Oral Presentation

The 7th TSME International Conference on Mechanical Engineering
13-16 December 2016

TSME ICOME 2016
Mechanical Engineering Innovation for Green Society

DRC0007

Control of Bicycle Leaning with Steering and Mass-Moving Stabilization

Pongsakorn Seekhao[1], Kanokvate Tungpimolrut[2], and Manukid Parnichkun[1,*]

[1] Mechatronics, Asian Institute of Technology, Klong Luang, Pathumthani, 12120, Thailand
[2] National Electronics and Computer Technology Center, Klong Luang, Pathumthani, 12120, Thailand
[*] Corresponding Author: E-mail: manukid@ait.asia, Phone: +662-524-5229, Fax: +662-524-5697

Abstract

This paper presents the design of an unmanned electrical bicycle which can balance itself in the upright position while moving at a constant forward speed. The bicycle is balanced by applying mass-moving together with steering to share the balancing control load. The nonlinear dynamics model of a bicycle along with a balancing mass is derived from the Euler-Lagrange equation of motion and nonholonomic constraints with respect to translation and rotation relative to the ground plane. This nonlinear dynamics model is then linearized around the upright position and combined with DC motor model to obtain the complete linearized dynamics model. The linear quadratic regulator (LQR) is implemented on the bicycle to control its balance. The simulation results using MATLAB/Simulink show that the system using both steering and mass-moving in balancing obtains better performance in terms of leaning range and balancing time than the system using only steering in balancing. Real experimental results are also in line with the simulation results.

Keywords: bicycle leaning, bicycle balancing, steering, mass-moving, linear quadratic regulator

1. Introduction

Electrical bicycles are very useful in transportation because of their advantages in environmental friendliness, light weight, and capability of traveling in narrow roads, when compared with other vehicles using internal combustion engines. However, the bicycles are unstable in nature. Without proper control, they easily fall down. Hence, the development of self-balancing electrical bicycles or bicycle robots is a very interesting topic for many researchers in recent years. An example of bicycle robots is Murata Boy developed by a Japanese company in 2005 [1].

Bicycle robots have many potential applications such as used as autonomous riderless bicycles for carrying products, used for disable people, and used as learning bicycles for new riders or children. Even for those people who have already known how to ride bicycles, it will be more comfortable and safer if no balancing effort is required during riding. Furthermore, balancing control of a bicycle robot is a challenge topic for many researchers in dynamics control and robotics fields because of the instability, nonlinear characteristics, parameter variations, and uncontrolled external disturbances of the system. Therefore, bicycle robot is an interesting platform to test and verify performances of control algorithms.

Balancing of bicycle robots can be separated into three major methods: using flywheel, heading steering, and mass moving.

In balancing by using flywheel [1-3], one or more flywheels are mounted on the bicycle robot in various ways to generate a torque against the bicycle's falling. Bicycle robots using this balancing method can be balanced at both zero and low forward speeds.

In balancing by heading steering [4-6], a steering motor is mounted on the bicycle robot to change the heading angle in order to generate a centrifugal force to balance the bicycle. Bicycle robots using this balancing method cannot be balanced at zero speed but can be balanced easier at higher speed compared with using other methods.

In balancing by mass moving [7-9], a balancing mass is mounted on the bicycle robot and can be moved in order to keep the combined center of gravity (COG) of the whole system on the line between both wheel-ground contact points such that the bicycle can remain upright. Bicycle robots using this balancing method can be balanced at both zero and low forward speeds. However, this method requires very high balancing torque and wide mass-moving range, which causes the bicycle to be easily unstable.

In this paper, an integrated stabilizing technique by both steering and mass moving is proposed to balance the bicycle robot in the upright position at a constant forward speed. Nonlinear dynamics model of this robot is derived by applying the Euler-Lagrange equation of motion to the model which consists of a bicycle and a downward balancing mass or pendulum. The derived nonlinear model is then linearized around the upright position and also combined with DC motor model to obtain the completed linearized dynamics model. LQR is implemented on the system, and then the control performance is evaluated by simulations and experiments.

Contents of this paper are organized as follows. In section 2, the dynamics model of the bicycle robot is derived. Section 3 describes the concept of LQR. Simulation and experimental results are shown in section 4. Finally, conclusions are mentioned in section 5.

2. Dynamics Model of Bicycle Robot

2.1 Mechanical Model

Fig. 1 and Fig. 2 show coordinate system and parameters of the bicycle robot in side view and back view respectively. The direction of bicycle forward movement is defined as *x*-axis, the right side direction is defined as *y*-axis, and the downward vertical direction is defined as *z*-axis. The model consists of two individual components: a bicycle and a downward balancing mass (which is also called as a pendulum). Three DC motors are installed in the bicycle robot for moving the pendulum, steering, and driving the bicycle.

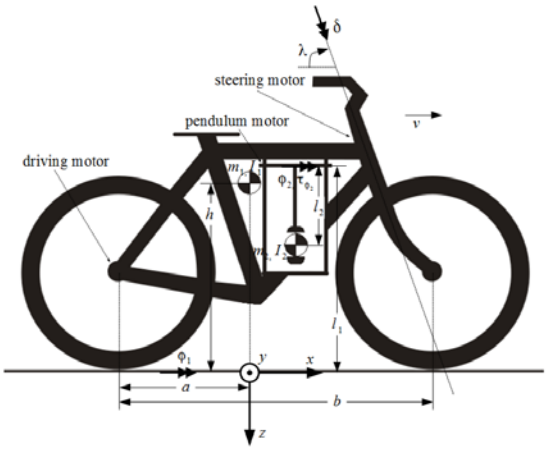

Fig. 1 Bicycle model in side view

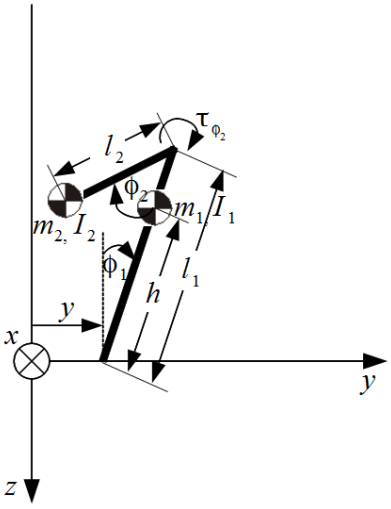

Fig. 2 Bicycle model in back view

Parameters in the model are defined as follows.

- m_1 is the mass of the bicycle (excluding pendulum).
- m_2 is the mass of the pendulum.
- I_1 is the inertia of the bicycle about *x*-axis.
- I_2 is the inertia of the pendulum about *x*-axis.
- a is the projection distance along *x*-axis from the rear wheel-ground contact point to the bicycle's COG.
- b is the wheel base.
- h is the length measured from the ground to the bicycle's COG.
- l_1 is the length measured from the ground to the pendulum's rotation axis.
- l_2 is the length measured from the pendulum's rotation axis to the pendulum's COG.
- λ is the head angle.
- v is the speed of the bicycle.
- ϕ_1 is the lean angle (measured from the vertical axis).
- ϕ_2 is the pendulum angle (in respect of the lean angle).
- δ is the steering angle.
- τ_{ϕ_2} is the required torque for pendulum moving.

Each wheel is assumed thin and thus touches the ground at a single contact point. The wheels, which are also assumed non-slipping, are modeled by holonomic constraint in the normal (vertical) direction and by nonholonomic constraint in the longitudinal and lateral directions. There is no aerodynamic drag, no frame flexibility, no suspension system, and no propulsion (the speed v is constant). Friction in the system is assumed very small and negligible. From Fig. 2, the system may also be considered moving laterally along *y*-axis while it is moving forward with steering, thus the system has three controlled degrees of freedom which are ϕ_1, ϕ_2, and y.

The Euler-Lagrange equation of motion as shown in Eq. (1) is used to derived the dynamics model of the bicycle robot.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_n}\right) - \frac{\partial L}{\partial q_n} = Q_n \qquad (1)$$

where

- L is the Lagrangian, defined as $L = K - P$.
- K is the total kinetic energy.
- P is the total potential energy.
- q_n is the generalized coordinate for the *n*th degree of freedom.
- Q_n is the generalized torque (or force) for the *n*th degree of freedom.

By solving Eq. (1), three nonlinear equations of motion for $q_n = \phi_1$, ϕ_2, and y are derived as Eqs. (2) – (4) respectively.

$$\left(I_1 + I_2 + m_1h^2 + m_2l_1^2 + m_2l_2^2 - 2m_2l_1l_2\cos\phi_2\right)\ddot{\phi}_1 + \left(I_2 + m_2l_2^2 - m_2l_1l_2\cos\phi_2\right)\ddot{\phi}_2$$

$$+\left(m_1 h\cos\phi_1 + m_2 l_1\cos\phi_1 - m_2 l_2\cos(\phi_1+\phi_2)\right)\ddot{y}$$
$$+ m_2 l_1 l_2\sin\phi_2\dot{\phi}_2^{\,2} + 2m_2 l_1 l_2\sin\phi_2\dot{\phi}_1\dot{\phi}_2$$
$$+\left(-m_1 h\sin\phi_1 - m_2 l_1\sin\phi_1 + m_2 l_2\sin(\phi_1+\phi_2)\right)g$$
$$= 0 \qquad (2)$$

$$\left(I_2 + m_2 l_2^{\,2} - m_2 l_1 l_2\cos\phi_2\right)\ddot{\phi}_1 + \left(I_2 + m_2 l_2^{\,2}\right)\ddot{\phi}_2$$
$$- m_2 l_2\cos(\phi_1+\phi_2)\ddot{y} - m_2 l_1 l_2\sin\phi_2\dot{\phi}_1^{\,2}$$
$$+ m_2 l_2\sin(\phi_1+\phi_2)g$$
$$= \tau_{\phi_2} \qquad (3)$$

$$\left(m_1 h\cos\phi_1 + m_2 l_1\cos\phi_1 - m_2 l_2\cos(\phi_1+\phi_2)\right)\ddot{\phi}_1$$
$$- m_2 l_2\cos(\phi_1+\phi_2)\ddot{\phi}_2 + (m_1+m_2)\ddot{y}$$
$$+\left(-m_1 h\sin\phi_1 - m_2 l_1\sin\phi_1 + m_2 l_2\sin(\phi_1+\phi_2)\right)\dot{\phi}_1^{\,2}$$
$$+ m_2 l_2\sin(\phi_1+\phi_2)\dot{\phi}_2^{\,2} + 2m_2 l_2\sin(\phi_1+\phi_2)\dot{\phi}_1\dot{\phi}_2$$
$$= \left(\frac{(m_1+m_2)v^2\sin\lambda}{b}\right)\delta \qquad (4)$$

It is observed from the above equations that there is no torque for $q_n = \phi_1$ because that joint is passive. The torque for $q_n = \phi_2$ is directly derived from the pendulum's motor torque and its transmission ratio. Moreover, the force for $q_n = y$ is the centrifugal force generated while the system is moving forward in curvature.

For simplicity, nonlinear equations of motion are linearized around the points of $\phi_1 \approx 0$ and $\phi_2 \approx 0$, then Eqs. (2) – (4) become Eqs. (5) – (7) respectively.

$$\left(I_1 + I_2 + m_1 h^2 + m_2(l_1-l_2)^2\right)\ddot{\phi}_1$$
$$+\left(I_2 + m_2 l_2(l_2-l_1)\right)\ddot{\phi}_2 + \left(m_1 h + m_2(l_1-l_2)\right)\ddot{y}$$
$$+\left(-m_1 h - m_2(l_1-l_2)\right)g\phi_1 + m_2 l_2 g\phi_2$$
$$= 0 \qquad (5)$$

$$\left(I_2 + m_2 l_2(l_2-l_1)\right)\ddot{\phi}_1 + \left(I_2 + m_2 l_2^{\,2}\right)\ddot{\phi}_2$$
$$- m_2 l_2\ddot{y} + m_2 l_2 g\phi_1 + m_2 l_2 g\phi_2$$
$$= \tau_{\phi_2} \qquad (6)$$

$$\left(m_1 h + m_2(l_1-l_2)\right)\ddot{\phi}_1 - m_2 l_2\ddot{\phi}_2 + (m_1+m_2)\ddot{y}$$
$$= \left(\frac{(m_1+m_2)v^2\sin\lambda}{b}\right)\delta \qquad (7)$$

Since driver circuits of both pendulum motor and steering motor require their inputs as PWM (Pulse Width Modulation) signals whose duty cycles are proportional to the corresponding voltages, both balancing-motor models have to be included in the linearized mechanical model to form the voltage-input system model.

2.2 Pendulum Motor Model

Eq. (6) shows that the required torque generated from the pendulum motor and its transmission needs to be controlled. For simplicity, the armature inductance of the DC motor is assumed very small and negligible. The required torque equation is derived as Eq. (8).

$$\tau_{\phi_2} = \left(\frac{k_g k_\tau}{R}\right)V_{\phi_2} - \left(\frac{k_g^2 k_b k_\tau}{R}\right)\dot{\phi}_2 \qquad (8)$$

where

- k_g is the transmission ratio.
- k_b is the back emf (electromotive force) constant of the motor.
- k_τ is the torque constant of the motor.
- R is the armature resistance.
- V_{ϕ_2} is the applied armature voltage.

By substituting Eq. (8) into Eq. (6),

$$\left(I_2 + m_2 l_2(l_2-l_1)\right)\ddot{\phi}_1 + \left(I_2 + m_2 l_2^{\,2}\right)\ddot{\phi}_2 - m_2 l_2\ddot{y}$$
$$+\left(\frac{k_g^2 k_b k_\tau}{R}\right)\dot{\phi}_2 + m_2 l_2 g\phi_1 + m_2 l_2 g\phi_2$$
$$= \left(\frac{k_g k_\tau}{R}\right)V_{\phi_2} \qquad (9)$$

All three equations of motion as expressed in Eq. (5), Eq. (9), and Eq. (7) can be rewritten in the matrix form.

$$M\ddot{\Phi} + H\dot{\Phi} + G\Phi = Q \qquad (10)$$

where

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}$$

$$M_{11} = I_1 + I_2 + m_1 h^2 + m_2(l_1-l_2)^2$$
$$M_{12} = I_2 + m_2 l_2(l_2-l_1)$$
$$M_{13} = m_1 h + m_2(l_1-l_2)$$
$$M_{21} = M_{12}$$
$$M_{22} = I_2 + m_2 l_2^{\,2}$$
$$M_{23} = -m_2 l_2$$
$$M_{31} = M_{13}$$
$$M_{32} = M_{23}$$
$$M_{33} = m_1 + m_2$$

$$H = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \dfrac{k_g^2 k_b k_\tau}{R} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} (-m_1 h - m_2(l_1 - l_2))g & m_2 l_2 g & 0 \\ m_2 l_2 g & m_2 l_2 g & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 \\ \left(\dfrac{k_g k_\tau}{R}\right) V_{\phi_2} \\ \left(\dfrac{(m_1 + m_2) v^2 \sin\lambda}{b}\right)\delta \end{bmatrix}, \quad \Phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ y \end{bmatrix}$$

From the matrix Q of Eq. (10), it is observed that the input applied to the pendulum motor is already the function of the voltage, but the input applied to the steering motor is still the function of the steering angle, so the steering motor and its load (the front-fork assembly) is considered in the next subsection.

2.3 Steering Motor Model

Eq. (7) shows that the required steering angle moved by the steering motor and its transmission needs to be controlled. For simplicity, the armature inductance of the DC motor is assumed very small and negligible. The equation representing the relation between the required steering angle and the applied voltage (V_δ) is derived as Eq. (11).

$$\ddot{\delta} + k_1 \dot{\delta} = k_2 V_\delta \tag{11}$$

where

- $k_1 = \dfrac{1}{J_{eq}}\left(D_{eq} + \dfrac{k_b k_\tau}{R}\right)$
- $k_2 = \dfrac{k_\tau}{R J_{eq}}$
- J_{eq} is the equivalent inertia at the armature (which includes both the armature inertia and the load inertia).
- D_{eq} is the equivalent viscous damping at the armature (which includes both the armature viscous damping and the load viscous damping).

Since the steering motor is loaded with the front-fork assembly, all steering components are considered as one subsystem. Therefore, k_1 and k_2 are derived by applying MATLAB System Identification Toolbox based on Eq. (11) with the experimental data which are collected from randomly moving the steering motor along with the front-fork assembly around the position of $\delta = 0$.

By combining Eq. (10) with Eq. (11), the completed linearized dynamics model can be represented by the state-space model.

$$\dot{x} = Ax + Bu \tag{12}$$

where

$$x = \begin{bmatrix} \phi_1 \\ \phi_2 \\ y \\ \delta \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{y} \\ \dot{\delta} \end{bmatrix}, \qquad u = \begin{bmatrix} V_{\phi_2} \\ V_\delta \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ & & & & & & & 0 \\ \left[-M^{-1}G\right] & \Delta & & \left[-M^{-1}H\right] & & & & 0 \\ & & & & & & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -k_1 \end{bmatrix}$$

$$\Delta = M^{-1}\begin{bmatrix} 0 \\ 0 \\ \dfrac{(m_1 + m_2) v^2 \sin\lambda}{b} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ & 0 \\ M^{-1}\begin{bmatrix} 0 \\ \dfrac{k_g k_\tau}{R} \\ 0 \end{bmatrix} & 0 \\ & 0 \\ 0 & k_2 \end{bmatrix}$$

The derived state-space model completely represents the system because it combines the bicycle dynamics together with motor models. The voltage inputs make the system model ready to be implemented because the control gains can be directly used to calculate the PWM signals of both motors.

3. LQR

The theory of optimal control [10] is concerned with operating a dynamics system at minimum cost. The case where the system dynamics are described by a set of linear differential equations and the cost is described by a quadratic function is called the LQ problem.

A particular form of the LQ problem that arises in many control system problems is that of the *Linear Quadratic Regulator* (*LQR*) where all of the matrices are constant, the initial time is set to zero, and the terminal time is taken in the limit of infinity (this last assumption is what is known as *infinite horizon*).

For a continuous-time linear system described by $\dot{x} = Ax + Bu$ with an infinite horizon quadratic cost function defined as $J = \int_0^\infty \left(x^T Q x + u^T R u\right) dt$, where Q and R are the state and control weighting matrices respectively, the feedback control law that minimizes the value of the cost is determined by Eq. (13).

$$u = -Kx \tag{13}$$

where

- K is the gain matrix, defined as $K = R^{-1}B^T S$.
- S is the positive definite (or positive semi-definite) solution of the *Continuous-time Algebraic Riccati Equation* (*CARE*), solved from $0 = SA + A^T S - SBR^{-1}B^T S + Q$.

4. Simulations and Experiments

The bicycle robot is designed and built as shown in Fig. 3.

Fig. 3 Bicycle robot

The robot is modified from a 26" mountain bicycle by installing a 24V 750W DC geared motor for mass-moving, a 24V 60W DC geared motor for steering, a 24V 350W brushless DC motor for forward driving, and their transmission mechanisms. Incremental rotary encoders are also mounted on each motor to measure their angles and further process their speeds. A MicroStrain 3DM-GX1 gyro sensor is used to measure the bicycle's lean angle. An ARM Cortex-M3 based microcontroller is selected as the main controller of the system. All system parameters are listed in Table. 1.

Table. 1 System parameters

Component	Parameter	Value	Unit
Mechanical parts	m_1	46.00	kg
	m_2	6.00	kg
	h	0.50	m
	l_1	0.65	m
	l_2	0.43	m
	I_1	11.50	kg·m^2
	I_2	1.11	kg·m^2
	a	0.49	m
	b	1.09	m
	λ	1.26	rad
Pendulum motor	k_g	15.60	-
	k_b	0.07	V/(rad/s)
	k_τ	0.07	Nm/A
	R	0.12	Ω
Steering motor	k_1	16.76	1/s
	k_2	0.45	(rad/s^2)/V
Driving motor	v	1.50	m/s

In order to evaluate the performance of the robot balancing control, MATLAB/Simulink is used to simulate the system control. All parameters in Table. 1 are substituted into the completed linearized dynamics model in the state-space form in Eq. (12), then the systems without and with mass-moving are tested by applying the LQR with different initial lean angles.

The weighting matrices Q and R are defined in such a way that the lean angle is prioritized compared with other states, and the balancing load is properly shared by both balancing motors. These weighting matrices are shown as follows.

$$Q = \begin{bmatrix} 10^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

DRC0007

$$R = \begin{bmatrix} 10^2 & 0 \\ 0 & 1 \end{bmatrix}$$

The simulation results are shown in Figs. 4 – 6.

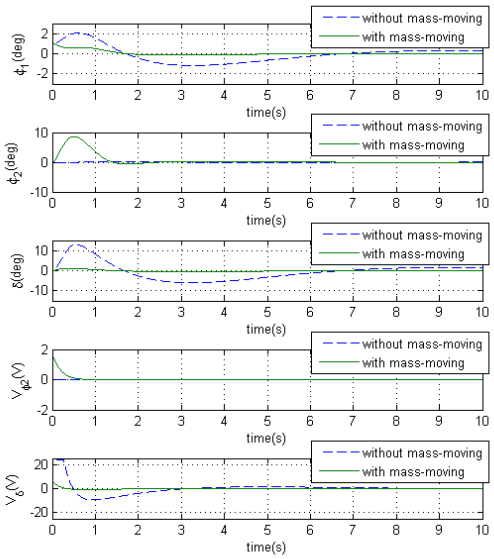

Fig. 4 Simulation results of robot balancing control in cases of without and with mass-moving when lean angle initialized as 1°

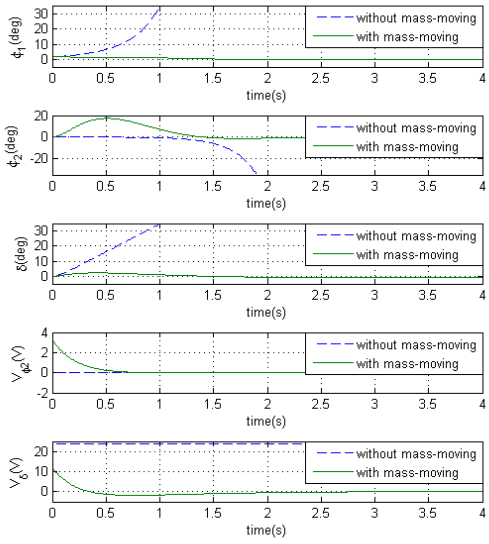

Fig. 5 Simulation results of robot balancing control in cases of without and with mass-moving when lean angle initialized as 2°

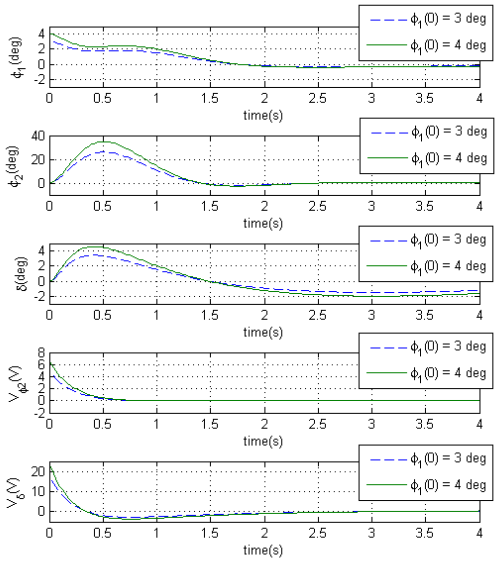

Fig. 6 Simulation results of robot balancing control with mass-moving when lean angles initialized as 3° and 4°

When the lean angle is initialized as 1°, it is observed from Fig. 4 that, in the case of balancing without mass-moving, the control signal of the steering motor starts at a maximum level of 24V, resulting in the maximum steering angle being increased to 12.9° at the time of 0.5 second, then both control signal and steering angle gradually decrease to zero after the bicycle posture returns to the upright position. The control signal of the pendulum motor shows at 0V because the system has no a mass-moving control. This makes the bicycle posture leans a bit more from 1° to 2.1° at the time of 0.6 second and then gradually converges to the upright position within 6 seconds. Compared to the case of balancing with mass-moving, the control signal of the pendulum motor starts at 1.6V, resulting in the maximum pendulum angle being increased to 8.7° at the time of 0.5 second. The control signal of the steering motor starts at 6V, resulting in the maximum steering angle being increased to 1.1° at the time of 0.4 second. Then control signals as well as pendulum and steering angles gradually decrease to zero after the bicycle posture returns to the upright position. This makes the bicycle posture suddenly leans back from 1° to 0.6° at the time of 0.4 second and then reaches the upright position within 1.5 seconds.

When the lean angle is initialized as 2°, it is observed from Fig. 5 that, in the case of balancing without mass-moving, the control signal of the steering motor starts at the maximum level of 24V to increase the steering angle with its full effort. However by using this system design and forward speed, the steering angle cannot reach the required position in proper time, resulting in the anti-falling torque being too small, so the bicycle starts to fall down ($\phi_1 > 30^\circ$) within 1 second. Compared to the case of balancing

with mass-moving, the control signal of the pendulum motor starts at 3.3V, resulting in the maximum pendulum angle being increased to 17.3° at the time of 0.5 second, while the control signal of the steering motor starts at 12V, resulting in the maximum steering angle being increased to 2.3° at the time of 0.4 second. Then control signals as well as pendulum and steering angles gradually decrease to zero after the bicycle posture returns to the upright position. This makes the bicycle posture leans back from 2° to the upright position within 1.5 seconds.

Balancing with mass-moving when lean angles initialized as 3° and 4° is shown in Fig. 6. It is observed that the results show similar trend to ones with smaller initial lean angles described above. The required motor control signals are within 24V operating range. The steering angles are within 5° range. The bicycle posture returns from initial lean angles to the upright position within 1.5 seconds. However, the maximum pendulum angle of balancing with the 4° initial lean angle is increased to 34.6°, which is greater than the allowed mechanical limit of the designed robot and may not conform to linearization of the system model. So 3° initial lean angle should be considered as the maximum lean angle which this balancing method can be achieved.

In real experiments, the balancing performance of the developed bicycle robot is tested while moving forward on a 50-meter long straight road. The lean angle is initialized at the upright position and the balancing control is randomly disturbed by imperfections of the road surface. The experimental results in cases of without and with mass-moving are compared as shown in Figs. 7 and 8.

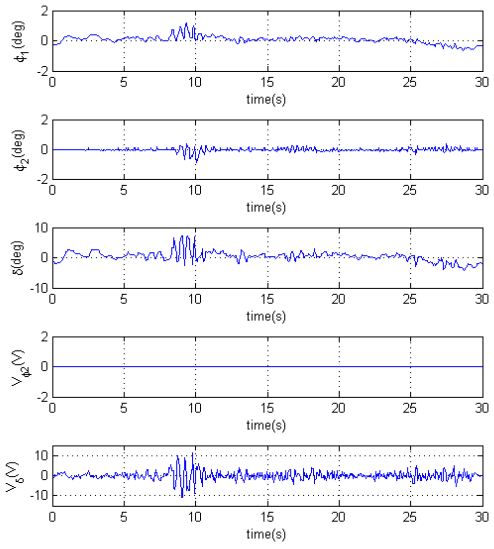

Fig. 7 Experimental results of robot balancing control without mass-moving

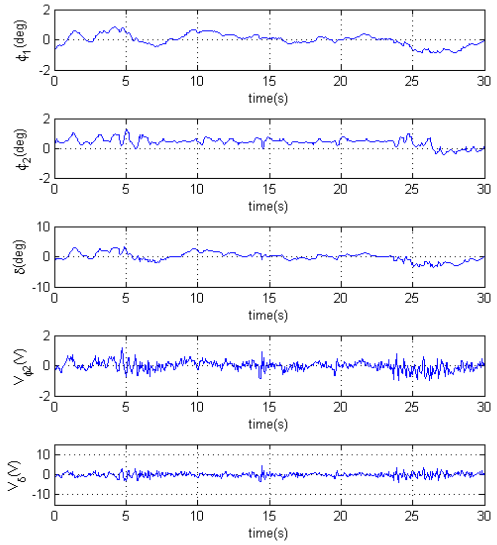

Fig. 8 Experimental results of robot balancing control with mass-moving

In case of balancing without mass-moving, it is observed from Fig. 7 that the control signal of the steering motor varies up to ±11.3V range, resulting in the steering angle being moved up to ±7.3° range. The control signal of the pendulum motor shows at 0V because the system has no a mass-moving control, resulting in the pendulum angle being freely moved. This makes the bicycle posture leans up to ±1.2° range.

Compared to the case of balancing with mass-moving, it is observed from Fig. 8 that the control signal of the steering motor varies within ±4.6V range, resulting in the steering angle being moved within ±3.7° range. The control signal of the pendulum motor varies within ±1.2V range, resulting in the pendulum angle being moved within ±1.2° range. This makes the bicycle posture leans within ±0.9° range.

The simulation and experimental results are consistent and show that the system using both steering and mass-moving in balancing obtains better performance in terms of leaning range and balancing time than the system using only steering in balancing.

5. Conclusion

This paper presented a control method for balancing a bicycle robot at a constant forward speed by integration of mass-moving and steering. To keep the robot in the upright position, the downward pendulum was moved by controlling its motor torque to keep the combined COG of the system on the line between both wheel-ground contact points, while the steering assembly was also moved by controlling its motor position to generate a centrifugal force which is considered as the anti-falling torque. The completed linearized dynamics model was derived by using the Euler-Lagrange equation of motion, linearizing around the upright position, and then combining with DC

motor model. The LQR was used to control its balance. MATLAB/Simulink was used to evaluate control performance with different initial lean angles and to compare the balancing capability of the systems without/with mass-moving. The simulation results showed that the system using both steering and mass-moving in balancing obtained better performance in terms of leaning range and balancing time than the system using only steering in balancing. The bicycle robot was already designed and built. Real experimental results were also in line with the simulation results.

6. References

[1] Murata Manufacturing Co., Ltd. (2005). *MURATA BOY*, URL: http://www.murata.com/about/mboymgirl/mboy, accessed on 15/08/2016

[2] Beznos, A.V., Formal'sky, A.M., et al. (1998). Control of Autonomous Motion of Two-Wheel Bicycle with Gyroscopic Stabilisation, paper presented in *the 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium.

[3] Thanh, B.T. and Parnichkun, M. (2008). Balancing Control of Bicyrobo by Particle Swarm Optimization-Based Structure-Specified Mixed H_2/H_∞ Control, *International Journal of Advanced Robotic Systems*, vol. 5(4), November 2008, pp. 395 – 402.

[4] Getz, N.H. and Marsden, J.E. (1995). Control for an Autonomous Bicycle, paper presented in *the 1995 IEEE International Conference on Robotics and Automation*, Nagoya, Japan.

[5] Tanaka, Y. and Murakami, T. (2004). Self Sustaining Bicycle Robot with steering controller, paper presented in *the 8th IEEE International Workshop on Advanced Motion Control*, Kawasaki, Japan.

[6] Vatanashevanopakorn, S. and Parnichkun, M. (2011). Steering Control based Balancing of a Bicycle Robot, paper presented in *the 2011 IEEE International Conference on Robotics and Biomimetics*, Phuket, Thailand.

[7] Yamakita, M. and Utano, A. (2005). Automatic Control of Bicycles with a Balancer, paper presented in *the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey, USA.

[8] Yamakita, M., Utano, A., and Sekiguchi, K. (2006). Experimental Study of Automatic Control of Bicycle with Balancer, paper presented in *the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China.

[9] Seekhao, P., Tungpimolrut, K., and Parnichkun, M. (2012). Balancing Control of Bicycle Robot Using Mass-Moving Stabilization, paper presented in *the 2012 TRS Conference on Robotics and Industrial Technology*, Nakhon Pathom, Thailand.

[10] Parnichkun, M. (2008). Optimal Control Theory, *Lecture Notes in Control Theory*. Asian Institute of Technology, Thailand.