KYBERNETIKA — VOLUME *51* (2015), NUMBER 1, PAGES 173–191

LQR AND MPC CONTROLLER DESIGN AND COMPARISON FOR A STATIONARY SELF-BALANCING BICYCLE ROBOT WITH A REACTION WHEEL

Kiattisin Kanjanawanishkul

A self-balancing bicycle robot based on the concept of an inverted pendulum is an unstable and nonlinear system. To stabilize the system in this work, the following three main components are required, i. e., (1) an IMU sensor that detects the tilt angle of the bicycle robot, (2) a controller that is used to control motion of a reaction wheel, and (3) a reaction wheel that is employed to produce reactionary torque to balance the bicycle robot. In this paper, we propose three control strategies: linear quadratic regulator (LQR), linear model predictive control (LMPC), and nonlinear model predictive control (NMPC). Several simulation tests have been conducted in order to show that our proposed control laws can achieve stabilizaton and make the system balance. Furthermore, LMPC and NMPC controllers can deal with state and input constraints explicitly.

Keywords: self-balancing bicycle robot, linear quadratic regulator, model predictive control

Classification: 49N05, 93C85

1. INTRODUCTION

A self-balancing bicycle robot is a system based on the concept of an inverted pendulum system, where its upright body is pivoted on two wheels. Without a proper control scheme, the system itself cannot be balanced and keeps falling off. Therefore, it is naturally an unstable system and its nonlinearity makes the problem of controlling it difficult, resulting in a challenging research topic to the control engineering community.

Over the past few decades, researchers have explored different solutions to dynamically balance a bicycle robot. Their ideas can be classified into four groups as follows:

- Gyroscopic stabilization, where one or more motorized gimbals tilt the angular momentum of a spinning rotor [3]. As the rotor tilts, the changing angular momentum causes a gyroscopic precessive torque that balances the bicycle. Advantages of this system are that it can produce large amount of torque, it has no ground reaction forces, and the system can be stable even when the bicycle is stationary. Disadvantages are that it consumes more energy and it is physically complex. Research

DOI: 10.14736/kyb-2015-1-0173

studies using this concept include Beznos et al. in [2], Gallaspy in [5], and Bui and Parnichkun in [3], while a remarkable example of applying this idea is a self-balancing electric motorcycle from Lit Motors Inc., where production will begin by the end of 2014.

- Steering control, where a controller controls the amount of torque applied to the steering handlebar to balance the bicycle. Advantages of this system are low mass and low energy consumption, while its disadvantages are that it requires ground reaction forces and it cannot withstand large tilt angle disturbance. Its mathematical model is also complex, however, several different dynamic models and control schemes have been proposed in the literature, such as Tanaka and Murakami in [13], Yi et al. in [14], Defoort and Murakami in [4], Lei et al. in [9], and Pongpaew in [11].

- Reaction wheel control, where speed of a reaction wheel is increased or decreased to generate a reactionary torque about the spin axis which is parallel to the bicycle's frame. As the bicycle begins to fall to one side, a motor mounted to the reaction wheel applies a torque on the reaction wheel, generating a reactionary torque on the bicycle, which brings back the bicycle's balance. Advantages of this system are that it is low cost, simple and no ground reaction, while disadvantages are that it consumes more energy and it cannot produce large amount of torque. A very well-known self-balancing bicycle robot using a reaciton wheel is the Murata Boy which was developed by Murata Manufacturing Co., Ltd in 2005.

- Mass Balancing, where the center of gravity or a mass balancer is controlled. Studies using this idea are Lee and Ham in [8], Keo and Masaki in [6].

Furthermore, Keo and Yamakita [7] derived a controller using a steering handlebar and a balancer torque to stabilize the bicycle. Numerical simulation results were shown that balancing control using both the steering and the balancer has a better performance than conventional ones with only balancer or steering.

However, in the literature, a PID controller has been most commonly used for achieving balance of the bicycle robot with a reaction wheel. Hence, in this study, we propose control algorithms for stabilization at zero forward speed by using linear quadratic regulator (LQR), linear model predictive control (LMPC), and nonlinear model predictive control (NMPC). They belong to an optimal control scheme where a control law is determined for a given system such that a certain optimality criterion is achieved. They can be applied for multivariable systems in a state-space form. Furthermore, constraints of the control input can be handled explicitly in case of using model predictive control (MPC).

In Section 2, we present a simplified dynamic model of the bicycle robot with a reaction wheel based on the concept of an inverted pendulum system from a Lagrange equation. Then, LQR, LMPC, and NMPC control laws are introduced in Section 3 for stabilization. In Section 4, numerical simulation tests are conducted. Several simulation scenarios with our proposed control laws are evaluated in detail and outputs are compared. The results are shown to verify the effectiveness of our proposed control strategy. Finally, our conclusions and future work are given in Section 5.

2. DESIGN AND MODELLING OF A SELF-BALANCING BICYCLE ROBOT WITH A REACTION WHEEL

The ultimate goal of our long-term project is to build a bicycle robot that can balance itself using a reaction wheel. This bicycle robot has to be able to keep balancing at zero forward speed and to follow a given path without losing its balance at desired forward speed.

The basic principle is that the reaction wheel attached to an electric motor can rotate clockwise or counterclockwise with desired speed in order to maintain balancing of the bicycle. Due to the fact that the motor applies a torque on the reaction wheel, which in turn applies an equal amount of torque back on the bicycle. With a proper control law, this action-reaction combination can balance the bicycle.

2.1. Hardware and electronic design

Figure 1 shows a block diagram of the electronic system of our bicycle robot. There are four state variables, i. e., the tilt angle and its first derivative obtained from an accelerometer-gyro-magnetometer sensor called a MinIMU-9 sensor, and the reaction wheel angle and its first derivative received from an encoder mounted at the motor shaft.

In our closed-loop control system, the output of the bicycle robot is the tilt angle, which is the instantaneous angle of the robot base with respect to the vertical upright position. Sensor data from the MinIMU-9 sensor are transmitted to the microcontroller via I2C communicaton. The received data are processed through a complementary filter algorithm in order to obtain a precise tilt angle. We determine the error, which is the difference between the desired tilt angle and the actual tilt angle. It is fed into our proposed controller so that the controller will process, calculate, and generate the corresponding motor voltage to control the DC motor (that, in turn, controls the reaction wheel) via the motor controller, in order to achieve balancing at the upright position.

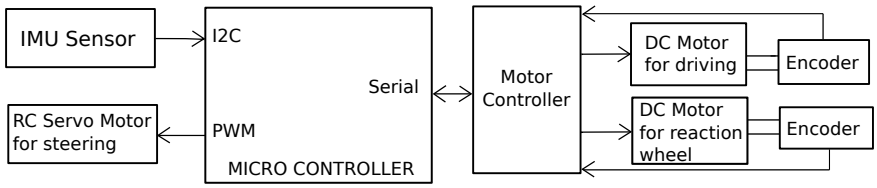
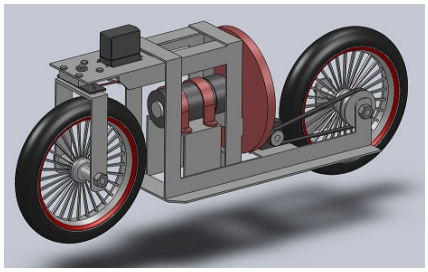


Fig. 1: A block diagram of the electronic system.

Figure 2 shows the mechanical structure of the self-balancing bicycle system. It consists of three motors: a metalgear DC motor with an encoder connected to the rear wheel, providing propulsion; a metalgear DC motor with an encoder connected to the reaction wheel, providing balance; and an RC servo motor connected to the steer, providing directional steering. However, the RC servo motor for steering and the DC motor for driving will be taken into account in the future work.



(a)

(b)

Fig. 2: The bicycle robot's mechanical frame design: (a) SolidWorks design drawing showing layout of major components and (b) a photograph of our prototype bicycle robot, dimensions approximately $(W \times L \times H) : 175 \times 1083 \times 453$ mm.

Since the rotation of the reaction wheel located at the center of the bicycle is the key of the balancing and energy is stored in the rotor as kinetic energy, a compromise between mass vs. moment of inertia of the wheel is very important. In order to minimize the overall mass, while attaining a high moment of inertia, the majority of its mass has to be concentrated at the outer edge of the wheel.

2.2. Mathematical model

Since our system is based on the principle of an inverted pendulum with a reaction wheel as seen in Figure 3, by disregarding forces generated from moving forward and steering, a simplified dynamic model of the bicycle robot can be derived using a Lagrange method. Let m_1, o_1, I_1, θ and L_1 be the bicycle mass, its center of mass, its moment of inertia about its center of mass, the angle between the bicycle and the vertical upright direction and the distance from the origin O to the center of mass of the bicycle, respectively and let m_2, o_2, I_2, ϕ and L_2 be the reaction wheel mass, its center of mass, its moment of inertia about the center of mass of the reaction wheel, the rotation angle of the reaction wheel, and the distance from its center of mass to the origin O, respectively, then a Lagrange equation of the dynamic model can be derived by using:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i \tag{1}$$

where $\tau_i, i = 1, 2, 3, \ldots, n$, denotes the external force corresponding to generalized coordinates q_i, and

$$L(q, \dot{q}) = KE(q, \dot{q}) - PE(q, \dot{q}) \tag{2}$$

where L, KE, PE, and q are the Lagrangian operator, the kinetic energy of the system, the potential energy of the system, and the generalized coordinates of the system, respectively. In this paper, θ and ϕ are considered as system generalized coordinates.

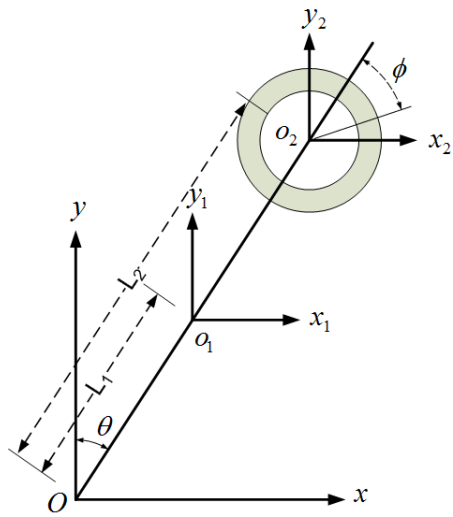


Fig. 3: A model of a bicycle robot with a reaction wheel system based on the principle of an inverted pendulum system.

The total kinetic energy of the system is written as

$$KE = \frac{1}{2}(m_1L_1^2 + m_2L_2^2 + I_1 + I_2)\dot{\theta}^2 + I_2\dot{\theta}\dot{\phi} + \frac{1}{2}I_2\dot{\phi}^2 \tag{3}$$

The potential energy of the bicycle body and of the reaction wheel are obtained as follows:

$$PE = (m_1L_1 + m_2L_2)g\cos\theta \tag{4}$$

After that, the Lagrangian operator can be obtained as follows:

$$\begin{aligned} L =& KE - PE \\ =& \frac{1}{2}(m_1L_1^2 + m_2L_2^2 + I_1 + I_2)\dot{\theta}^2 + I_2\dot{\theta}\dot{\phi} + \frac{1}{2}I_2\dot{\phi}^2 - (m_1L_1 + m_2L_2)g\cos\theta. \end{aligned} \tag{5}$$

Using the Lagrange equation defined in (1), the following two equations are derived

$$\begin{aligned} (m_1L_1^2 + m_2L_2^2 + I_1 + I_2)\ddot{\theta} + I_2\ddot{\phi} - (m_1L_1 + m_2L_2)g\sin\theta = 0 \\ I_2(\ddot{\theta} + \ddot{\phi}) = T_r \end{aligned} \tag{6}$$

where T_r is a driving torque of the reaction wheel as friction forces and electrical dynamics of the DC motor are neglected.

The equation above is a nonlinear mathematical model expression of the inverted pendulum system with a reaction wheel. In order to simplify analysis and computation, when θ is very small, according to Jacobian linearization, we obtain:

$$\begin{aligned} (m_1L_1^2 + m_2L_2^2 + I_1 + I_2)\ddot{\theta} + I_2\ddot{\phi} - (m_1L_1 + m_2L_2)g\theta = 0 \\ I_2(\ddot{\theta} + \ddot{\phi}) = T_r. \end{aligned} \tag{7}$$

Let $\mathbf{x} = [x_1, x_2, x_3, x_4]^T = [\theta, \dot{\theta}, \phi, \dot{\phi}]^T$ be state variables of the system, then the state space expression of the system can be given as follows:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ b/a & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -b/a & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ -1/a \\ 0 \\ (a+I_2)/(aI_2) \end{bmatrix} T_r, \tag{8}$$

where $a = m_1 L_1^2 + m_2 L_2^2 + I_1, b = (m_1 L_1 + m_2 L_2)g$.

Next, to control the speed of the reaction wheel to the desired one, the following mathematical model is the physical behavior of the motor with a gear system:

$$\begin{aligned} V &= L_m \frac{\mathrm{d}i}{\mathrm{d}t} + R_m i + K_e \omega_m \\ T_m &= K_t i \\ T_r &= N_g T_m, \end{aligned} \tag{9}$$

where V is motor supply voltage, K_e is a motor back emf constant, ω_m is motor angular speed, L_m is armature coil inductance, R_m is armature coil resistance, i is armature current, T_m is motor generated torque, K_t is a motor torque constant, and N_g is the gear ratio.

The term of inductance can be neglected since, in general, the motor inductance value is much less than the motor resistance value ($L_m \ll R_m$). By using the relationship between the motor and the reaction wheel, i. e., $\dot{\phi} = \omega_r$, $\omega_m = N_g \omega_r$, where ω_r is reaction wheel angular speed, the motor supply voltage can be found in terms of the reaction wheel angle as follows:

$$T_r = N_g K_t \big(\frac{V - K_e N_g \dot{\phi}}{R_m} \big) \tag{10}$$

Then, the mathematical model of the motor (10) is combined with the mathematical model of the bicycle (8) in state space form:

$$\begin{aligned} \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{21} & 0 & 0 & a_{24} \\ 0 & 0 & 0 & 1 \\ a_{41} & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b_4 \end{bmatrix} V \\ y &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix}, \end{aligned} \tag{11}$$

where

$$\begin{aligned} a_{21} &= \frac{b}{a},\ a_{24} = \frac{K_t K_e N_g^2}{a R_m},\ a_{41} = -\frac{b}{a},\ a_{44} = -\big(\frac{a+I_2}{aI_2}\big)\big(\frac{K_t K_e N_g^2}{R_m}\big), \\ b_2 &= -\frac{K_t N_g}{a R_m},\ b_4 = \big(\frac{a+I_2}{aI2}\big)\frac{K_t N_g}{R_m}. \end{aligned}$$

Parameters	Values
Bicycle c.g. upright height (L_1)	0.25 m
Reaction wheel c.g. upright height (L_2)	0.35 m
Mass of bicycle (m_1)	20.1 kg
Mass of reaction wheel (m_2)	3.7 kg
Bicycle moment of inertia about c.g. (I_1)	1.504 kgm^2
Reaction wheel moment of inertia about c.g. (I_2)	0.052 kgm^2
Motor resistance (R_m)	0.71 Ω
Motor torque constant (K_t)	0.0229 Nm/A
Motor back emf constant (K_e)	0.0229 Vs/rad
Motor gear ratio (N_g)	25 : 1
Gravitational acceleration (g)	9.81 $\mathrm{m/s}^2$

Tab. 1: System parameters.

Finally, by substituting the parameter values in Table 1, the state space expression of the system is changed into:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 22.4350 & 0 & 0 & 0.1670 \\ 0 & 0 & 0 & 1 \\ -22.4350 & 0 & 0 & -9.0445 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ -0.2918 \\ 0 \\ 15.7983 \end{bmatrix} V. \tag{12}$$

According to control theory, the characteristic equation of the system is given by $\mathbf{det}(\lambda I - A) = 0$, resulting in four eigenvalues: $\lambda_1 = 0,\ \lambda_2 = 4.7077,\ \lambda_3 = -4.6458,\ \lambda_4 = -9.1065$. Since one of eigenvalues is located in the right half-plane of complex frequency domain, the system is unstable. Furthermore, the controlability matrix of the system $CO = [B, AB, A^2B, A^3B]$ and $\mathbf{rank}(CO) = 4$, therefore the system is controllable.

3. CONTROL ALGORITHM

As mentioned previously, the self-balancing bicycle robot is an unstable system, therefore, a proper control law is required to balance the system. In this paper, to maintain balancing at the zero forward speed, three different types of controllers, i. e., LQR, LMPC, and NMPC, are designed and evaluated. For these model-based controllers, the state space model obtained by linearization (12) is used in LQR and LMPC design process, while the nonlinear model (6) and (10) are used in the NMPC strategy.

Although the cost function often used in MPC is similar to the cost in LQR, the main difference between MPC and LQR is that MPC is formulated as the repeated solution of a finite horizon open-loop optimal control problem subject to the predicted system behavior and input and state constraints. This is an important advantage of this type of control.

3.1. Linear Quadratic Regulator (LQR)

An LQR controller is a solution to the linear quadratic (LQ) problem where a set of linear differential equations and a quadratic objective function are defined. It can stabilize the system by changing the location of poles of the system to the optimal location since time response, overshoot and steady state depend on the location of poles. It also has simpler structure than MPC, for example, only instant input and output variables are penalized by weight matrices, while variation of the input and future values of inputs/outputs are not penalized.

For a continuous time system described by

$$\begin{aligned} \dot{\mathbf{x}} &= A_c\mathbf{x} + B_c\mathbf{u} \\ \mathbf{y} &= C_c\mathbf{x} + D_c\mathbf{u}, \end{aligned} \tag{13}$$

the LQR controller is basically an algorithm that finds an optimal gain matrix (K) for a state-feedback controller based on weighting factors (Q, R and M) of minimization of the following quadratic objective function:

$$J(\mathbf{u}) = \int_0^\infty (\mathbf{x}^T Q\mathbf{x} + \mathbf{u}^T R\mathbf{u} + 2\mathbf{x}^T M\mathbf{u})\,\mathrm{d}t, \tag{14}$$

where Q is a symmetric positive semi-definite matrix, R is a symmetric positive definite matrix, and M is the cross term that relates $\mathbf{u}$ to $\mathbf{x}$ in the objective funtion. Then, the gain matrix K can be calculated by the following equation:

$$K = R^{-1}(B^T S + M^T), \tag{15}$$

where S is found by solving the continuous time Riccati equation:

$$A^T S + SA - (SB + M)R^{-1}(B^T S + M^T) + Q = 0. \tag{16}$$

With this optimal gain matrix, a control law can be defined as $\mathbf{u} = -K\mathbf{x}$.

However, a disadvantage of the LQR is that the controller requires full state feedback, which in most cases are unavailable. An observer used to estimate the unmeasured states will be implemented in our future work.

3.2. Linear MPC (LMPC)

Model predictive control (MPC) is one of the most advanced control methods that can deal with constraints on inputs and states explicitly. It is very important since almost all physical systems have constraints such as limitations on the input and output signals. If such constraints are not taken into account in the design of control systems, performance degradation or, at worst, instability might happen. Although MPC is apparently not a new control method, research studies dealing with MPC of a self-balancing bicycle robot are rare.

The conceptual structure of MPC is depicted in Figure 4. The basic principle of this technique [1] is to take a sample from sensors and then future control inputs and future plant responses are predicted using a system model for the next N time steps

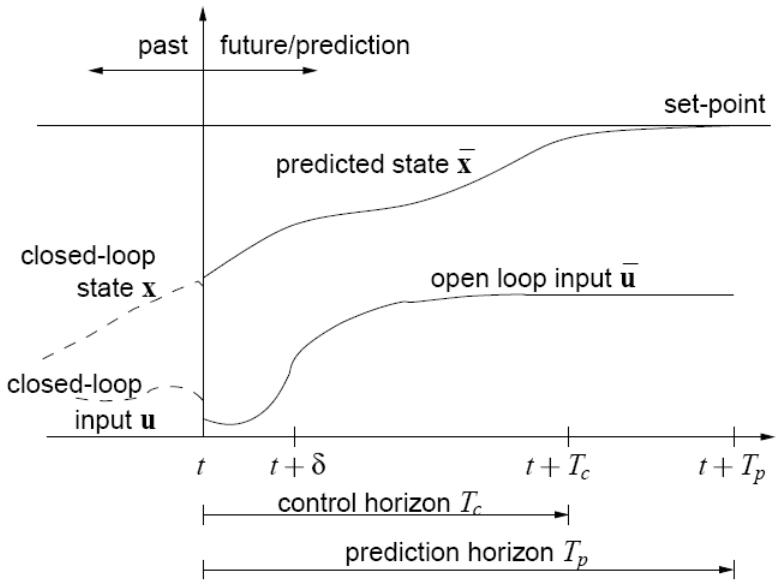


Fig. 4: Principle of model predictive control [1].

called a horizon. From that, a control law is determined for every time step by solving a constrained optimization problem and only the first one is implemented. This process is then repeated for the next time step.

A linear model is used for controller design in this subsection, while a nonlinear version will be considered in the next subsection for NMPC design. Therefore, the linearized system (12) can be written in a discrete state space system as follows:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k) \tag{17}$$

where $A \in \mathbb{R}^n \times \mathbb{R}^n$, n is the number of the state variables and $B \in \mathbb{R}^n \times \mathbb{R}^m$, m is the number of input variables. The discrete matrices A and B can be obtained as follows:

$$A = I + \delta A_c, \quad B \quad = \delta B_c \tag{18}$$

where $\dot{\mathbf{x}} = A_c\mathbf{x} + B_c\mathbf{u}$ is a continuous time system and δ is a sampling time.

The goal is to find the control-variable values that minimize the quadratic objective function. With the linear system, the optimization problem can be transformed into a QP problem. Since it turns into a convex problem, solving the QP problem results in global optimal solutions. After solving the QP problem at each time instant, control-variable values are obtained.

To make the LMPC objective function equivalent to the LQR objective function [12],

the quadratic objective function with a prediction horizon N is given by

$$\begin{aligned} J(k) = &\sum_{j=0}^{N-1} \{\mathbf{x}^T(k+j|k)Q\mathbf{x}(k+j|k) + \mathbf{u}^T(k+j|k)R\mathbf{u}(k+j|k)\} \\ &+ \mathbf{x}^T(k+N|k)Q_N\mathbf{x}(k+N|k) \end{aligned} \tag{19}$$

where $Q \in \mathbb{R}^n \times \mathbb{R}^n$ and $R \in \mathbb{R}^m \times \mathbb{R}^m$ are the penalty weighting matrices, with $Q \succeq 0$ and $R \succ 0$. Q_N is a penalty applied at the terminal prediction horizon step. The double subscript notation $(k+j|k)$ denotes the prediction made at time k of a value at time $k+j$.

By introducing matrices such that all $\mathbf{u}(\cdot)$ and $\mathbf{x}(\cdot)$ are conveniently stored, we obtain a more compact expression that a QP solver can solve. Defining the prediction state vector

$$X(k) = [\mathbf{x}^T(k+1|k), \mathbf{x}^T(k+2|k), \ldots, \mathbf{x}^T(k+N|k)]^T$$

where $X \in \mathbb{R}^{n \cdot N}$ and the control input vector

$$U(k) = [\mathbf{u}^T(k|k), \mathbf{u}^T(k+1|k), \ldots, \mathbf{u}^T(k+N-1|k)]^T$$

where $U \in \mathbb{R}^{m \cdot N}$, it can be shown that

$$X(k) = G(k)\mathbf{x}(k|k) + S(k)U(k) \tag{20}$$

where $G(k) \in \mathbb{R}^{n \cdot N} \times \mathbb{R}^{n \cdot N}$ and $S(k) \in \mathbb{R}^{n \cdot N} \times \mathbb{R}^{m \cdot N}$ are defined as follows

$$G(k) = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} \quad \text{and} \quad S(k) = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}.$$

After defining $\bar{Q} \in \mathbb{R}^{n \cdot N} \times \mathbb{R}^{n \cdot N}$ as a block diagonal matrix containing Q repeated $N-1$ times and Q_N as follows

$$\bar{Q} = \begin{bmatrix} Q & 0 & 0 & \cdots & 0 \\ 0 & Q & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & Q & 0 \\ 0 & 0 & \cdots & 0 & Q_N \end{bmatrix}$$

and $\bar{R} \in \mathbb{R}^{m \cdot N} \times \mathbb{R}^{m \cdot N}$ as a block diagonal matrix containing R repeated N times, the objective function of the QP problem can then be rewritten

$$\begin{aligned} J(k) &= \mathbf{x}^T(k|k)Q\mathbf{x}(k|k) + X^T\bar{Q}X + U^T\bar{R}U \\ &= \mathbf{x}^T(k|k)Q\mathbf{x}(k|k) + (G\mathbf{x}(k|k) + SU)^T\bar{Q}(G\mathbf{x}(k|k) + SU) + U^T\bar{R}U \\ &= \frac{1}{2}U^T 2(\bar{R} + S^T\bar{Q}S)U + \mathbf{x}^T(k|k)2G^T\bar{Q}SU + \frac{1}{2}\mathbf{x}^T(k|k)2(Q + G^T\bar{Q}G)\mathbf{x}(k|k). \end{aligned} \tag{21}$$

After some algebraic manipulations, we can rewrite the objective function (19) in a standard quadratic form:

$$\bar{J}(k) = \frac{1}{2}U^T H(k)U + \mathbf{x}^T(k|k)F(k)U + \frac{1}{2}\mathbf{x}^T(k|k)Y(k)\mathbf{x}(k|k) \tag{22}$$

with

$$\begin{aligned} H(k) &= 2(\bar{R} + S^T\bar{Q}S) \\ F(k) &= 2G^T\bar{Q}S \\ Y(k) &= 2(Q + G^T\bar{Q}G) \end{aligned}$$

where $H \in \mathbb{R}^{m\cdot N} \times \mathbb{R}^{m\cdot N}$ is a Hessian matrix which is always positive definite. It describes the quadratic part of the objective function. $F \in \mathbb{R}^{m\cdot N}$ is a vector which describes the linear part of the objective function.

To handle the input constraints, we consider the existence of bounds in the amplitude of the control variables:

$$\mathbf{u}_{\min} \leq \mathbf{u}(k+j|k) \leq \mathbf{u}_{\max} \tag{23}$$

where $j \in [0,\ N-1]$, and $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ denote the lower and upper bounds, respectively. In this paper, the system is driven by a DC motor, therefore we consider the following constraints:

$$V_{\min} \quad \leq \quad V \quad \leq \quad V_{\max} \tag{24}$$

where $V_{\min}$ and $V_{\max}$ are the lower and upper boundaries of the motor voltage, respectively.

Thus, with the standard expression of the QP problem, the optimization problem to be solved at each sampling time is stated as follows:

$$\begin{aligned} U^\star = \arg\ \min_U \{&\frac{1}{2}U^T H(k)U + \mathbf{x}^T(k|k)F(k)U\} \\ \text{subject to} \quad &\mathbf{u}_{\min} \leq \mathbf{u}(k+j|k) \leq \mathbf{u}_{\max}. \end{aligned} \tag{25}$$

Note that $\frac{1}{2}\mathbf{x}^T(k|k)Y(k)\mathbf{x}(k|k)$ is constant, therefore it is excluded in (25).

After the QP problem at time t_k is solved, an optimal control sequence is generated. The motor voltage input for a reaction wheel can be obtained from the first element of this sequence and is then applied to the system.

3.3. Nonlinear MPC (NMPC)

Although LMPC theory is well studied by now, LMPC may not be suitable for some nonlinear systems. Therefore, nonlinear models must be used [1]. In general, the main difference between LMPC and NMPC is that LMPC refers to an MPC scheme in which linear models are used to predict the system dynamics and linear constraints on the states and inputs are considered. NMPC refers to an MPC scheme that is based on nonlinear models and/or considers a non-quadratic cost function and nonlinear constraints [1].

A nonlinear system is normally described by the following nonlinear differential equation:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \text{subject to:} \quad &\mathbf{x}(t) \in \mathcal{X},\ \mathbf{u}(t) \in \mathcal{U},\ \forall t \geq 0 \end{aligned} \tag{26}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$ are the n dimensional state vector and the m dimensional input vector of the system, respectively. $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$ denote the set of feasible states and inputs of the system, respectively. The input applied to the system is then given by the solution of the following finite horizon open-loop optimal control problem:

$$\min_{\mathbf{u}(\cdot)} \int_t^{t+T_p} \{\mathbf{x}^T(\tau) Q \mathbf{x}(\tau) + \mathbf{u}^T(\tau) R \mathbf{u}(\tau)\} \, \mathrm{d}\tau + V_t(\mathbf{x}(t+T_p)) \tag{27}$$

$$\begin{aligned} \text{subject to:} \quad & \dot{\mathbf{x}}(\tau) = \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau)) && \text{(28a)} \\ & \mathbf{u}(\tau) \in \mathcal{U} \quad \forall \tau \in [t, t+T_c] && \text{(28b)} \\ & \mathbf{x}(\tau) \in \mathcal{X} \quad \forall \tau \in [t, t+T_p] && \text{(28c)} \\ & \mathbf{x}(t+T_p) \in \mathcal{X}_f && \text{(28d)} \end{aligned}$$

where $\mathcal{X}_f$ is a terminal constraint set and V_t is a terminal cost.

Although, from a theoretical point of view, an infinite predictive control horizon can guarantee stability of a system, it may not be feasible for a nonlinear system in practice [1]. For this reason, finite prediction and control horizons are considered, however, the major concern is whether such a finite horizon NMPC strategy can guarantee stability of the closed-loop system. Mayne et al. [10] presented essential principles for the stability of NMPC for constrained nonlinear systems. Most of the approaches, that achieve closed-loop stability for NMPC, modify the NMPC setup such that stability of the closed-loop can be guaranteed. This is usually achieved by adding suitable equality or inequality terminal constraints and/or a suitable additional terminal cost to the cost function. Due to a large number of research papers concerning NMPC stability, we refer the reader to [1, 10]. In this work, the terminal cost, $V_t = x_N^T Q_N x_N$, corresponds to infinite horizon cost. It is used to ensure stability and feasibility. For more detail, the reader is referred to [10]. Furthermore, the constraints in (28b) denote the bounded control inputs given by (24).

Besides the stability issue, NMPC requires the repeated online solution of a nonlinear optimal control problem. In the case of LMPC, the solution of the optimal control problem can be cast as the solution of a QP problem which can be solved efficiently online. For NMPC, one has to solve a nonlinear program, which is in general computationally expensive. A commonly used approach to solve the problem is reformulation to a finite dimensional nonlinear programming problem (NLP) by a suitable parameterization. The most recent research in NMPC suggests to perform this parameterization by using direct multiple shooting method. The NLP can be solved by sequential quadratic programming (SQP) approach, which means solving a QP subproblem at each iteration. An updated estimate of the Hessian is computed in each iteration using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula. The QP subproblem is solved using an active set strategy and the gradients of the cost function and of equality and inequality constraint functions are computed using finite difference.

4. NUMERICAL SIMULATION

This section presents numerical simulation in a MATLAB environment where the bicycle robot with LQR, LMPC, and NMPC control laws are implemented and evaluated.

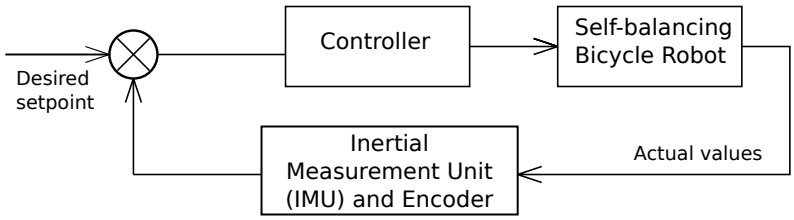


Fig. 5: A block diagram of the closed-loop control system.

Figure 5 shows a block diagram of the bicycle closed-loop control for stabilization by using different kinds of controllers. The four state variables including tilt angles and its angular speed, reaction wheel angles and its angular speed, are assumingly available without any disturbances and the reaction wheel is attached to a 25:1 DC motor operating at 12V DC. Therefore the motor voltage input is bounded to $V_{\min} = -12\,\mathrm{V}$ and $V_{\max} = 12\,\mathrm{V}$.

To show the validity and to evaluate the performance of the proposed control laws for bicycle stabilization at zero forward velocity, we perform several simulation scenarios with different control parameters and different initial tilt angles, while the initial values of the other state variables are zero. The desired goal of this study is to achieve the transient response and the stabilization of the bicycle robot back to the upright position.

4.1. LQR Controller

Selecting proper weight matrices Q and R are the most important part of designing an LQR controller. There is a trade-off between the small effort of control inputs and the controlled variables' error from the reference. In general, they are selected as diagonal matrix in order to simplify controller design process. Each diagonal value penalizes relevant state's or input's error. In the test, the following weighting parameters are first assigned: $Q = \mathrm{diag}(1000, 0, 0.001, 0)$ and $R = 10$. Then, the optimal gain matrix (K) determined by using (15) results in $K = [-435.5414, -92.4958, -0.0095, -1.1245]$. After that, the control law, $u = -K\mathbf{x}$, is obtained. Furthermore, to prevent the constraint violation, we impose the following saturation function for the motor voltage input (V):

$$V = \begin{cases} V_{\min} & \text{if} \quad V < V_{\min} \\ V & \text{if} \quad V_{\min} \leq V \leq V_{\max} \\ V_{\max} & \text{if} \quad V > V_{\max}. \end{cases} \tag{29}$$

The closed-loop system using the LQR controller is evaluated by setting the initial tilt angle to $1\,^\circ$, $2\,^\circ$, and $3\,^\circ$. System responses with these initial conditions due to the LQR controller are shown in Figure 6. As seen in the results, the bicycle robot can be stabilized back to the upright position.

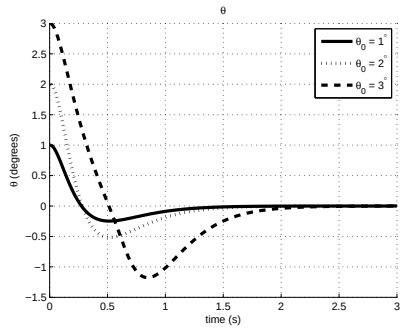
Next, weight matrices Q and R are varied. In Figure 7, the initial tilt angle is set to $3\,^\circ$, and G1, G2, and G3 are defined as follows:

G1 denotes $Q = \mathrm{diag}(1000, 0, 0.001, 0)$ and $R = 10$,

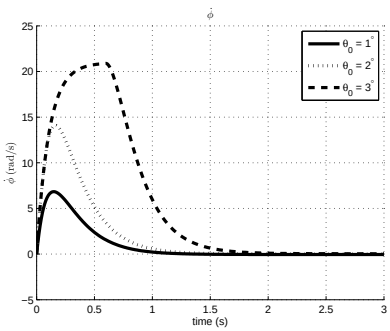
G2 denotes $Q = \mathrm{diag}(100, 100, 0.0001, 0)$ and $R = 0.01$,

G3 denotes $Q = \mathrm{diag}(1000, 1000, 0.0001, 0.01)$ and $R = 0.01$.

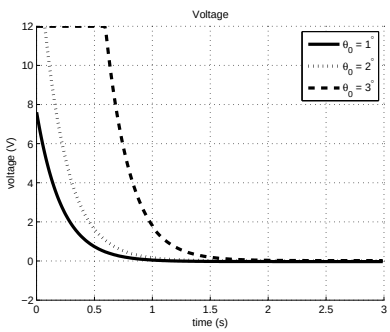
It is shown that higher overshoot results in shorter settling time.



(a)

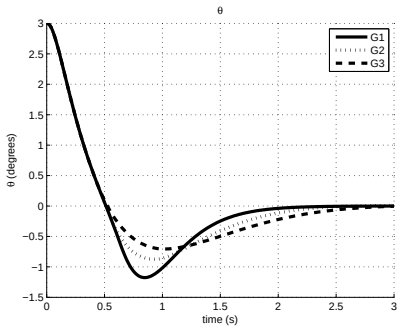


(b)

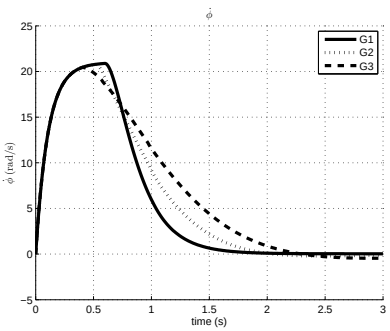


(c)

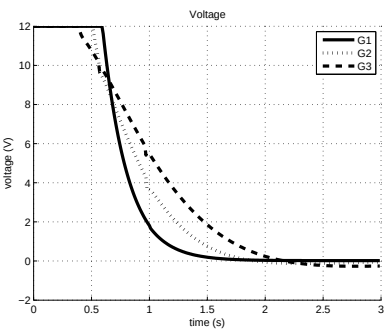
Fig. 6: The simulation results using the LQR control law when the initial tilt angles are set to $1\,^\circ$, $2\,^\circ$, and $3\,^\circ$, and $Q = \mathrm{diag}(1000, 0, 0.001, 0)$ and $R = 10$: (a) tilt angle, (b) reaction wheel angular speed, and (c) the motor voltage input for driving the reaction wheel.



(a)



(b)



(c)

Fig. 7: The simulation results using the LQR control law when the initial tilt angle is set to $3\,^\circ$: (a) tilt angle, (b) reaction wheel angular speed, and (c) the motor voltage input for driving the reaction wheel.

4.2. LMPC controller

In this paper, we select the following parameters for our LMPC scheme:
$Q = \mathrm{diag}(1000, 0, 0.001, 0)$, $R = 10$, $N = 5$, $T_c = T_p = 0.05$ s, $\delta = 0.01$ s, and Q_N is derived from an algebraic Riccati equation.

Figure 8 shows system responses due to the LMPC controller with different initial tilt angles. One of the most advantages of using LMPC is that constraints can be explicitly handled by an online optimization solver. As seen in Figure 8(c), the imposed input constraints (± 12 V) are satisfied.

4.3. NMPC controller

We select the same parameter values as the LMPC controller. Figure 9 shows system responses due to the NMPC controller with different initial tilt angles. Like LQR and LMPC, NMPC can stabilize the bicycle robot as expected and the boundaries of motor voltage input are not violated as seen in Figure 9(c).

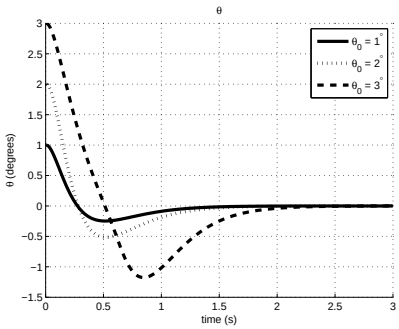
4.4. Comparison

In order to assess the performance of our three controller schemes, mean absolute error (MAE) computed for the controlled variable (i. e., tilt angle), absolute maximum overshoot, and settling time (using a band of ±2% of the total change in the controlled variables) are used as performance indicators. The comparison results of the LQR, LMPC, and NMPC controllers are illustrated in two cases, i. e., motor voltage input does not reach the boundary and reaches the boundary. In the former case, the initial tilt angle is set to 1 ° and the results are shown in Figure 10 and Table 2. In the latter case, the initial tilt angle is set to 3 ° and the results are shown in Figure 11 and Table 3. As seen in the results, the values of the performance indicators of all three controllers are similar and they can stabilize the bicycle robot with a reaction wheel effectively. The LQR controller has an advantage of fast computation, while the NMPC controller has an advantage of using a nonlinear model directly. Although the NMPC controller requires much more computational time than the others, it can handle complex behaviors and constraints of the system. Hence, it will be further investigated in the future.

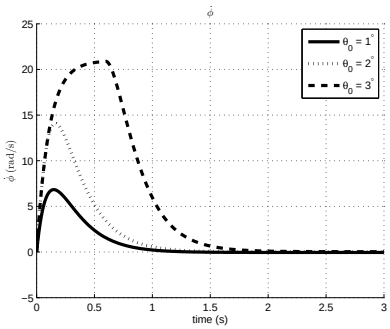
It has to be noted that the greater initial tilt angle is; the higher motor torque is required. But the motor torque cannot exceed maximum torque. Thus, when the initial tilt angle becomes too large that the required motor torque cannot be produced, the bicycle robot cannot be back to the upright position.

Criterion	LQR	LMPC	NMPC
MAE (rad)	0.0017	0.0017	0.0018
Absolute maximum overshoot (rad)	0.0043	0.0043	0.0043
Settling time (s)	1.42	1.42	1.45

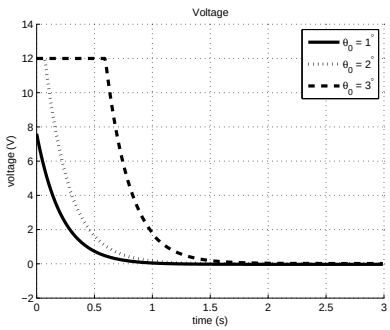
Tab. 2: Performance Criteria as the initial tilt angle is 1 °.



(a)

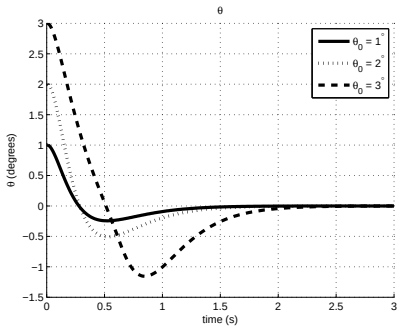


(b)

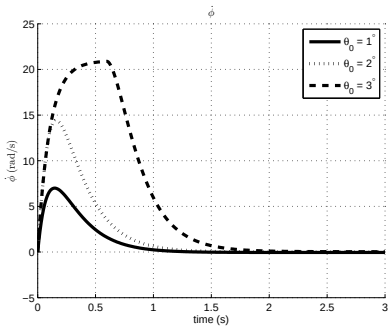


(c)

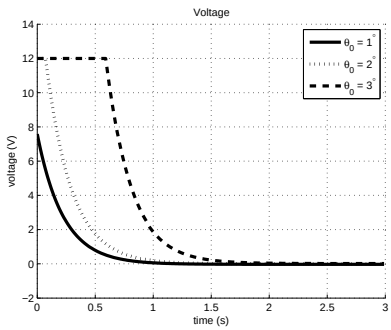
Fig. 8: The simulation results using the LMPC control law when the initial tilt angles are set to 1 °, 2 °, and 3 °: (a) tilt angle, (b) reaction wheel angular speed, and (c) the motor voltage input for driving the reaction wheel.



(a)

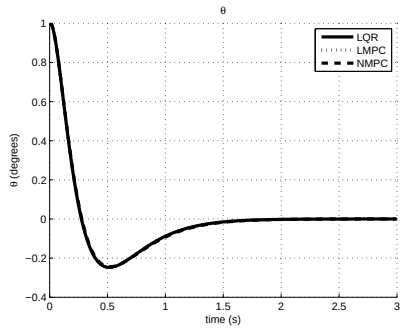


(b)

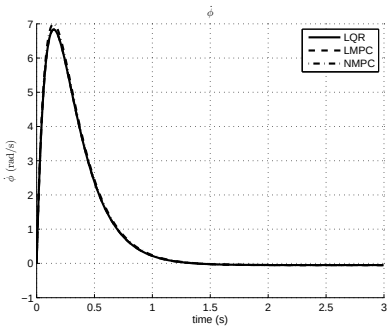


(c)

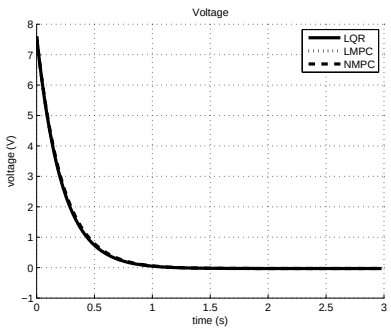
Fig. 9: The simulation results using the NMPC control law when the initial tilt angles are set to 1 °, 2 °, and 3 °: (a) tilt angle, (b) reaction wheel angular speed, and (c) the motor voltage input for driving the reaction wheel.



(a)

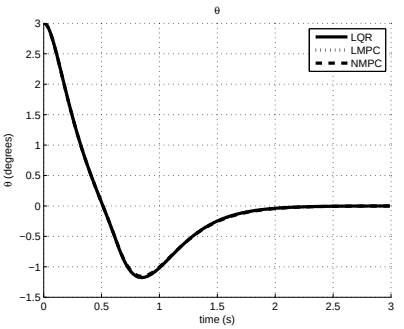


(b)

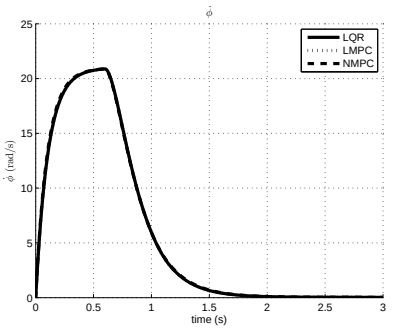


(c)

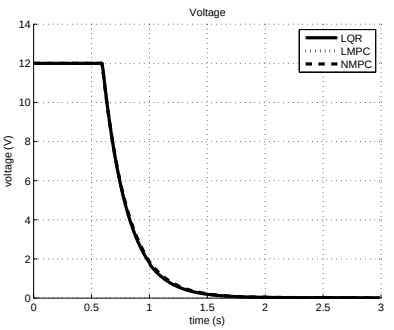
Fig. 10: The simulation results using the LQR, LMPC, and NMPC control laws when the initial tilt angle is set to 1 °: (a) tilt angle, (b) reaction wheel angular speed, and (c) the motor voltage input for driving the reaction wheel.



(a)



(b)



(c)

Fig. 11: The simulation results using the LQR, LMPC, and NMPC control laws when the initial tilt angle is set to 3 °: (a) tilt angle, (b) reaction wheel angular speed, and (c) the motor voltage input for driving the reaction wheel.

Criterion	LQR	LMPC	NMPC
MAE (rad)	0.0133	0.0133	0.0132
Absolute maximum overshoot (rad)	0.0205	0.0205	0.0202
Settling time (s)	2.16	2.16	2.20

Tab. 3: Performance Criteria as the initial tilt angle is 3 °.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we studied stabilization of a self-balancing bicycle robot with a reaction wheel under voltage input constraints, where the objective is to balance the bicycle robot at the upright position at the zero forward speed. Although there have been many controller schemes in the literature to stabilize a bicycle robot, in this study, LQR, LMPC and NMPC controllers were designed and implemented. Simulation results showed that the bicycle robot has been successfully stabilized by using our proposed three controllers.

Currently, we are building a real bicycle robot which can be used to validate our control laws in real-world environments. In addition, we will extend our controller to accomplish path following tasks using a GPS module.

ACKNOWLEDGEMENTS

This research has been financially supported by Faculty of Engineering, Mahasarakham University.

(Received May 11, 2013)

REFERENCES

[1] F. Allgöwer, R. Findeisen, and Z. K. Nagy: Nonlinear model predictive control: from theory to application. J. Chin. Inst. Chem. Eng. *35* (2004), 3, 299–315.

[2] A. V. Beznos, A. M. Formalsky, 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 stabilization. In: Proc. International Conference on Robotics and Automation, Leuven 1998, pp. 2670–2675. DOI:10.1109/robot.1998.680749

[3] T. Bui and M. Parnichkun: Balancing control of bicyrobo by particle swarm optimization-based structure-specified mixed h2/hinf control. Internat. J. Adv. Robot. Syst. *5* (2008), 4, 395–402. DOI:10.5772/6235

[4] M. Defoort and T. Murakami: Second order sliding mode control with disturbance observer for bicycle stabilization. In: Proc. International Conference on Intelligent Robots and Systems, Nice 2008, pp. 2822–2827. DOI:10.1109/iros.2008.4650685

[5] J. Gallaspy: Gyroscopic Stabilization of an Unmanned Bicycle. Master's Thesis, Auburn University, 1999.

[6] L. Keo and Y. Masaki: Trajectory control for an autonomous bicycle with balancer. In: Proc. IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Xi'an 2008, pp. 676–681. DOI:10.1109/aim.2008.4601741

[7] L. Keo and M. Yamakita: Control of an autonomous electric bicycle with both steering and balancer controls. Adv. Robot. *25* (2011), 1–22. DOI:10.1163/016918610x538462

[8] S. Lee and W. Ham: Self-stabilizing strategy in tracking control of unmanned electric bicycle with mass balance. In: Proc. International Conference on Intelligent Robots and Systems, Lausanne 2002, pp. 2200–2205. DOI:10.1109/irds.2002.1041594

[9] G. Lei, L. Qi-zheng, W. Shi-min, and Z. Yu-feng: Design of linear quadratic optimal controller for bicycle robot, automation and logistics. In: Proc. International Conference on Automation and Logistics (ICAL), Shenyang 2009, pp. 1968–1972. DOI:10.1109/ical.2009.5262628

[10] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert: Constrained model predictive control: Stability and optimality. Automatica *36* (2000), 6, 789–814. DOI:10.1016/s0005-1098(99)00214-9

[11] P. Pongpaew: Balancing Control of a Bicycle Robot by Centrifugal Force. Master's Thesis, Asian Institute of Technology, 2010.

[12] P. O. M. Scokaert and J. B. Rawlings: Constrained linear quadratic regulation. IEEE Trans. Automat. Control *43* (1998), 8, 1163–1169. DOI:10.1109/9.704994

[13] Y. Tanaka and T. Murakami: Self sustaining bicycle robot with steering controller. In: Proc. IEEE International Workshop on Advanced Motion Control, Kawasaki 2004, pp. 193–197. DOI:10.1109/amc.2004.1297665

[14] J. Yi, D. Song, A. Levandowski, and S. Jayasuriya: Trajectory tracking and balance stabilization control of autonomous motorcycle. In: Proc. International Conference on Robotics and Automation, Orlando 2006, pp. 2583–2589. DOI:10.1109/robot.2006.1642091

Kiattisin Kanjanawanishkul, Mechatronics Research Unit, Faculty of Engineering, Mahasarakham University. Thailand.
e-mail: kiattisin.k@msu.ac.th