

# Trajectory planning and tracking control for autonomous bicycle robot

Jing Yuan · Huan Chen · Fengchi Sun ·  
Yalou Huang

Received: 13 January 2014 / Accepted: 29 April 2014 / Published online: 25 May 2014  
© Springer Science+Business Media Dordrecht 2014

**Abstract** Control of the autonomous bicycle robot offers considerable challenges to the field of robotics due to its nonholonomic, underactuated, and nonminimum-phase properties. Furthermore, instability and complex dynamic coupling make the trajectory planning of the bicycle robot even more challenging. In this paper, we consider both trajectory planning and tracking control of the autonomous bicycle robot. The desired motion trajectory of the contact point of the bicycle's rear wheel is constructed using the parameterized polynomial curve that can connect two given endpoints with associated tangent angles. The parameters of the polynomial curve are determined by minimizing the maximum of the desired roll angle's equilibrium of the bicycle, and this optimization problem is solved by the particle swarm optimization algorithm. Then, a control scheme that can achieve full-state trajectory tracking while maintaining the bicycle's balance is proposed by combining a planar trajectory tracking controller with a roll angle balance controller. Simulation results are presented to demonstrate the effectiveness of the proposed method.

**Keywords** Trajectory planning · Trajectory tracking control · Autonomous bicycle robot · Particle swarm optimization

J. Yuan (✉) · H. Chen · F. Sun · Y. Huang  
College of Computer and Control Engineering,  
Nankai University, 94, Weijin Road, Tianjin 300071, China  
e-mail: nkyuanjing@gmail.com

## 1 Introduction

Single-track vehicles, such as motorcycles and bicycles, provide superior maneuverability and deployment performance in comparison to double-track vehicles, such as cars and trucks. Recently, autonomous bicycle and motorcycle robots attract much attention, because they can be used as a kind of efficient transportation tools working in the cities and/or in the mountains without any environmental burden and do not require wide contact space to the ground. Moreover, the light weight of the single-track vehicles also provides attractive properties, such as high energy efficiency and fast acceleration.

Research on the modeling of the bicycles and motorcycles has a long history. The reader is referred to [1] and [2] for a comprehensive review of the existing methods. Bicycles share many similar properties with motorcycles. Generally speaking, the motorcycle models differ from the bicycle models in the way contact between tires, and ground is modeled. The bicycle models, such as Getz's model proposed in [3] and [4], assume nonholonomic contact, while the motorcycle models include a more realistic tire-ground interaction model, considering lateral sliding and normal load. Getz's bicycle model, also called the simplified motorcycle model in [5], captures many important aspects of real bicycle and motorcycle dynamics and has been applied effectively in the development of a smart motorcycle driver [6]. More complex motorcycle models were considered in [7] and [8]. In [7],

a dynamic model of the motorcycle was developed by using Getz's modeling approach [4]. The model is modified by adding motorcycle trail and caster angle, and can capture the steering effect on the vehicle motion. In [8], Meijaard and Popov analyzed the dynamic behaviors of the motorcycle models by using the motion equations generated by a symbolic multibody program. Branches of stationary and periodic solutions together with bifurcations of these solutions were calculated for the motorcycle dynamics.

Motion control of the single-track vehicles became a popular topic for researchers in the latter half of the last century. It is normally classified into two categories: stabilization control [9–12] and tracking control [4, 7, 13–18].

Stabilization control is to balance a stationary robot around the upright position with the steering handlebar and/or a balancer. In [9], decoupling and linearization were achieved based on feedback linearization for the single-input single-output dynamic model of the bicycle robot. Then, a steering handlebar and a balancer were simultaneously used to stabilize the robot around the upright position. Keo and Yamakita proposed a nonlinear output-zeroing controller for stabilizing a bicycle robot at zero speed, also with both steering handlebar and balancer [10]. A first-order sliding mode controller with an observer is given in [11] to stabilize a bicycle robot at zero velocity and varying velocities. In [12], stability analysis of a human neurological control of a stationary bicycle was accomplished by the sweeping frequency method and the stability charts for time-delay dynamical systems.

Tracking control is to guide a robot to track a given trajectory at a non-zero velocity, while preventing the robot from falling down. In [4], Getz proposed a feedback control law such that the bicycle robot's position can track a desired trajectory. The tracking control of robot's position was also solved in [7], where an asymptotically stable controller was designed. However, tracking of the robot's yaw angle was not considered in the above two controllers. In [13], a simplified inverted pendulum model was utilized, and a proportional derivative controller with a disturbance observer was designed for a bicycle robot to track a straight-line trajectory. However, the controller was designed based on an approximate linearized model, and the linearization errors were not considered. In addition, the robot can only execute its motion approximately on a straight-line trajectory, which is quite limited.

Yamaguchi et al. [14] designed a self-sustaining controller for an electric bicycle using acceleration control based on the Lyapunov method and the backstepping technique. The running performance in low-speed range was improved by the proposed acceleration control. Thus, the controller is applicable to a wider speed range than the conventional methods. Defoort and Murakami proposed a second-order sliding mode controller for stabilizing a bicycle robot around a trajectory. The controller is robust against disturbances and measurement noises [15]. Hwang et al. [16] proposed a variable structure underactuated tracking controller for an electric bicycle. The time-varying system uncertainties and the wind effect were both considered. In [17], a virtual holonomic constraint (VHC) which specifies the roll angle of the bicycle robot as a function of its position along a strictly convex Jordan curve was adopted. It was shown that if the mean curvature of the curve is sufficiently small, then the VHC manifold is invariant, and the closed orbit is asymptotically stable. The bicycle robot can traverse the curve with bounded speed, and its speed is periodic in steady-state. In [18], Gundes and Nanjangud proposed low-order controllers with only the steering torque as input to control any number of linear bicycle models at different forward velocity. The controllers are simple and provide some freedom in the design parameters. Frezza et al. [19] implemented a path following controller for a simplified motorcycle model using a model predictive control (MPC) strategy. Yi et al. [20] analyzed the external/internal convertible (EIC) dynamical structure of the motorcycle dynamics. A nonlinear tracking controller was designed based on the EIC system. The controller guarantees exponential convergence of the motorcycle's position to a neighborhood of the desired trajectory while the roll angle converges to a neighborhood of the desired equilibrium.

Trajectory planning is also a key issue for the control and navigation of the bicycle and motorcycle robots. Hauser et al. addressed the trajectory exploration for the nonholonomic motorcycle model in [5, 21]. In [5], the problem whether a bounded roll trajectory is guaranteed to exist was addressed for a simplified motorcycle, given a planar trajectory with a desired velocity signal. The motorcycle was constrained on the given planar trajectory, and then an iterative algorithm was designed for calculating the required roll trajectory approximately consistent with the planar trajectory. In [21], an extended control system was embedded into the motorcycle dynamics by adding two nonphysical con-

trol inputs. Then, these additional control inputs were optimized away by the projection operator approach, such that a flatland trajectory can be lift into a state-input trajectory.

From the above literature review, we can see that almost all the related researches focus on modeling and motion control of the single-track robots. However, the problem of trajectory planning of the bicycle and motorcycle robots has not been fully addressed. Moreover, most of the previous researches on tracking control of the bicycles and motorcycles, e.g., [3, 4, 7, 13, 20], only solved the problem of partial-state tracking control, i.e., only the position coordinates and the roll angle of the system were controlled to track their desired trajectories, without taking the yaw angle into account. In this paper, we consider both the trajectory planning and tracking control for the Getz's bicycle model. The main contributions of this paper are twofold. (1) We plan an optimal desired trajectory for the bicycle robot by minimizing the maximum of the roll angle's equilibrium of the robot. This optimization problem is solved by the use of the PSO algorithm. (2) We design a full-state tracking controller for the bicycle robot to approximately track the desired trajectory with a good balance. Compared with the existing tracking controllers, the proposed controller can drive not only the position and roll angle but also the yaw angle to track the desired trajectories.

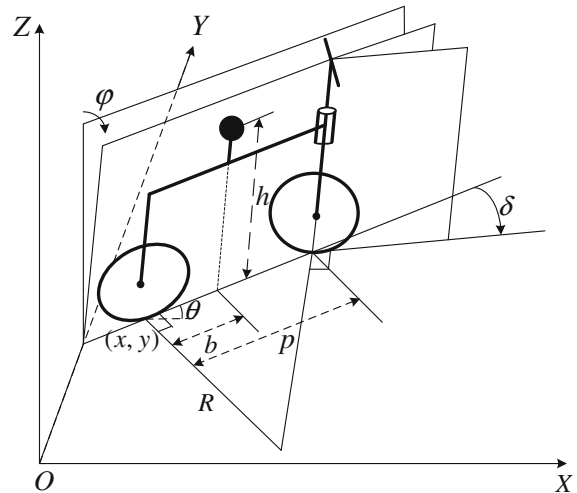
The remainder of the paper is organized as follows: the dynamics of the bicycle robot is described in Sect. 2. In Sect. 3, the problem of trajectory planning is solved by the PSO algorithm. Then, the tracking and balance controller is designed in Sect. 4. Simulations are given in Sect. 5 to verify the proposed method. The paper is concluded in Sect. 6.

## 2 Dynamics of bicycle robot

We view the bicycle robot [4, 5] as a plane that is allowed to move and roll on the ground, as shown in Fig. 1. For simplicity, the wheels of the bicycle robot are considered to roll without sideslip and to move in a flat plane without vertical motion.

Consider a ground-fixed inertial reference frame  $O - XYZ$  with  $x$  and  $y$  axes in the ground plane and  $z$  axis perpendicular to the ground plane in the direction opposite to gravity. The key variables and parameters are

$x$   $x$  coordinate of the point of contact of the rear wheel



**Fig. 1** Bicycle robot model

- $y$   $y$  coordinate of the point of contact of the rear wheel
- $\theta$  Yaw (or heading) angle of the robot frame (rear wheel)
- $\varphi$  Roll (or lean) angle of the robot frame (rear wheel)
- $\delta$  Steering angle
- $m$  Robot mass
- $h$  Height of the center of mass (when the robot is vertical)
- $p$  Robot wheelbase
- $b$  Horizontal distance between rear wheel contact point and the robot center of mass
- $R$  Turning radius
- $g$  Gravity acceleration

Consider Fig. 1, the generalized coordinates describing the configuration of the bicycle robot are the position  $(x, y)$  of the contact point of rear wheel, yaw angle  $\theta$ , and roll angle  $\varphi$ , which are written in short as  $\mathbf{q} = (x, y, \theta, \varphi)^T$ . For roll angle  $\varphi$ , we define tilting right from the vertical line is positive and for steering angle  $\delta$ , turning left is the positive direction.

The kinematics of the planar motion is given as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cdot \cos \theta \\ v \cdot \sin \theta \\ v \cdot \sigma \end{bmatrix} \quad (1)$$

where  $v$  is the longitudinal velocity of the rear wheel, and  $\sigma$  is the curvature. Note that the nonholonomic constraint of the rear wheel implies that the lateral velocity of the rear wheel is  $v_{\perp} = 0$ .

The roll dynamics of the bicycle robot is that of an inverted pendulum given by

$$h\ddot{\varphi} = g \sin \varphi + [(1 + h\sigma \sin \varphi)\sigma v^2 + b(\dot{v}\sigma + v\dot{\sigma})] \cos \varphi \tag{2}$$

Equation (2) describes the internal dynamics of the system. The control inputs of the model given by (1) and (2) are  $\dot{v}$  and  $\dot{\sigma}$ , which are denoted by  $\mathbf{u} = (u_1, u_2)^T = (\dot{v}, \dot{\sigma})^T$ . The complete Lagrangian dynamics and the derivation for the reduced equations were given in [4]. The model given by (1) and (2) is the same as that described in [4]. It is a simplified version of the motorcycle model in [7], because the motorcycle trail and caster angle are not considered.

Next, we consider the roll angle's equilibrium, denoted by  $\varphi_e$ . We define the implicit function  $F(\varphi)$  of  $\varphi$  as

$$F(\varphi) = g \sin \varphi + [(1 + h\sigma \sin \varphi)\sigma v^2 + b(\dot{v}\sigma + v\dot{\sigma})] \cos \varphi \tag{3}$$

For a given external control input  $\mathbf{u}^{\text{ext}} = (\dot{v}^{\text{ext}}, \dot{\sigma}^{\text{ext}})^T$ , the roll angle's equilibrium  $\varphi_e$  is a solution of the algebraic equation  $F(\varphi) = 0$ , by substituting  $\mathbf{u}^{\text{ext}} = (\dot{v}^{\text{ext}}, \dot{\sigma}^{\text{ext}})^T$  into (3). The internal equilibrium manifold for the system is

$$\varepsilon(t) = \{(x, y, \varphi, \dot{\varphi}) | \varphi = \varphi_e, \dot{\varphi} = 0\} \tag{4}$$

### 3 Trajectory planning for bicycle robot

Assume that a start point  $P_0 = (x_0, y_0)$  and a destination point  $P_f = (x_f, y_f)$  of the robot have been assigned in the Cartesian plane. We will plan a curve between  $P_0$  and  $P_f$  which satisfies given constraints on the initial and final tangent angles  $\theta_0$  and  $\theta_f$ . The curve can be represented by a parameterization  $P_d(t) = (x_d(t), y_d(t))$  with time parameter  $t \in [t_0, t_f]$ , where the initial time  $t_0$  can be set as  $t_0 = 0$  without loss of generality, and the final time  $t_f$  can be assigned according to the dynamic properties of the bicycle robot. The final time  $t_f$  can also be selected by some optimality criterion. In this paper, we set  $t_f$  to be a constant.

At least the first derivative of  $P_d(t)$ 's curvature should exist, because one of the control inputs of the robot is  $\dot{\sigma}$ . We choose 3rd order polynomials of time to parameterize  $x_d(t)$  and  $y_d(t)$  as

$$\begin{cases} x_d(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 \\ y_d(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 \end{cases} \tag{5}$$

Subject to the boundary conditions

$$\begin{cases} P_d(t_0) = P_0 \\ \theta_d(t_0) = \theta_0 \\ P_d(t_f) = P_f \\ \theta_d(t_f) = \theta_f \end{cases} \tag{6}$$

The polynomial coefficients  $\alpha_i$  and  $\beta_i$  ( $i = 0, 1, 2, 3$ ) are detailed by the following linear algebraic equations:

$$\begin{cases} \alpha_0 = x_0 \\ \beta_0 = y_0 \\ \alpha_1 = \lambda_1 \cos \theta_0 \\ \beta_1 = \lambda_1 \sin \theta_0 \\ \alpha_0 + \alpha_1 t_f + \alpha_2 t_f^2 + \alpha_3 t_f^3 = x_f \\ \beta_0 + \beta_1 t_f + \beta_2 t_f^2 + \beta_3 t_f^3 = y_f \\ \alpha_1 + 2\alpha_2 t_f + 3\alpha_3 t_f^2 = \lambda_2 \cos \theta_f \\ \beta_1 + 2\beta_2 t_f + 3\beta_3 t_f^2 = \lambda_2 \sin \theta_f \end{cases} \tag{7}$$

where the real parameters  $\lambda_1$  and  $\lambda_2$  in (7) can be freely selected, and they can influence the curve shape without violating the endpoint constraint (6). The parameter vector  $\lambda = [\lambda_1, \lambda_2]^T$  can be optimized to adjust the shape of the planned trajectory such that the maximum of the desired roll angle's equilibrium of the robot is minimized.

Note that the planar trajectory of the bicycle robot has six constraints on the initial and final states, i.e., four position constraints  $P_0 = (x_0, y_0)$  and  $P_f = (x_f, y_f)$ , as well as two yaw angle constraints  $\theta(t_0) = \theta_0$  and  $\theta(t_f) = \theta_f$ . The 3rd order polynomial in terms of time given in (5) has eight parameters. Thus, the two parameters  $\lambda_1$  and  $\lambda_2$  can be freely chosen. Higher order polynomials can be adopted when more constraints are imposed on the robot's motion trajectory.

Assume that the desired planar motion trajectory of the bicycle is denoted by  $\Gamma_d(t) = (x_d(t), y_d(t), \theta_d(t))$ , where  $x_d(t)$  and  $y_d(t)$  are given in (5). From (5), it is shown that  $x_d(t)$  and  $y_d(t)$  are at least thrice differentiable. From (1), the desired linear velocity is  $v_d = \pm \sqrt{\dot{x}_d^2 + \dot{y}_d^2}$  and the desired yaw angle is  $\theta_d = a \tan 2(\dot{y}_d, \dot{x}_d) + k\pi$  ( $k \in \mathbb{Z}$ ). In this paper, we assume that the bicycle robot can only move forward.

Thus, the desired linear velocity and yaw angle of the system are

$$v_d = \sqrt{\dot{x}_d^2 + \dot{y}_d^2} \quad (8)$$

$$\theta_d = a \tan 2(\dot{y}_d, \dot{x}_d) + 2n\pi \quad (n \in \mathbb{Z}) \quad (9)$$

The desired curvature of the path is

$$\sigma_d = \frac{\ddot{y}_d \dot{x}_d - \ddot{x}_d \dot{y}_d}{(\dot{x}_d^2 + \dot{y}_d^2)^{3/2}} \quad (10)$$

Then the desired roll angle's equilibrium  $\varphi_{ed}$ , compatible with  $v_d$  and  $\sigma_d$ , can be determined by  $F(\varphi_d) = 0$ , where

$$F(\varphi_d) = g \sin \varphi_d + [(1 + h\sigma_d \sin \varphi_d)\sigma_d v_d^2 + b(\dot{v}_d \sigma_d + v_d \dot{\sigma}_d)] \cos \varphi_d \quad (11)$$

In order to prevent the robot from falling down and retain balance more reliably,  $\lambda$  is calculated by solving the following optimization problem:

$$\min_{\lambda \in \mathbb{R}^2} \max_{t \in [t_0, t_f]} |\varphi_{ed}(\lambda, t)| \quad (12)$$

For a given  $\lambda$ , e.g.,  $\lambda = \lambda_0$ , the problem to solve

$\max_{t \in [t_0, t_f]} |\varphi_{ed}(\lambda_0, t)|$  is equivalent to

$$\begin{cases} \max_{t \in [t_0, t_f]} \varphi_{ed}(\lambda_0, t), & \text{if } \varphi_{ed}(\lambda_0, t) \geq 0 \\ \min_{t \in [t_0, t_f]} \varphi_{ed}(\lambda_0, t), & \text{otherwise} \end{cases} \quad (13)$$

Subject to

$$F(\varphi_{ed}, t) = g \sin \varphi_{ed} + [(1 + h\sigma_d \sin \varphi_{ed})\sigma_d v_d^2 + b(\dot{v}_d \sigma_d + v_d \dot{\sigma}_d)] \cos \varphi_{ed} = 0 \quad (14)$$

In fact, the solution to (13) subject to (14) is to calculate the local extrema of the implicitly defined function of  $\varphi_{ed}$  about the time variable  $t$ . The detailed derivation is placed in the appendix.

The optimization problem in (12) is strongly nonlinear and thus has many local minima. For this reason, it has to be solved by an algorithm which has the ability of global optimization. In this paper, the optimal solution is obtained by using the PSO algorithm. The PSO algorithm was proposed by Eberhart and Kennedy in [22]. It has been widely used to solve nonlinear optimization problems. Some improvements and applications have also been discussed in [23–27]. The PSO algorithm always converges faster and searches better than

the gradient descent technique or direct search, which may fall into a local extremum. On the contrary, the PSO algorithm has the ability of global search.

In the PSO algorithm, a candidate solution to the optimization problem is represented by a particle. The PSO uses a population of particles to constitute multiple candidate solutions. The PSO iteratively improves each candidate solution with regard to a given fitness function, by moving the particles around in the search-space over the particle's position and velocity.

The particle with the optimal fitness value is regarded as the local optimal solution found so far and is called *pBest*. The best one among all the *pBest* particles in the search-space is regarded as the global optimal solution and is called *gBest*. Each particle's movement is influenced by *pBest* but is also guided toward *gBest*. The velocity of a particle is updated by

$$V_{iq}^{\text{new}} = \omega \cdot V_{iq}^{\text{old}} + c_1 \cdot \text{Rand}_1() \cdot (pBest_{iq} - x_{iq}) + c_2 \cdot \text{Rand}_2() \cdot (gBest_q - x_{iq}) \quad (15)$$

where  $V_{iq}^{\text{new}}$  denotes the velocity of the  $i$ th particle in the  $q$ th dimension in the next generation;  $V_{iq}^{\text{old}}$  denotes the velocity of the  $i$ th particle in the  $q$ th dimension in the current generation;  $pBest_{iq}$  is the current *pBest* value of the  $i$ -th particle in the  $q$ th dimension;  $gBest_q$  is the current *gBest* value of the whole particle swarm in the  $q$ th dimension;  $x_{iq}$  is the current position of the  $i$ th particle in the  $q$ th dimension;  $\omega$  is the weight;  $c_1$  is the acceleration of a particle to move to its *pBest*;  $c_2$  is the acceleration of a particle to move to the *gBest*;  $\text{Rand}_1()$  and  $\text{Rand}_2()$  are two random numbers between 0 and 1.

The new position of a particle is obtained by

$$x_{iq}^{\text{new}} = x_{iq}^{\text{old}} + V_{iq}^{\text{new}} \quad (16)$$

We use the PSO algorithm to search the optimal  $\lambda$ . The fitness function is defined by  $f(\lambda_0) = 1/[1 + \max_{t \in [t_0, t_f]} |\varphi_{ed}(\lambda_0, t)|]$  for a given  $\lambda = \lambda_0$ . As a result, a desired motion trajectory with minimized maximum roll angle's equilibrium is achieved.

#### 4 Controller design

Trajectory tracking control is one of the representative problems of the mobile mechanical platforms, such as

unicycle systems, car-like robots, and bicycle robots. Trajectory tracking control of the mobile robots has been investigated for many years and various kinds of methods have been proposed in [28–32]. However, trajectory tracking control for the bicycle and motorcycle robots is a new issue in the field of robotics. In this section, we consider the trajectory tracking control for the bicycle robot given in (1) and (2). This problem has to satisfy two specifications. First, the robot has to be balanced such that it will not fall down. Second, the robot must follow  $\Gamma_d$ . To meet these two requirements, the controller is developed in three steps. The first step is to design a balancing controller for roll dynamics. The second step is to design a controller for tracking the desired planar trajectory  $\Gamma_d$ . Finally, we design a controller for tracking with balance by balancing the roll dynamics around its equilibrium manifold, which depends on the tracking controller designed in the second step.

#### 4.1 Controller for balancing roll angle

For roll dynamics in (2), we consider the control input

$$\begin{aligned} \dot{\sigma}^{int} = & \frac{1}{bv \cos \varphi} [h\ddot{\varphi}_d - k_1(\dot{\varphi} - \dot{\varphi}_d) - k_2(\varphi - \varphi_d) \\ & - g \sin \varphi - (1 + h\sigma^{int} \sin \varphi)v^2\sigma^{int} \\ & \cos \varphi - b\dot{v}\sigma^{int} \cos \varphi] \end{aligned} \tag{17}$$

where the constants  $k_i$  ( $i = 1, 2$ ) are chosen such that the polynomial equation  $hs^2 + k_1s + k_2 = 0$  is Hurwitz. It is apparent that by (17), the roll angle  $\varphi$  exponentially converges to its desired trajectory.

#### 4.2 Controller for tracking desired planar trajectory

If we ignore the roll angle, a tracking controller can be designed for planar motion of the robot. We define the tracking errors with respect to the moving body frame and represent them as seen by an observer riding the vehicle.

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix} \tag{18}$$

From (18),  $\lim_{t \rightarrow \infty} (e_x, e_y, e_\theta) = (0, 0, 2n\pi) \Leftrightarrow \lim_{t \rightarrow \infty} (x, y, \theta) = (x_d, y_d, \theta_d + 2n\pi) \quad n \in \mathbf{Z}$ .

For the planar motion of the robot, we give a tracking controller as follows.

**Theorem 1** Assume that the desired planar trajectory is given by  $\Gamma_d(x_d, y_d, \theta_d)$ , where  $x_d$  and  $y_d$  are thrice differentiable,  $v_d \neq 0$ ,  $\lim_{t \rightarrow \infty} \sigma_d \neq 0$ , if the control inputs are

$$\begin{aligned} v_{pl}^{ext} &= v_d \\ \sigma_{pl}^{ext} &= \sigma_d + \frac{2}{k_4 \cdot v_d} \\ & \cdot \left[ 2k_3 \cdot v_d \cdot \left( e_y \cdot \cos \frac{e_\theta}{2} - e_x \cdot \sin \frac{e_\theta}{2} \right) + \sin \frac{e_\theta}{2} \right] \end{aligned} \tag{19}$$

$(x, y, \theta)$  will asymptotically converge to  $\Gamma_d$ , where  $k_3 > 0, k_4 > 0$ .

*Proof* From (18),

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} v_{pl}^{ext} \\ v_{pl}^{ext} \sigma_{pl}^{ext} \end{bmatrix} + \begin{bmatrix} v_d \cdot \cos e_\theta \\ v_d \cdot \sin e_\theta \\ v_d \sigma_d \end{bmatrix} \tag{20}$$

Choose the Lyapunov function as  $V = \frac{1}{2}k_3 \cdot (e_x^2 + e_y^2) + k_4 \cdot (1 - \cos \frac{e_\theta}{2})$ .

Then

$$\begin{aligned} \dot{V} &= k_3 \cdot (e_x \cdot \dot{e}_x + e_y \cdot \dot{e}_y) + \frac{k_4}{2} \cdot \sin \frac{e_\theta}{2} \cdot \dot{e}_\theta \\ &= k_3 \cdot \left[ e_x \cdot \left( e_y v_{pl}^{ext} \sigma_{pl}^{ext} - v_{pl}^{ext} + v_d \cdot \cos e_\theta \right) \right. \\ & \quad \left. + e_y \cdot \left( -e_x \cdot v_{pl}^{ext} \sigma_{pl}^{ext} + v_d \cdot \sin e_\theta \right) \right] + \frac{k_4}{2} \\ & \quad \cdot \sin \frac{e_\theta}{2} \cdot \left( v_d \sigma_{pl}^{ext} - v_{pl}^{ext} \sigma_{pl}^{ext} \right) \\ &= k_3 v_d \cdot [(\cos e_\theta - 1) \cdot e_x + e_y \cdot \sin e_\theta] \\ & \quad + \frac{k_4}{2} \left( v_d \sigma_d - v_{pl}^{ext} \sigma_{pl}^{ext} \right) \cdot \sin \frac{e_\theta}{2} \\ &= \left[ 2k_3 v_d \cdot \left( e_y \cdot \cos \frac{e_\theta}{2} - e_x \cdot \sin \frac{e_\theta}{2} \right) \right. \\ & \quad \left. + \frac{k_4}{2} \left( v_d \sigma_d - v_{pl}^{ext} \sigma_{pl}^{ext} \right) \right] \cdot \sin \frac{e_\theta}{2} \\ &= -\sin^2 \frac{e_\theta}{2} \leq 0 \end{aligned}$$

$$\begin{aligned} V \geq 0, \dot{V} \leq 0 &\Rightarrow V \in L_\infty \Rightarrow e_x, e_y \in L_\infty \\ &\Rightarrow v_{pl}^{ext}, \sigma_{pl}^{ext} \in L_\infty \Rightarrow \dot{e}_x, \dot{e}_y, \dot{e}_\theta \in L_\infty \\ &\Rightarrow \lim_{t \rightarrow \infty} \sin \frac{e_\theta}{2} = 0 \Rightarrow \lim_{t \rightarrow \infty} e_\theta = 2n\pi, n \in \mathbf{Z}. \end{aligned}$$

In addition,

$$\begin{aligned} \ddot{\theta} &= \dot{v}_{pl}^{ext} \sigma_{pl}^{ext} + v_{pl}^{ext} \dot{\sigma}_{pl}^{ext} - (\dot{v}_d \sigma_d + v_d \dot{\sigma}_d) \\ &= -\frac{4k_3}{k_4} \left[ \dot{v}_d \left( e_y \cdot \cos \frac{e_\theta}{2} - e_x \cdot \sin \frac{e_\theta}{2} \right) \right. \\ &\quad \left. + v_d \left( \dot{e}_y \cos \frac{e_\theta}{2} - \frac{1}{2} e_y \cdot \sin \frac{e_\theta}{2} \cdot \dot{e}_\theta - \dot{e}_x \cdot \sin \frac{e_\theta}{2} \right. \right. \\ &\quad \left. \left. - \frac{1}{2} e_x \cdot \cos \frac{e_\theta}{2} \cdot \dot{e}_\theta \right) \right] - \frac{1}{k_4} \cdot \cos \frac{e_\theta}{2} \cdot \dot{e}_\theta \end{aligned}$$

where  $\dot{v}_d = \frac{\dot{x}_d \cdot \dot{x}_d + \dot{y}_d \cdot \dot{y}_d}{\sqrt{\dot{x}_d^2 + \dot{y}_d^2}} \in L_\infty$ . Thus,  $\ddot{\theta} \in L_\infty$ , we know that  $\dot{e}_\theta$  is uniformly continuous. According to Extended Barbalat Lemma,  $\lim_{t \rightarrow \infty} \dot{e}_\theta = \lim_{t \rightarrow \infty} -\frac{2}{k_4} [2k_3 \cdot v_d \cdot (e_y \cdot \cos \frac{e_\theta}{2} - e_x \cdot \sin \frac{e_\theta}{2}) + \sin \frac{e_\theta}{2}] = \lim_{t \rightarrow \infty} (-\frac{4k_3}{k_4} \cdot v_d \cdot e_y) = 0 \Rightarrow \lim_{t \rightarrow \infty} e_y = 0$ . Then, from (19) it is known that  $\lim_{t \rightarrow \infty} \sigma = \sigma_d$ .

From (20),

$$\begin{aligned} \ddot{e}_y &= -\dot{e}_x v_{pl}^{ext} \sigma_{pl}^{ext} - e_x (\dot{v}_{pl}^{ext} \sigma_{pl}^{ext} + v_{pl}^{ext} \dot{\sigma}_{pl}^{ext}) \\ &\quad + \dot{v}_d \sin e_\theta + v_d \cdot \cos e_\theta \cdot \dot{e}_\theta \\ &= -\dot{e}_x v_{pl}^{ext} \sigma_{pl}^{ext} - e_x (\dot{v}_d \sigma_d \\ &\quad + v_d \dot{\sigma}_d - \ddot{\theta}) + \dot{v}_d \sin e_\theta + v_d \cos e_\theta \cdot \dot{e}_\theta \end{aligned}$$

where  $\dot{v}_d \sigma_d + v_d \dot{\sigma}_d = \frac{\ddot{y}_d \dot{x}_d - \dot{x}_d \ddot{y}_d}{\dot{x}_d^2 + \dot{y}_d^2} - \frac{2(\ddot{y}_d \dot{x}_d - \dot{x}_d \ddot{y}_d)(\dot{x}_d \dot{x}_d + \dot{y}_d \dot{y}_d)}{(\dot{x}_d^2 + \dot{y}_d^2)^2} \in L_\infty$ . Thus  $\ddot{e}_y \in L_\infty$ ,  $\dot{e}_y$  is uniformly continuous. As a result,  $\lim_{t \rightarrow \infty} \dot{e}_y = \lim_{t \rightarrow \infty} (-e_x v_{pl}^{ext} \sigma_{pl}^{ext} + v_d \sin e_\theta) = \lim_{t \rightarrow \infty} (-e_x v_d \sigma_d) = 0$ . From (20),  $\dot{e}_x = e_y v_{pl}^{ext} \sigma_{pl}^{ext} - v_d(1 - \cos e_\theta) \rightarrow 0$  and  $e_x \in L_\infty$ , thus  $\lim_{t \rightarrow \infty} e_x = 0$  or  $\lim_{t \rightarrow \infty} \sigma_d = 0$ . In the assumption of theorem 1,  $\lim_{t \rightarrow \infty} \sigma_d \neq 0$ , thus  $\lim_{t \rightarrow \infty} e_x = 0$  holds. Hence the proposed controller ensures that the planar motion of the system converges to  $\Gamma_d$  asymptotically, i.e.,  $\lim_{t \rightarrow \infty} (x, y, \theta) = (x_d, y_d, \theta_d)$ .  $\square$

**Remark 1** If  $\lim_{t \rightarrow \infty} \sigma_d = 0$ ,  $\Gamma_d$  will tend to a straight line. In the proof of theorem 1, we have obtained that  $\lim_{t \rightarrow \infty} e_y = 0$  and  $\lim_{t \rightarrow \infty} e_\theta = 0$ . Therefore, we can deduce that  $-\sin \theta_d \cdot (x_d - x) + \cos \theta_d \cdot (y_d - y) = 0$ , i.e.,  $(x, y)$  lies on the desired straight line. In other words, when  $\lim_{t \rightarrow \infty} e_y = 0$  and  $\lim_{t \rightarrow \infty} e_\theta = 0$ , the system converges to the desired straight line.

**Remark 2** In the controller (19), the linear velocity of the system is chosen as the desired one, i.e.,  $v_{pl}^{ext} = v_d$ , thus it can drive the bicycle robot to track the desired trajectory in an identical direction all the time, and the backward behaviors can be avoided.

### 4.3 Controller for tracking with balance

Unlike the systems with stable internal dynamics, the nonminimum phase systems with given initial conditions cannot be made to exactly track arbitrary elements of an open set of the output desired trajectory while maintaining bounded internal dynamics. Instead, we will design a controller for approximate tracking of the desired trajectory while maintaining the balance of the robot.

To achieve tracking the desired planar trajectory, the real external control input  $u^{ext} = (v^{ext}, \dot{\sigma}^{ext})^T$  is designed as

$$\begin{cases} \dot{v}^{ext} = \dot{v}_{pl}^{ext} - k_5(v^{ext} - v_{pl}^{ext}) \\ \dot{\sigma}^{ext} = \dot{\sigma}_{pl}^{ext} - k_6(\sigma^{ext} - \sigma_{pl}^{ext}) \end{cases} \quad (21)$$

where  $k_5$  and  $k_6$  are positive constants.

Then, we compute the roll angle's equilibrium compatible with  $v^{ext}$  and  $\sigma^{ext}$  by submitting them into (3). The external vector field corresponding to  $v^{ext}$  and  $\sigma^{ext}$  is

$$VF^{ext} = \begin{bmatrix} v^{ext} \cdot \cos \theta \\ v^{ext} \cdot \sin \theta \\ v^{ext} \cdot \sigma^{ext} \end{bmatrix} \quad (22)$$

For the robot balance system, we control the roll angle around its equilibrium manifold  $\varepsilon(t)$  while the planar motion of the robot tracks  $\Gamma_d$ . Thus we approximate the derivatives  $\dot{\varphi}_e$  and  $\ddot{\varphi}_e$  by using directional derivative along the vector field  $VF^{ext}$ . We introduce the Lie derivative as  $\bar{L}_{VF^{ext}} \varphi_e = L_{VF^{ext}} \varphi_e + \partial \varphi_e / \partial t$  and  $\bar{L}_{VF^{ext}}^2 \varphi_e = \bar{L}_{VF^{ext}} \bar{L}_{VF^{ext}} \varphi_e$ . If  $\varphi$  is close to  $\varphi_e$  and  $(x, y, \theta)$  is approximately tracking its desired trajectory  $(x_d, y_d, \theta_d)$ , then  $\bar{L}_{VF^{ext}} \varphi_e$  and  $\bar{L}_{VF^{ext}}^2 \varphi_e$  will be close to  $\dot{\varphi}_e$  and  $\ddot{\varphi}_e$ , respectively [4]. We now give a control input

$$\begin{aligned} \dot{\sigma}_e^{int} &= \frac{1}{bv^{ext} \cos \varphi} \left[ h \bar{L}_{VF^{ext}}^2 \varphi_e - k_1 (\dot{\varphi} - \bar{L}_{VF^{ext}} \varphi_e) \right. \\ &\quad \left. - k_2 (\varphi - \varphi_e) - g \cdot \sin \varphi - \left( 1 + h \sigma_e^{int} \sin \varphi \right) \right. \\ &\quad \left. \times v^{ext} \sigma_e^{int} \cos \varphi - b \dot{v}^{ext} \sigma_e^{int} \cos \varphi \right] \quad (23) \end{aligned}$$

The final controller  $u = (u_1, u_2)^T = (\dot{v}, \dot{\sigma})^T$  is achieved by combining  $\dot{v}^{ext}$  in (21) and  $\dot{\sigma}_e^{int}$  in (23) as

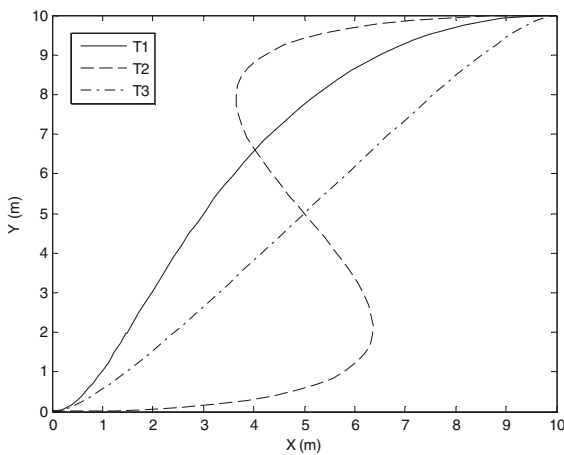
$$\begin{cases} \dot{v} = \dot{v}^{\text{ext}} \\ \dot{\sigma} = \dot{\sigma}_e^{\text{int}} \end{cases} \quad (24)$$

Thus, for initial values of  $\varphi$  and  $\dot{\varphi}$  sufficiently close to  $\varphi_e$  and  $\bar{L}_V F^{\text{ext}} \varphi_e$ , the system state  $(x, y, \theta, \varphi)$  converges to an arbitrarily small neighborhood of  $(x_d, y_d, \theta_d, \varphi_e)$  under the controller of (24).

### 5 Simulations

In the simulations, the parameters of the bicycle robot are  $h = 1.0 \text{ m}$ ,  $p = 1.0 \text{ m}$ ,  $b = 0.5 \text{ m}$ ,  $m = 20 \text{ kg}$ , and  $g = 9.8$ . The initial pose of the bicycle is  $[x_0, y_0, \theta_0]^T = [0, 0, 0]^T$ ,  $\varphi_0 = 0$ ,  $\dot{\varphi}_0 = 0$ . The control parameters are  $k_1 = 1.5$ ,  $k_2 = 1.0$ ,  $k_3 = 1.5$ ,  $k_4 = 1.5$ ,  $k_5 = 5.0$ , and  $k_6 = 5.0$ . The parameters of the PSO algorithm are  $\omega = 1$ ,  $c_1 = 2$ , and  $c_2 = 2$ .

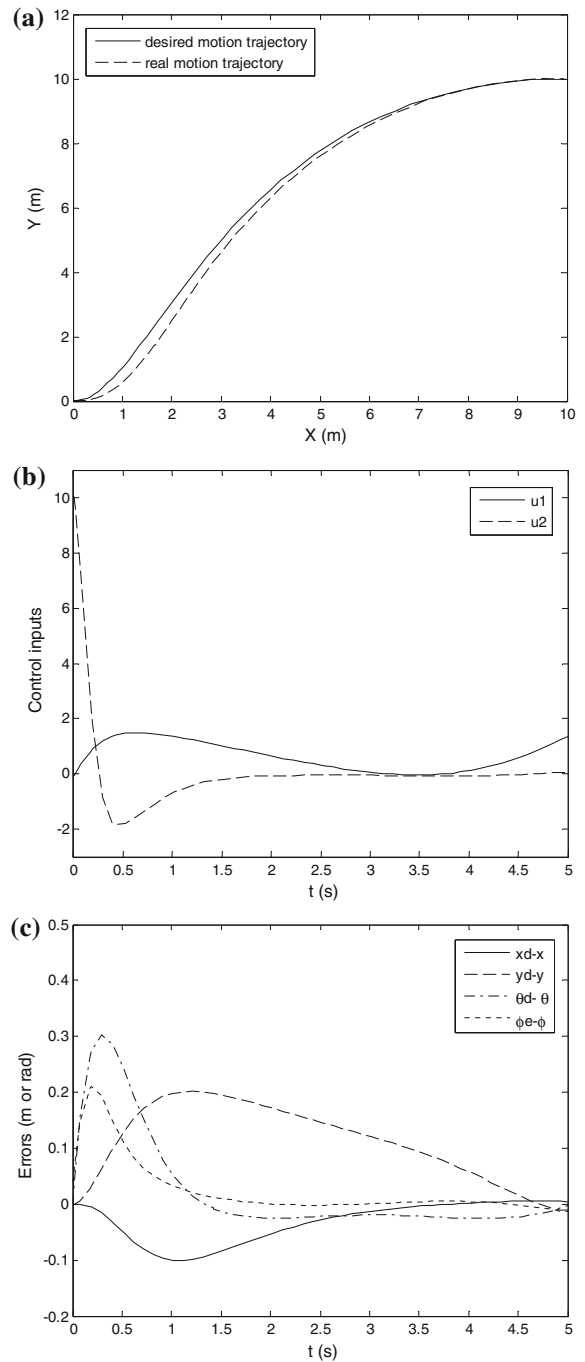
Figure 2 shows a simulation result of trajectory planning, in which the desired final pose is  $[x_f, y_f, \theta_f]^T = [10, 10, 0]^T$ . We set the final time to be  $t_f = 5 \text{ s}$ . T1, T2, and T3 denote, respectively, three trajectories under different parameters, as shown in Table 1. T1 is the planned trajectory using the proposed method, where



**Fig. 2** Simulation result of trajectory planning for  $[x_f, y_f, \theta_f]^T = [10, 10, 0]^T$

**Table 1** Simulation comparison with different parameters

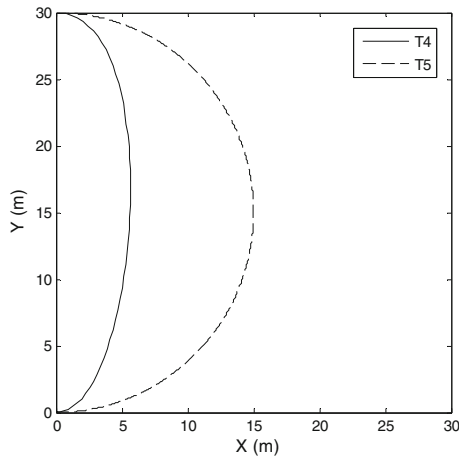
Trajectory	$\lambda$	$\max  \varphi_{\text{ed}} $
T1	$\lambda_1 = 0.98, \lambda_2 = 4.19$	0.23 rad
T2	$\lambda_1 = 10.0, \lambda_2 = 10.0$	0.59 rad
T3	$\lambda_1 = 1.0, \lambda_2 = 1.0$	0.53 rad



**Fig. 3** Simulation result of trajectory tracking control. **a** Motion trajectory of the bicycle. **b** Control inputs. **c** Errors of the system states

$\lambda = [0.98, 4.19]^T$  is determined by the PSO algorithm. T2 and T3 denote the trajectories whose parameters  $\lambda$  are given as  $[10.0, 10.0]^T$  and  $[1.0, 1.0]^T$ , respectively. From this simulation, it can be seen that we can retain





**Fig. 4** Simulation result of trajectory planning for  $[x_f, y_f, \theta_f]^T = [0, 30, \pi]^T$

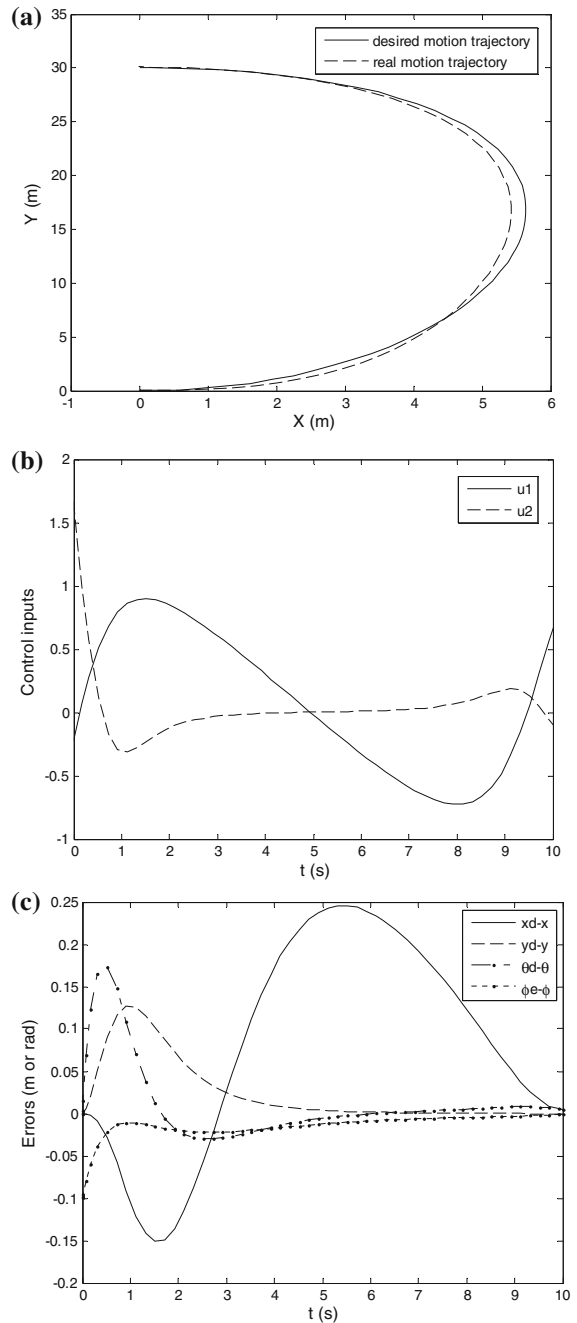
the balance of the bicycle around T1 more easily and reliably, since it has a smaller maximum of the desired roll angle.

Then, the trajectory tracking control is implemented, where T1 is chosen as the desired trajectory. The simulation result is shown in Fig. 3. Figure 3a shows the desired trajectory T1, as well as the resulting trajectory in the plane. Figure 3b shows the control inputs. Figure 3c shows the tracking errors of the system.

Figure 4 shows another simulation result of trajectory planning, in which the desired final pose is  $[x_f, y_f, \theta_f]^T = [0, 30, \pi]^T$ . The final time is  $t_f = 10s$ . T4 denotes the planned trajectory using the proposed method, and T5 is a circular trajectory that connects the initial pose with the final pose. Table 2 shows the parameters of the two trajectories in detail. The curvature of T5 is a constant; thus, the robot can track T5 with a constant linear velocity, and the desired equilibrium of the roll angle is also a constant, that is  $\varphi_{ed} = -0.15$ . From this simulation, it can be seen that the maximum roll angle's equilibrium of T4 is a bit larger than that of the circular trajectory T5, because the robot needs to make a bit sharper turn at the initial and final stages

**Table 2** Simulation comparison between two different trajectories

Trajectory	$\lambda$	$\max  \varphi_{ed} $
T4	$\lambda_1 = 1.87, \lambda_2 = 2.62$	0.17 rad
T5	$R = 15\text{ m (radius)}$	0.15 rad



**Fig. 5** Simulation result of trajectory tracking control. **a** Motion trajectory of the bicycle. **b** Control inputs. **c** Errors of the system states

when it moves along T4. However, T4 is much shorter than T5, such that energy can be saved when the robot tracks T4. Figure 5 shows the simulation result of the trajectory tracking control, where T4 is chosen as the desired trajectory.

## 6 Conclusions

In this paper, trajectory planning and tracking control have been solved comprehensively for a nonlinear, nonholonomic, and nonminimum phase bicycle robot. The planar motion trajectory of the system has been designed as a parameterized polynomial curve. The parameters of the polynomial curve have been calculated by the PSO algorithm to achieve an optimal desired trajectory, of which the maximum of the roll angle is minimal. A trajectory tracking controller has been designed for approximate tracking of desired trajectory while maintaining balance of the bicycle. The proposed method can generate a motion trajectory with superior performances, such as small roll angle and short path length. In addition, the designed controller can provide good tracking behaviors with good balance for tracking the planned trajectory. As a result, the moving performance of the bicycle robot is improved by the combination of the proposed trajectory planner and tracking controller.

**Acknowledgments** The authors would like to thank the financial support from the National Natural Science Foundation of China (Grant No.61175085 and 61175083)

## Appendix

We consider an implicit function defined by

$$F(t, z) = 0 \quad (25)$$

Assuming the existence of all derivatives we may need, we denote the partial derivatives of  $F$  with respect to  $t$  and  $z$  by  $F_t = \partial F / \partial t$  and  $F_z = \partial F / \partial z$ , respectively, and denote by  $F_t^r$  the  $r$ -th partial derivative with respect to  $t$ , that is  $\partial^r F / \partial t^r$ . Assuming  $F_z \neq 0$  and taking the total differential of (25), we get

$$z' = \frac{dz}{dt} = -\frac{F_t}{F_z} (F_z \neq 0) \quad (26)$$

Since a local extremum of a smooth function satisfies  $z' = 0$ , the equilibrium points  $P_e(t_e, z_e)$  of (25) are defined by

$$\begin{cases} F(t, z) = 0 \\ F_t(t, z) = 0 \end{cases} \quad (27)$$

Note that  $H_1(t, z)$  equals to zero by (27) at the equilibrium points  $P_e$ . Consider

$$H_r(t, z) = \frac{F_t^r}{F_z} \quad (r = 2, 3, \dots) \quad (28)$$

at the equilibrium points. If the first index  $r$  for which  $H_r(t, z) \neq 0$  is an odd number, the point at which  $H_r(t, z)$  is computed is not an extremum. If  $r$  is an even number, the equilibrium point is a minimum when  $H_r(t, z) < 0$  or a maximum when  $H_r(t, z) > 0$ .

## References

1. David, J., Limebeer, N., Sharp, R.S.: Bicycles, motorcycles and models. *IEEE Control Syst. Mag.* **26**(5), 34–61 (2006)
2. Astrom, K., Klein, R., Lennartsson, A.: Bicycle dynamics and control. *IEEE Control Syst. Mag.* **25**(4), 26–47 (2005)
3. Getz, N., Marsden, J.: Control for an autonomous bicycle. In: *Proceedings of IEEE International Conference on Robotics and Automation, Nagoya*, pp. 1397–1402 (1995)
4. Getz, N.: Dynamic inversion of nonlinear maps with applications to nonlinear control and robotics. Ph.D Dissertation, University of California at Berkeley, CA (1995)
5. Hauser, J., Saccon, A., Frezza, R.: Achievable motorcycle trajectories. In: *Proceedings of the IEEE Conference on Decision and Control, Atlantis*, pp. 3944–3949 (2004)
6. Frezza, R., Beghi, A.: Simulating a motorcycle driver. In: Kang, W., Xiao, M., Borges, C. (eds.) *New trends in Nonlinear Dynamics and Control and Their Applications*. Springer, New York (2003)
7. Yi, J., Song, D., Levandowski, A., Jayasuriya, S.: Trajectory tracking and balance stabilization control of autonomous motorcycle. In: *Proceedings of IEEE International Conference on Robotics and Automation, Florida*, pp. 2583–2589 (2006)
8. Meijaard, J.P., Popov, A.A.: Numerical continuation of solutions and bifurcation analysis in multibody systems applied to motorcycle dynamics. *Nonlinear Dyn.* **43**(1), 97–116 (2006)
9. Guo, L., Liao, Q., Wei, S.: Nonlinear stabilization of bicycle robot steering control system. In: *Proceedings of IEEE International Conference on Mechatronics and Automation, Chang Chun*, pp. 3185–3189 (2009)
10. Keo, L., Yamakita, M.: Controlling balancer and steering for bicycle stabilization. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis*, pp. 4541–4546 (2009)
11. Yetkin, H., Ozguner, U.: Stabilizing control of an autonomous bicycle. In: *Proceedings of Asian Control Conference, Istanbul*, pp. 1–6 (2013)
12. Soudbakhsh, D., Zhang, Y., Yi, J.: Stability analysis of human rider's balance control of stationary bicycles. In: *Proceedings of American Control Conference, Montreal*, pp. 2755–2760 (2012)

13. Tanaka, Y., Murakami, T.: A study on straight-line tracking and posture control in electric bicycle. *IEEE Trans. Ind. Electron.* **56**(1), 159–168 (2009)
14. Yamaguchi, T., Shibata, T., Murakami, T.: Self-sustaining approach of electric bicycle by acceleration control based backstepping. In: *Proceedings of 33rd Annual Conference of the IEEE Industrial Electronics Society, Taipei*, pp. 2610–2614 (2007)
15. Defoort, M., Murakami, T.: Second order sliding mode control with disturbance observer for bicycle stabilization. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice*, pp. 2822–2827 (2008)
16. Hwang, C., Wu H., Shih, C.: Autonomous dynamic balance of an electrical bicycle using variable structure underactuated control. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice*, pp. 3737–3743 (2008)
17. Consolini, L., Maggiore, M.: Control of a bicycle using virtual holonomic constraints. In: *Proceedings of IEEE Conference on Decision and Control, Atlanta*, pp. 5204–5209 (2010)
18. Gundes, A. N., Nanjangud, A.: Low-order simultaneous stabilization of linear bicycle models at different forward speeds. In: *Proceedings of American Control Conference, Washington, DC*, pp. 1840–1845 (2013)
19. Frezza, R., Beghi, A., Saccon, A.: Model predictive for path following with motorcycles: application to the development of the pilot model for virtual prototyping. In: *Proceedings of IEEE Conference on Decision and Control, Atlantis*, pp. 767–772 (2004)
20. Yi, J., Zhang, Y., Song, D.: Autonomous motorcycles for agile maneuvers, part II: control systems design. In: *Proceedings of IEEE Conference on Decision and Control, Shanghai*, pp. 4619–4624 (2009)
21. Saccon, A., Hauser, J., Beghi, A.: Trajectory exploration of a rigid motorcycle model. *IEEE Trans. Control Syst. Technol.* **20**(2), 424–437 (2011)
22. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks, Perth*, pp. 1942–1948 (1995)
23. Seifried, R.: Integrated mechanical and control design of underactuated multibody systems. *Nonlinear Dyn.* **67**(2), 1539–1557 (2012)
24. Soltanpour, M.R., Khooban, M.H.: A particle swarm optimization approach for fuzzy sliding mode control for tracking the robot manipulator. *Nonlinear Dyn.* **74**(1), 467–478 (2013)
25. Valle, Y.D., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.C., Harley, R.G.: Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Trans. Evol. Comput.* **12**(2), 171–195 (2008)
26. Wai, R.J., Chuang, K.L., Lee, J.D.: Online supervisory control design for maglev transportation system via total sliding-mode approach and particle swarm optimization. *IEEE Trans. Autom. Control* **55**(7), 1544–1559 (2010)
27. Tian, J., Gu, H.: Anomaly detection combining one-class SVMs and particle swarm optimization algorithms. *Nonlinear Dyn.* **61**(1), 303–310 (2010)
28. Hendzel, Z.: An adaptive critic neural network for motion control of a wheeled mobile robot. *Nonlinear Dyn.* **50**(4), 849–855 (2007)
29. Egerstedt, M., Hu, X., Stotskyk, A.: Control of mobile platforms using a virtual vehicle approach. *IEEE Trans. Autom. Control* **46**(11), 1777–1782 (2001)
30. Zhong, G., Kobayashi, Y., Hoshino, Y., Emaru, T.: System modeling and tracking control of mobile manipulator subjected to dynamic interaction and uncertainty. *Nonlinear Dyn.* **73**(1), 167–182 (2013)
31. Nunes, U., Bento, L.C.: Data fusion and path-following controllers comparison for autonomous vehicles. *Nonlinear Dyn.* **49**(4), 445–462 (2007)
32. Jian, C., Dixon, W.E., Dawson, M., McIntyre, M.: Homography-based visual servo tracking control of a wheeled mobile robot. *IEEE Trans. Robot.* **22**(2), 406–415 (2006)

Reproduced with permission of copyright owner. Further reproduction prohibited without permission.