant and N_g is the gear ratio. T_r is the motor generated torque after the gear box. It is also the torque driving the inertia wheel in this autonomous bicycle. In practice, for a motor $L_m << R_m$. Using the above model in (7), the required motor torque can be calculated as

$$T_r = N_gK_t\Big(\frac{V_m-K_eN_g\dot{\phi}}{R_m}\Big). \tag{8}$$

We combine the bicycle model (5)-(6) with the motor model (8), and then rewrite them in state space representation:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}(V)\boldsymbol{x}(t)+\boldsymbol{B}(V)\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t) \end{cases} \tag{9}$$

where the system output is $\boldsymbol{y}$, the inner state is

$$\boldsymbol{x} = \begin{bmatrix}\theta & \dot{\theta} & \dot{\phi} & \delta\end{bmatrix}^T \tag{10}$$

and the system input is

$$\boldsymbol{u} = \begin{bmatrix}V_m & \dot{\delta}\end{bmatrix}^T \tag{11}$$

with

$$\boldsymbol{A}(V) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{bg}{a} & 0 & \frac{N_g^2K_tK_e}{aR_m} & -\frac{bV^2}{aL} \\ -\frac{bg}{a} & 0 & -\frac{I_2+a}{aI_2}\frac{N_g^2K_tK_e}{R_m} & \frac{bV^2}{aL} \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{12}$$

$$\boldsymbol{B}(V) = \begin{bmatrix} 0 & 0 \\ -\frac{N_gK_t}{aR_m} & \frac{Vbd}{aL} \\ \frac{I_2+a}{aI_2}\frac{N_gK_t}{R_m} & -\frac{Vbd}{aL} \\ 0 & 1 \end{bmatrix}, \tag{13}$$

and

$$\begin{aligned} a &= m_1L_1^2+m_2L_2^2+I_1, \\ b &= m_1L_1+m_2L_2, \end{aligned} \tag{14}$$

where $\boldsymbol{C}$ and $\boldsymbol{D}$ are the identity matrix and zero matrix with appropriate dimensions, respectively.

Because V is changing with respect to time, (9) is a linear time-varying system. Traditional linear time-invariant controller could not work on this system. To solve this problem, we design the controller based on the idea of gain scheduling and prove the stability of the closed-loop bicycle system in Section III.

III. GAIN SCHEDULING AND ITS APPLICATION ON OPTIMAL CONTROL

In this section, firstly we show how to design the state feedback control law using the idea of gain scheduling. Then we present the design procedure in detail.

A. Problem Statement

In the above model, since the state space model is time-varying as the moving speed V changes, we define the system in a more general form. We use an inner parameter α to represent the changing parameter (V in our robot bicycle). The nonlinear system can be rephrased as a linear MIMO parameter-varying system represented for all $t \geq 0$ by:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}(\alpha)\boldsymbol{x}(t)+\boldsymbol{B}(\alpha)\boldsymbol{u}(t), \\ \boldsymbol{y}(t) &= \boldsymbol{C}(\alpha)\boldsymbol{x}(t), \\ \alpha &= \alpha(t), \quad \boldsymbol{x}(0) = \boldsymbol{x_0}, \end{aligned} \tag{15}$$

where for all $t \geq 0$ the state $\boldsymbol{x}(t) \in \mathbb{R}^n$, the input $\boldsymbol{u}(t) \in \mathbb{R}^{n_i}$, and the output $\boldsymbol{y}(t) \in \mathbb{R}^{n_o}$; for all $t \geq 0$ the parameter $\alpha = \alpha(t) \in [\alpha_0, \alpha_n] =: I \subset \mathbb{R}$; for all $\alpha \in I$ the coefficient matrices $\boldsymbol{A}(\alpha) = [a_{ij}(\alpha)] \in \mathbb{R}^{n\times n}$, $\boldsymbol{B}(\alpha) = [b_{ij}(\alpha)] \in \mathbb{R}^{n\times n_i}$, and $\boldsymbol{C}(\alpha) = [c_{ij}(\alpha)] \in \mathbb{R}^{n_o\times n}$; the number of inputs $n_i \leq n$, and for all $\alpha \in I$ the matrix $\boldsymbol{B}(\alpha)$ is full column rank.

The following assumptions are made [20]:

Assumption 3.1: The parameter α is a continuous and bounded function of t, differentiable almost everywhere with bounded derivative, and is measured for all $t \geq 0$.

Assumption 3.2: The system described in the first formula of (15) is completely controllable for all $\alpha \in I$.

We apply the state feedback control law

$$\begin{aligned} \boldsymbol{u}(t) &= -\boldsymbol{K}(\alpha)\boldsymbol{x}(t), \\ \alpha &= \alpha(t), \end{aligned} \tag{16}$$

to the system (15) for all $t \geq 0$, where for all $\alpha \in I = [\alpha_0, \alpha_n]$ the matrix $\boldsymbol{K}(\alpha) \in \mathbb{R}^{n_i\times n}$ is the state feedback gain matrix.

Remark 3.3: For a more general case, $\boldsymbol{u}(t) = -\boldsymbol{K}(\alpha)\boldsymbol{x}(t)+\boldsymbol{v}(t)$ is used in (16), and for all $t \geq 0$ the input $\boldsymbol{v}(t) \in \mathbb{R}^{n_i}$ is some exogenous input to the system. In this paper, the problem can be described as a special case when $\boldsymbol{v}(t) = \boldsymbol{0}$.

For a fixed α, the control law can be designed to ensure stability by traditional linear control methods. As the inner parameter changes, the control law design is given by gain scheduling.

B. Gain Scheduling

We describe the main idea of gain scheduling [20][21] as follows. We choose a finite number of fixed $\alpha_l \in I$. For each fixed α_l, the gain matrix $\boldsymbol{K}(\alpha_l)$ is designed by the linear quadratic regulator (LQR), i.e. $\boldsymbol{K}(\alpha_l) = \boldsymbol{R}^{-1}\boldsymbol{B}^T(\alpha_l)\boldsymbol{P}(\alpha_l)$,

where $\boldsymbol{P}(\alpha_l)$ is the solution to the following algebraic Riccati equation

$$\begin{aligned} &\boldsymbol{A}^T(\alpha_l)\boldsymbol{P}(\alpha_l)+\boldsymbol{P}(\alpha_l)\boldsymbol{A}(\alpha_l)-\boldsymbol{P}(\alpha_l)\boldsymbol{L}\boldsymbol{P}(\alpha_l)+\boldsymbol{Q}(\alpha_l)=\boldsymbol{0}, \\ &\boldsymbol{L}(\alpha_l)=\boldsymbol{B}(\alpha_l)\boldsymbol{R}^{-1}(\alpha_l)\boldsymbol{B}^T(\alpha_l). \end{aligned} \tag{17}$$

For $\alpha \in [\alpha_l, \alpha_{l+1}]$, the gain matrix $\boldsymbol{K}(\alpha)$ can be determined by the linear interpolation between $\boldsymbol{K}(\alpha_l)$ and $\boldsymbol{K}(\alpha_{l+1})$, which can be expressed as

$$\boldsymbol{K}(\alpha)=\boldsymbol{K}(\alpha_l)+\frac{\boldsymbol{K}(\alpha_{l+1})-\boldsymbol{K}(\alpha_l)}{\alpha_{l+1}-\alpha_l}(\alpha-\alpha_l). \tag{18}$$

Intuitively, by choosing an appropriate number of $\alpha_l \in I$, provided that the rate of change of α is sufficiently small, the design criteria for the system (15) are satisfied.

Theorem 3.4: Let Assumption 3.1-3.2 hold and $h = \alpha_{l+1} - \alpha_l$. For the system (15) with the controller $\boldsymbol{u} = -\boldsymbol{K}(\alpha)\boldsymbol{x}$, where $\boldsymbol{K}(\alpha)$ is obtained by (18), there exists $\mu, \varepsilon > 0$, such that if $|\dot{\alpha}| \leq \mu$ and $h < \varepsilon$, $\lim_{t\to\infty}\boldsymbol{x} = \boldsymbol{0}$.

Proof: According to (17), it can be easily checked that $\boldsymbol{P}(\alpha_l+h) = \boldsymbol{P}(\alpha_l)+\boldsymbol{o}(h)$, and $\boldsymbol{K}(\alpha_l+h) = \boldsymbol{K}(\alpha_l)+\boldsymbol{o}(h)$. From (18), it is clear that $\boldsymbol{K}(\alpha) = \boldsymbol{K}(\alpha_l)+\boldsymbol{o}(\alpha-\alpha_l)$. Define $\boldsymbol{A}_c(\alpha) = \boldsymbol{A}(\alpha)-\boldsymbol{B}(\alpha)\boldsymbol{K}(\alpha)$, and let $\lambda(\boldsymbol{A})$ denote the eigenvalues of $\boldsymbol{A}$. Thus $\boldsymbol{A}_c(\alpha) = \boldsymbol{A}_c(\alpha_l)+\boldsymbol{o}(\alpha-\alpha_l)$. Now applying the results on analytic perturbation of eigenvalues of a matrix [23], [24], one can obtain $\lambda(\boldsymbol{A}_c(\alpha)) = \lambda(\boldsymbol{A}_c(\alpha_l))+\boldsymbol{o}(\alpha-\alpha_l)$. Therefore, when $h \to 0$, $\lambda(\boldsymbol{A}_c(\alpha)) \in \mathbb{C}^-$. Thus according to the Lemma 3.5 in [20], the theorem can be proved. ■

C. Application in the Control of Autonomous Bicycle

For the autonomous bicycle, via Popov–Belevitch–Hautus test, one can easily find that for any velocity V, $(\boldsymbol{A}, \boldsymbol{B})$ are controllable. Besides, in practice, the velocity of the bicycle is continuous with respect to time. Thus Assumptions 3.1-3.2 are satisfied.

Now we consider the application of gain scheduling in the control of the autonomous robot bicycle model. Matrices in the state space representation $\boldsymbol{A}(V)$ and $\boldsymbol{B}(V)$ are time-varying with $V(t)$. At a series of fixed speeds $V = V_l$, for $l = 1, 2, \ldots n$, the LQR controller is used to obtain the gain matrix $\boldsymbol{K}(V_l)$. When choosing the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ in the controller design, practical physical limitations should be taken into consideration. For example, there is a rotational speed limit of the inertia wheel driving motor. The selection of parameters in $\boldsymbol{Q}$ and $\boldsymbol{R}$ needs to ensure the motor operates under this speed limitation. As we know, the inertia wheel needs to work when the bicycle's moving speed is slow, and the operation of the steering wheel is helpful but not a must. Similarly, when the bicycle runs at a relatively high speed with the rotation of the steering handlebar and the centrifugal force is large enough, the operation of the steering wheel is rather important for the system compared to the inertia wheel. It means that we need to adjust the parameters of controller design so that under different situations, two inputs of the system contribute accordingly as we expected. One option is to set $\boldsymbol{R} = \mathrm{diag}(r_{11}, r_{22}/(V^\gamma+\varepsilon))$ for sufficient small ε and $\gamma > 1$. In the experiments proposed in this paper, by testing performances when the bicycle is stationary and moving along a straight line or a circle as the special cases respectively, $\gamma = 4$ is used. Then the value $r_{11} = 10$ and $r_{22} = 20$ are used considering the contributions of the inertia wheel and the steering handlebar at different bicycle's moving speeds. According to Theorem 3.4, applying the gain matrix $\boldsymbol{K}(V)$, with V covering the whole speed range, gain scheduling control guarantees the stability of the closed-loop bicycle system.

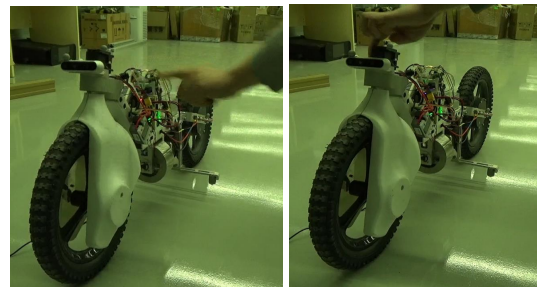

Fig. 3. Static disturbances experiments: (a) a disturbance towards one direction; (b) a disturbance towards the opposite direction.

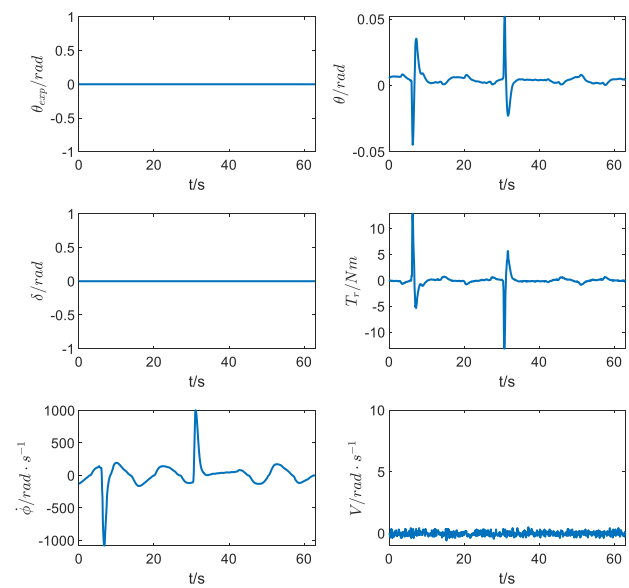

Fig. 4. Results of Experiment 1: Static disturbances

IV. Experiments

A. Experimental Setup

A two-wheeled autonomous robot bicycle depicted in Fig. 1, namely RoBicycle, is built up. It is a nonlinear system with three driving inputs: the steering handlebar, the rear wheel rotation and the inertia wheel. The steering handlebar is driven by a servoing motor and can be controlled by either position or speed instructions. The inertia wheel motor is driven by torque using Field-Oriented Control. The rear wheel can be driven by either torque command or speed command.

Although practical applications of RoBicycle demonstrate good performance when it runs on both flat surface and small

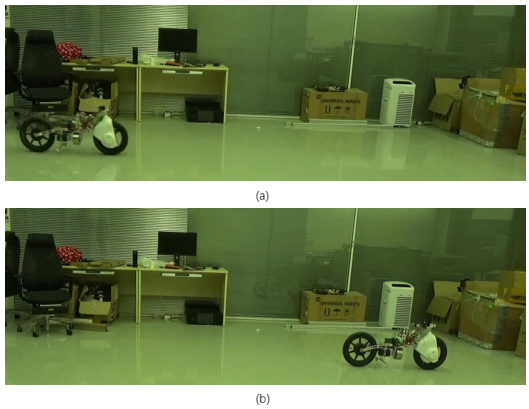

Fig. 5. Accelerating and braking experiment: (a) RoBicycle started running on the left of the screen; (b) RoBicycle stopped on the right of the screen.

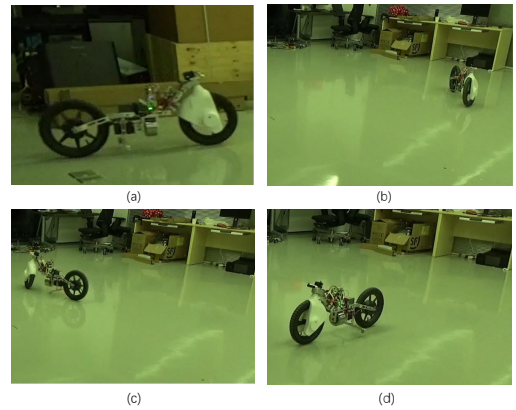

Fig. 7. Hybrid Line - Circle - Line experiment: (a) started moving forward; (b)-(c) run circle; (d) changed the curve from a circle back to a straight line.

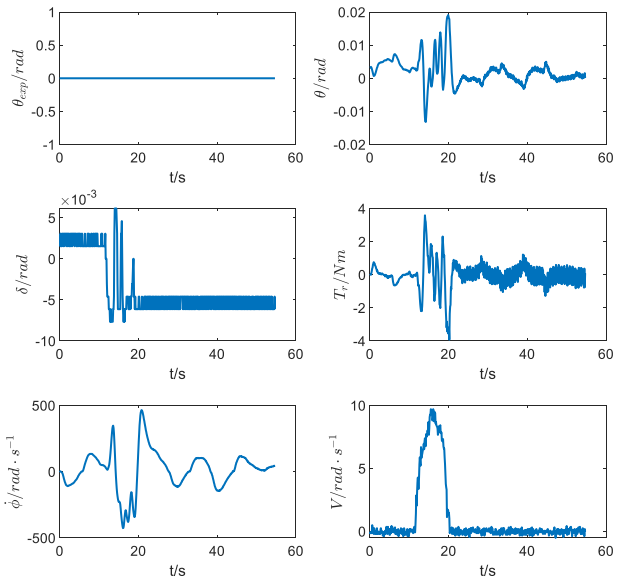

Fig. 6. Results of Experiment 2: Accelerating and braking

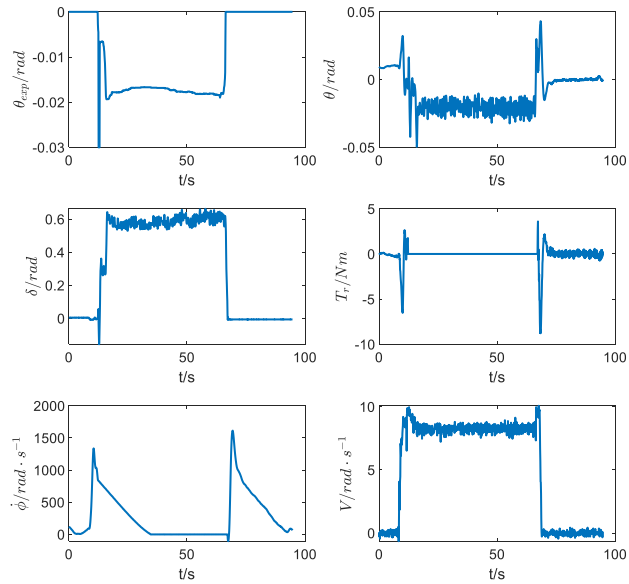

Fig. 8. Results of Experiment 3: Hybrid Line - Circle - Line

bumps or bridges, the analyses and control of the bicycle running on the non-flat surface are beyond the scope of this paper. In this paper, all experiments have been performed on a flat non-slippery surface. The full robot state information is available by using the IMU to obtain Euler angle and attitude of the robot with Kalman filter and data fusion, and encoders of the driving motors to monitor the rotation angle of each motor. The control law of the robot has been implemented on an STM32H7 chip in C language with a sampling time of $T_s = 0.005s$. ROS middleware is used for data transmission and recording and other high-level strategy, such as planning and decision, which is beyond the scope of this paper.

B. Experimental Results

Three experiments were performed to show the effectiveness and robustness of the autonomous robot bicycle system with the gain scheduled controller. These experiments are shown in the accompanying video.

1) *Static disturbances*: As shown in Fig. 3(a)-(b), after the autonomous robot bicycle kept self-balanced in the equilibrium point for a few seconds, a disturbance towards one direction was made by human hand before the system returned to the equilibrium point and the inertia wheel rotational angular velocity slowed down. Then a similar disturbance towards the opposite direction was made. The results are depicted in Fig. 4. After each disturbance, the Euler angle θ changed dramatically. The inertia wheel controller gave a peak torque over 10 Nm before the tilting Euler angle moved back to the equilibrium point. The inertia wheel rotational angular velocity $\dot{\phi}$ also changed during this period as a result of the torque applied to the flywheel. The rear wheel did not move, so the speed $V = 0$, with some noises obtained from the rear wheel motor encoder.

2) *Accelerating and braking*: As shown in Fig. 5(a)-(b), RoBicycle started balancing from the standstill, and then moved forward. The human operator gave an acceleration instruction to the rear wheel using the remote controller followed by a braking instruction to stop the moving. The

RoBicycle is balanced by the proposed controller during the whole accelerating and braking period. The results are depicted in Fig. 6. The accelerating and braking could be regarded as external disturbances to the balance control. That is why θ, T_r and $\dot{\phi}$ changed largely during the speed changing between 10-22 s. The steering handlebar angle changed at the time when RoBicycle started moving forward to correct a static error of the initial value once it was powered up. Once it stopped moving forward, the Euler angle damped before it finally settled down around the balancing point. Tests on uneven surfaces such as going across deceleration belts also demonstrate the effectiveness and robustness of the controller.

3) *Hybrid Line - Circle - Line - Stop*: As shown in Fig. 7(a)-(d), the RoBicycle accelerated along a straight line. Then the human operator sent a desired Euler angle using the remote controller to change the path to a circle. After the bicycle running along a circle curve for about 60 seconds, an instruction was given to change the path back to a straight line, before a braking signal to speed it down. The RoBicycle kept self-balanced through the whole procedure, even when the moving speed turned back to 0 finally. The results are depicted in Fig. 8. In this experiment, all previous control design and strategies were applied. Initially, when the desired Euler angle is zero, the inertia wheel torque contributed greatly to the self-balance, with the steering handlebar angle almost unchanged. After the bicycle received the deired Euler angle and moved along a circle, the handlebar contributed largely to the balancing, with the inertia wheel speed slowing down. The bicycle can balance itself to the desired Euler angle with the proposed controller. Again, RoBicycle kept self-balanced by the gain scheduled controller during the whole Stationary - Line - Circle - Line - Stop period. It was shown that the robotic system was stable with the proposed gain scheduled controller.

V. CONCLUSIONS

In this paper, we have designed a gain scheduled controller to balance an autonomous bicycle. Compared with other controllers, the proposed controller can balance the bicycle without switching controllers between the stationary mode and the dynamic mode. Experiments demonstrate that the proposed gain scheduled balance controller is effective whether the autonomous bicycle is stationary or moving. Our future work will be dedicated to the model-free robust balance controller design by means of robust dynamic programming techniques developed in [25].

REFERENCES

[1] J. D. G. Kooijman, J. P. Meijaard, J. M. Papadopoulos, A. Ruina, and A. L. Schwab, "A bicycle can be self-stable without gyroscopic or caster effects," *Science*, vol. 332, no. 6027, pp. 339-342, 2011.

[2] W. J. M. Rankine, "On the dynamical principles of the motion of velocipedes," *The Engineer*, vol. 28, no. 79, pp. 129, 1869.

[3] M. Cook, "It takes two neurons to ride a bicycle," Demonstration at NIPS 4, 2004.

[4] L. Guo, Q. Liao, S. Wei, and Y. Huang, "A kind of bicycle robot dynamic modeling and nonlinear control," In *2010 IEEE International Conference on Information and Automation*, IEEE, 2010, pp. 1613-1617.

[5] C.-L. Hwang, H.-M. Wu, and C.-L. Shih, "Fuzzy sliding-mode underactuated control for autonomous dynamic balance of an electrical bicycle," *IEEE Transactions on Control Systems Technology,* vol. 17, no. 3, 2009, pp. 658-670.

[6] Y. Yavin, "Stabilization and control of the motion of an autonomous bicycle by using a rotor for the tilting moment," *Computer Methods in Applied Mechanics and Engineering,* vol. 178, no. 3-4, 1999, pp. 233-243.

[7] S. Lee and W. Ham, "Self stabilizing strategy in tracking control of unmanned electric bicycle with mass balance," In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, IEEE, 2002, pp. 2200-2205.

[8] A. V. Beznos, A. M. Formal'Sky, E. V. Gurfinkel, D. N. Jicharev, A. V. Lensky, K. V. Savitsky, and L. S. Tchesalin, "Control of autonomous motion of two-wheel bicycle with gyroscopic stabilisation," In *Proceedings of 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 3, IEEE, 1998, pp. 2670-2675.

[9] H. Yetkin and U. Ozguner, "Stabilizing control of an autonomous bicycle," In *2013 9th Asian Control Conference (ASCC)*, IEEE, 2013, pp. 1-6.

[10] Y. Zhang, P. Wang, J. Yi, D. Song, and T. Liu, "Stationary balance control of a bikebot," In *Proceedings of 2014 IEEE International Conference on Robotics and Automation*, Hongkong, 2014, pp. 6706-6711.

[11] H. Yetkin, S. Kalouche, M. Vernier, G. Colvin, K. Redmill, and U. Ozguner, "Gyroscopic stabilization of an unmanned bicycle," In *2014 American Control Conference (ACC)*, IEEE, 2014, pp. 4549-4554.

[12] K. Kanjanawanishkul, "LQR and MPC controller design and comparison for a stationary self-balancing bicycle robot with a reaction wheel," *Kybernetika*, 2015, vol. 51, no. 1, pp. 173-191.

[13] H.-W. Kim, J.-W. An, H. D. Yoo, and J.-M. Lee, "Balancing control of bicycle robot using PID control," In *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, IEEE, 2013, pp. 145-147.

[14] Y. Kim, H. Kim, and J. Lee, "Stable control of the bicycle robot on a curved path by using a reaction wheel," *Journal of Mechanical Science and Technology,* vol. 29, no. 5, 2015, pp. 2219-2226.

[15] K. J. Astrom, R. E. Klein, and A. Lennartsson, "Bicycle dynamics and control," *IEEE Control Systems Magazine*, vol. 25, no. 4, IEEE, 2005, pp. 26-47.

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

[17] Q. He, X. Fan, and D. Ma, "Full bicycle dynamic model for interactive bicycle simulator," 2005, pp. 373-380.

[18] C. Yang and T. Murakami, "Full-speed range self-balancing electric motorcycles without the handlebar," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, IEEE, 2015, pp. 1911-1922.

[19] A. Türkmen, M. Y. Korkut, M. Erdem, O. Gonul, and V. Sezer, "Design, implementation and control of dual axis self balancing inverted pendulum using reaction wheels," In *2017 10th International Conference on Electrical and Electronics Engineering (ELECO)*, IEEE, 2017, pp. 717-721.

[20] S. M. Shahruz and S. Behtash, "Design of controllers for linear parameter-varying systems by the gain scheduling technique," *Journal of Mathematical Analysis and Application*, vol. 168, pp. 195-217, 1990.

[21] D. J. Leith and W. E. Leithead, "Survey of gain-scheduling analysis and design," *International Journal of Control*, vol. 73, no. 11, 2000, pp. 1001-1025.

[22] J. S. Shamma, "Analysis and design of gain scheduled control systems," PhD diss., Massachusetts Institute of Technology, 1988.

[23] T. Kato, "A short introduction to perturbation theory for linear operators," *Springer Science & Business Media*, 2012.

[24] P. Lancaster and M. Tismenetsky, The theory of matrices: with applications, Elsevier, 1985.

[25] T. Bian and Z. P. Jiang, "Continuous-time robust dynamic programming," SIAM Journal on Control and Optimization, vol. 57, no. 6, pp. 4150-4174, 2019.