Mixed Self-Balancing Two-Wheeled Mobile Platform

Hsiu-Ting Chang, Feng-Chun Tai, and Ching-Chih Tsai
Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan, ROC

***Abstract*—This paper presents a motion control method for a mixed self-balancing two-wheeled mobile platform used for autonomous service robots. Based on the dynamic mathematical modeling with nominal parameters, two backstepping sliding-mode controllers are designed to achieve desired speed tracking and yaw rate control, in order to accomplish trajectory tracking and stabilization. In particular, there are extra knee joints so as to maintain body balancing when encountering an uneven terrain. The mechatronic design of the robot uses OpenCR controllers and motors from Dynamixel, where the OpenCR controllers are coded by Arduino. In order to achieve speed tracking and yaw rate control for the robot, a kinematical trajectory motion generator and two nonlinear sliding-mode controllers are designed based on the nonlinear robot model, thus enabling the robot to move at desired speeds and rate commands. The feasibility and effectiveness of the proposed methods are well exemplified by conducting several simulations on the mobile platform moving along its working space. [1]**

Keywords: *motion control, mobile robotics, sliding-mode control, self-balancing.*

I. Introduction

In recent years, science and technology have been advancing in each passing day, and inventions and patents researched by different talents have appeared from all over the world. Therefore, automation has become a part of human industry that cannot be ignored. The benefits of automation can not only reduce personnel costs, but also maintain the quality of work. In the past years, industrial robots and human workers were independent entities operating separately; however, for the coming and future applications, this situation will no longer occur, because in future, robots will need to assist humans. To allow the robots "coexist" with humans, human–robot interaction needs to be taken into account in all steps of the robot design. Recently, both academia and industry have conducted extensive research on self-balancing two-wheeled platforms. They have been successfully applied to construct several autonomous service robots [1-10]. On the other hand, many researchers [3]-[10] have shown that the two-wheeled self-balancing platforms have gained many applications, including personal transportation vehicles, soccer games and etc. With the advent of modern control technology, such platforms with sophisticated control functions and safety features can be cost down so that they have highly potential to satisfy stringent requirements of various autonomous service robots with high linear speed.

[1] This work was supported in part by the MOST under Grant No. 109-2221-E-005 -066-MY2.

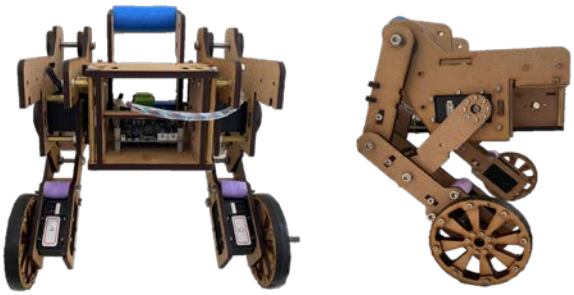

Fig.1. (a) Whole picture of the Self-balancing two-wheeled platform. (b) Sideview of the self-balancing two-wheeled platform.

EMIEW has been regarded as one of the most successful SBTWRs developed for working helpers. Hosoda *et al.* [2] detailed the basic design of this kind of human-symbiotic robot whose linear motion speed was up to 1.67m/sec; their EMIEW was designed based on a self-balancing two-wheeled structure. In addition to EMIEW, there have been many successful autonomous mobile robots based on the self-balancing two-wheeled base.

Since the self-balancing wheeled mobile robots have become increasingly important for applications in academia and industry, many interesting research topics remain to be done. In 2002, Segway began to product, after two years or so, the second-generation Segway was significantly improved. New Segway uses a handle bar to change motion directions. Moreover, the second-generation Segway has a wireless smart key controller, called Infokey, thereby providing an anti-theft function via remote control. Each Infokey has its own exclusive programming code to open and lock its corresponding Segway, namely that each Segway can be activated by its unique Infokey.

Two-wheel self-balancing can maintain a slim shape compared to three-wheels and four-wheels, and is easier to balance than a single wheel does. However, ordinary two-wheels are prone to tipping when there is a drop in the terrain. Therefore, the design of bird-shaped feet makes the platform more adaptable to uneven terrains.

The rest of this paper is outlined as follows. Sections II introduces the system design and mathematical modeling of the system. Section III introduces the motion control using backstepping sliding-mode control. Section IV uses the fuzzy basis function network to design the proposed intelligent adaptive motion control laws. In Section V, simulations are carried out to prove the feasibility and effectiveness of the proposed control methods. Section VI ends with the conclusions and future work of the paper.

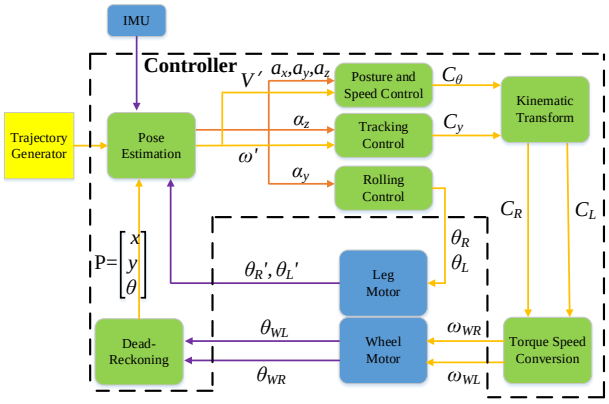

Fig.2. Block diagram of the control architecture for the robot.

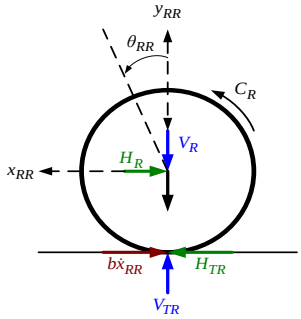

Fig.3. Free body diagram of the right wheel.

II. SYSTEM DESCRIPTION AND MODELING

A. Mechatronic Design of the Platform

Fig.1 displays the picture of the experimental autonomous service robot with the self-balancing two-wheeled platform. As shown in Fig. 1, this self-balancing two-wheeled platform is composed of one OpenCR controller with embedded inertial measurement unit (IMU), four Dynamixel motors, and one 12-volt sealed rechargeable lead-acid battery. The OpenCR controller works as the main controller, the IMU in the OpenCR controller is used to measure the rate and the angle of the platform inclination. In addition, the OpenCR controller also provides control actions for the yaw rate of the platform. Two optical encoders installed in the driving motors are employed to achieve dead-reckoning calculation of the platform.

B. Control Architecture

Fig. 2 explains the block diagram of the overall control architecture of the mobile platform. There are four main control modules: trajectory generator, motion command generator, posture and speed control and yaw rate control. The motion command generator provides linear and angular speed commands for the speed and yaw rate controllers for steering the robot to exactly follow the desired trajectories. The internal signal, pitch angle, is fed back to the posture and speed control module, and the motor feeds back the real readings of the robot speed and yaw rate. The actual speeds of mobile platform are also fed back to the posture and speed control module. Besides, the actual yaw rate is fed back to the yaw rate control module. The feedback signals from the IMU

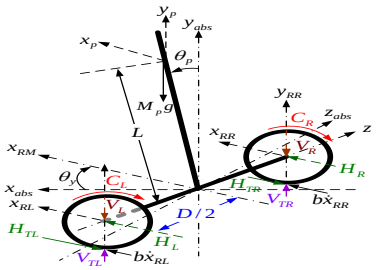

Fig.4. Free body diagram of the platform.

are utilized via the controller to maintain the robot balancing and posture control.

C. Nonlinear Mathematical Modeling

To steer the robot, mathematical model is a necessary such that the robot with the designed controllers can successfully achieve desired control objectives. The procedure to establish the mathematical model of the mobile platform is briefly described in the following. This model will include viscous friction, which depends on the moving speed of the platform, and static friction opposite to the moving direction. In order to simplify the derivation, Table 1 and Fig.4 list all the used symbols and their definitions. By using the Newtonian mechanism, one develops the following set of motion equations to describe the robot dynamics in the moving frame. The nonlinear system model of the robot is modified from [2] and described in the following state-space form.

$$\dot{\omega}_P = A_{43}\sin\theta_p + B_4 C_L + B_4 C_R + B_{45} f_{dp}, \tag{1}$$

$$\dot{\omega}_y = A_{66}\omega_y + B_{61} C_L + B_{62} C_R + B_{63} f_{dRL} + B_{64} f_{dRR}. \tag{2}$$

Defining the following six state variables

$$\mathbf{X} = [x_{RM} \quad v_{RM} \quad \theta_p \quad \omega_p \quad \theta_y \quad \omega_y]^T \tag{3}$$

one can easily derive the nonlinear state equation the robot in the following state-space form

$$\begin{bmatrix} \dot{x}_{RM} \\ \dot{v}_{RM} \\ \dot{\theta}_p \\ \dot{\omega}_p \\ \dot{\theta}_y \\ \dot{\omega}_y \end{bmatrix} = \begin{bmatrix} v_{RM} \\ A_{22}v_{RM} + A_{23}\sin\theta_p \\ \omega_p \\ A_{43}\sin\theta_p \\ \omega_y \\ A_{66}\omega_y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ B_2 & B_2 \\ 0 & 0 \\ B_4 & B_4 \\ 0 & 0 \\ B_6 & -B_6 \end{bmatrix} \begin{bmatrix} C_L \\ C_R \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ B_{23} & B_{24} & B_{25} \\ 0 & 0 & 0 \\ 0 & 0 & B_{45} \\ 0 & 0 & 0 \\ B_{63} & B_{64} & 0 \end{bmatrix} \begin{bmatrix} f_{dRL} \\ f_{dRR} \\ f_{dP} \end{bmatrix}$$

$$= \begin{bmatrix} v_{RM} \\ A_{22}v_{RM} + A_{23}\sin\theta_p \\ \omega_p \\ A_{43}\sin\theta_p \\ \omega_y \\ A_{66}\omega_y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ B_2 & B_2 \\ 0 & 0 \\ B_4 & B_4 \\ 0 & 0 \\ B_6 & -B_6 \end{bmatrix} \begin{bmatrix} C_L \\ C_R \end{bmatrix} + \begin{bmatrix} 0 \\ f_2 \\ 0 \\ f_4 \\ 0 \\ f_6 \end{bmatrix} \tag{4}$$

where $f_2 = B_{23} f_{dRL} + B_{24} f_{dRR} + B_{25} f_{dP}$, $f_4 = B_{45} f_{dP}$, $f_6 = B_{63} f_{dRL} + B_{64} f_{dRR}$;all the parameters are given by [2].From (4), if the robot has reached a steady-state inclination θ_p, i.e., $\dot{\theta}_p = 0$, then it will generate a steady-state linear speed expressed by

$$v_{SS} = \frac{B_2 A_{43} - A_{23} B_6}{A_{22} B_6} \sin\theta_p \Big|_{\theta_p = 0} \tag{5}$$

This steady-state linear speed aims to move the robot without falling down. With the use of the transformation of the torques C_y and C_θ into the wheel torques C_L and C_R

$$\begin{bmatrix} C_L \\ C_R \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} C_\theta \\ C_y \end{bmatrix} \tag{6}$$

(4) becomes

$$\begin{bmatrix} \dot{x} \\ \dot{v} \\ \dot{\theta}_p \\ \dot{\omega}_p \\ \dot{\theta}_y \\ \dot{\omega}_y \end{bmatrix} = \begin{bmatrix} v \\ A_{22}v + A_{23}\sin\theta_p \\ \omega_p \\ A_{43}\sin\theta_p \\ \omega_y \\ A_{66}\omega_y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ B_2 & B_2 \\ 0 & 0 \\ B_4 & B_4 \\ 0 & 0 \\ B_6 & -B_6 \end{bmatrix} \begin{bmatrix} C_L \\ C_R \end{bmatrix} + \begin{bmatrix} 0 \\ f_2 \\ 0 \\ f_4 \\ 0 \\ f_6 \end{bmatrix} \tag{7}$$

Observing (7) reveals that there are two independent subsystems: one is concerned with the mobile inverted pendulum subsystem regarding the self-balancing of the platform, i.e.

$$\begin{bmatrix} \dot{x} \\ \dot{v} \\ \dot{\theta}_p \\ \dot{\omega}_p \end{bmatrix} = \begin{bmatrix} v \\ A_{22}v + A_{23}\sin\theta_p \\ \omega_p \\ A_{43}\sin\theta_p \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \\ 0 \\ B_4 \end{bmatrix} C_\theta + \begin{bmatrix} 0 \\ f_2 \\ 0 \\ f_4 \end{bmatrix} \tag{8}$$

and the other is the yaw motion system model regarding the yaw motion, i.e.,

$$\begin{bmatrix} \dot{\theta}_y \\ \dot{\omega}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & A_{66} \end{bmatrix} \begin{bmatrix} \theta_y \\ \omega_y \end{bmatrix} + \begin{bmatrix} 0 \\ B_6 \end{bmatrix} (C_y + \bar{f}_6),\ \bar{f}_6 = f_6 / B_6, B_6 > 0 \tag{9}$$

From (8) and (9), it indicates that two controllers for C_θ and C_y can be synthesized independently from each other and then combined together to accomplish the control goal.

D. Torque-To-Speed Conversion

Owing to the fact the DSC controller is designed for carrying out the speed tracking control of the DC motors, not for direct torque control, the torque obtained in the previous section cannot be straightforward applied. In addition, a torque-to-speed conversion is required to calculate the corresponding speed commands for each motor. In doing so, one respectively finds the torques for the right and left in the following.

$$\tau_R(k) = \frac{1}{2}\left(C_\theta(k) + C_y(k)\right) \tag{10}$$

$$\tau_L(k) = \frac{1}{2}\left(C_\theta(k) - C_y(k)\right) \tag{11}$$

Note that, for the sake of low cost and easy implementation, the digital controller does not directly use torque controller's structure with current feedback, but the DSC controller employs PWM (pulse-width-modulation) driving technology to fulfil the speed tracking. This is can easily be done using Newton's second law for rotation, thus giving which leads to the following digital velocity commands for the left wheel using the backward difference proceeding with the same procedure for digital velocity commands for the right wheel yields.

$$\tau_L = J_L \cdot \alpha = J_L \cdot \dot{\omega}_L \tag{12}$$

$$\tau_R = J_R \cdot \alpha = J_R \cdot \dot{\omega}_R \tag{13}$$

The results reveal that once both the torques for the right and left wheels have been obtained; the digital velocity commands for both wheels will be immediately computed. The commands are then inputted to the two independent, digital open-loop speed loops, thereby accomplishing the self-balancing and rotation actions.

$$\omega_L(k) = \omega_L(k-1) + \frac{T \cdot \tau_L(k)}{J_L} \tag{14}$$

$$\omega_R(k) = \omega_R(k-1) + \frac{T \cdot \tau_R(k)}{J_R} \tag{15}$$

E. Odometry

In general, the pose (position and orientation) of the two-wheeled platform is written by the following vector

$$P = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{16}$$

By the odometry technique, the robot pose can be estimated from a known position by integrating the movement. Given a affixed sampling time Δt , the incrementally travelling distances and direction (Δx , Δy , $\Delta\theta$) are expressed as follows;

$$\Delta x = \Delta s \cos\left(\theta + \Delta\theta/2\right) \tag{17}$$

$$\Delta y = \Delta s \sin\left(\theta + \Delta\theta/2\right) \tag{18}$$

$$\Delta\theta = \frac{\Delta s_R - \Delta s_L}{b} \tag{19}$$

$$\Delta s = \frac{\Delta s_R + \Delta s_L}{2} \tag{20}$$

The path travelled in the last sampling interval time are (Δx , Δy , $\Delta\theta$) ; ΔS_L and ΔS_R are the travelling distances about the right and left wheel respectively; b is the distance between the two wheels of two-wheeled differential-driving robot. Thus,

$$\begin{aligned} p' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} &= p + \begin{bmatrix} \Delta s \cos\left(\theta + \Delta\theta/2\right) \\ \Delta s \sin\left(\theta + \Delta\theta/2\right) \\ \Delta\theta \end{bmatrix} \\ &= \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cos\left(\theta + \Delta\theta/2\right) \\ \Delta s \sin\left(\theta + \Delta\theta/2\right) \\ \Delta\theta \end{bmatrix} \end{aligned} \tag{21}$$

$$\begin{aligned} p' &= f\left(x, y, \theta, \Delta s_R, \Delta s_L\right) \\ &= \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_R + \Delta s_L}{2} \cos\left(\theta + \frac{\Delta s_R - \Delta s_L}{2b}\right) \\ \frac{\Delta s_R + \Delta s_L}{2} \sin\left(\theta + \frac{\Delta s_R - \Delta s_L}{2b}\right) \\ \frac{\Delta s_R - \Delta s_L}{b} \end{bmatrix} \end{aligned} \tag{22}$$

By using the relations about ΔS_L and ΔS_R both in (19) and (20), one further obtains the basic equation for odometer position update (for differential drive robots).

As discussed earlier, the odometer position updates can give only rough estimates of the actual positions. It is owing to integration errors of the uncertainties of p and the motion errors during the incremental motion. Both ΔS_L and ΔS_R is the position error based on odometer integration grow with time.

III. TRAJECTORY TRACKING

A. Sliding-Mode Yaw Rate Controller Design

This subsection proposes the sliding-mode yaw rate controller to control the self-balancing two-wheeled robot to accurately track the virtual angular velocity command ϕ_3 .The yaw rate controller can be designed according to the following decoupled and simplified yaw motion model

$$\dot{\omega} = f_{23}(x) + \frac{R \cdot b}{G_\alpha} C_y$$

Define the sliding surface S_η by

$$S_\eta = \omega - \phi_3 \tag{23}$$

Differentiating S_η with respect to time gives

$$\dot{S}_\eta = \dot{\omega} - \dot{\phi}_3 = f_{23}(x) + \frac{R \cdot b}{G_\alpha} C_y - \dot{\phi}_3 \tag{24}$$

The control goal is to find a sliding-mode control law for C_y such that $S_\eta \to 0$ in finite time. Thus, let the yaw rate control law be

$$C_y = \frac{G_\alpha}{R \cdot b}\left[-f_{23}(x) + \dot{\phi}_3 - K_{\eta\omega 1}\,\mathrm{sgn}(S_\eta) - K_{\eta\omega 2} S_\eta \right] \tag{25}$$

such that

$$\dot{S}_\eta = -K_{\eta\omega 1}\,\mathrm{sgn}(S_\eta) - K_{\eta\omega 2} S_\eta \tag{26}$$

To show that $S_\eta \to 0$ in finite time, one proposes the Lyapunov function $V_2 = S_\eta^2/2$ whose time derivative is

$$\dot{V}_2 = S_\eta \dot{S}_\eta = -K_{\eta\omega 1}|S_\eta| - K_{\eta\omega 2} S_\eta^{\,2} \le -K_{\eta\omega 1}|S_\eta| \tag{27}$$

This indicates that $S_\eta \to 0$ in finite time. In practice, the signum function will be replaced by a saturation function to avoid the chattering phenomenon.

B. Aggregate Hierarchical Sliding-Mode Speed Control

The aggregate hierarchical sliding-mode control method approach in [21] is employed to find a torque control law for C_θ by using the following simplified and coupled 2-state equation.

$$\begin{bmatrix} \dot{\omega}_\alpha \\ \dot{v} \end{bmatrix} = \begin{bmatrix} f_{21}(x) \\ f_{22}(x) \end{bmatrix} + \begin{bmatrix} g_{21}(x) \\ g_{22}(x) \end{bmatrix} C_\theta \tag{28}$$

In order to develop the decoupling sliding-mode controller, we propose the two first-layer sliding surfaces

$$S_\alpha = \eta_\alpha = \omega_\alpha - \phi_1 = \omega_\alpha - (-K_\alpha \alpha) \tag{29}$$

$$S_v = \eta_v = v - \phi_2 \tag{30}$$

Then the second-layer sliding surface is proposed by

$$S_1 = r_1 S_\alpha + r_2 S_v \tag{31}$$

where r_1 and r_2 are two real parameters . To construct the decoupling sliding-mode control law such that S_1, S_v, $S_\alpha \to 0$ as $t \to \infty$, one takes the time derivatives of the second-layer sliding surface to be zero, i.e.,

$$\begin{aligned} \dot{S}_1 &= r_1 \dot{S}_v + r_2 \dot{S}_\alpha \\ &= r_1\left[f_{22}(x) + g_{22}(x) C_\theta - \dot{\phi}_2 \right] \\ &\quad + r_2\left[f_{21}(x) + g_{21}(x) C_\theta - K_\alpha \dot{\alpha}_1 \right] \\ &= \left[r_1 f_{22}(x) + r_2 f_{21}(x) \right] + \left[r_1 g_{22}(x) + r_2 g_{21}(x) \right] C_\theta \\ &\quad + \left[r_2 K_\alpha \dot{\alpha} - r_1 \dot{\phi}_2 \right] \end{aligned} \tag{32}$$

The control law for C_θ is chosen by

$$C_\theta = \frac{1}{r_1 g_{22}(x) + r_2 g_{21}(x)} \begin{bmatrix} -r_1 f_{22}(x) - r_2 f_{21}(x) - r_2 K_\alpha \dot{\alpha} \\ + r_1 \dot{\phi}_2 - K_{S1}\,\mathrm{sgn}(S_1) - K_{S2} S_1 \end{bmatrix} \tag{33}$$

which leads to $\dot{S}_1 = -K_{S1}\,\mathrm{sgn}(S_1) - K_{S2} S_1$. The fact that S_1 approaches zero in finite time can be easily shown by proposing the Lyapunov function $V_3 = S_1^2/2$ .Then, $\dot{V}_3 = S_1 \dot{S}_1 = -K_{S1}|S_1| - K_{S2} S_1 \le 0$. From(33), it easily implies that $S_1, S_v, S_\alpha \to 0$ in finite time.

C. Motion Commands Generator

This section presents a motion commands generator for the self-balancing two-wheeled mobile robot. The proposed motion command generator will be proven to be global asymptotically stable. In the sequel, the motion command generator is developed without consideration of the dynamic effects. The kinematical model of the mobile robot under the assumption of pure rolling is governed by:

$$\dot{q} = S(q) \cdot \mu \tag{34}$$

, where $q(t) = [x\ y\ \theta]^T \in R^3$ and $x,\ y,\ \theta \in R^1$ denote the actual posture of the AMR; $\mu = [v\ \omega]^T$ where v is the linear velocity and ω is the angular velocity. The matrix S(q) is giving by

$$S(q) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \tag{35}$$

The motion commands generation problem is to find a global motion commands generation so that the AMR can asymptotically follow the desired posture trajectories written as:

$$\dot{q}_r = S(q_r) \cdot \mu_r \tag{36}$$

where $q_r(t) = [x_r\ y_r\ \theta_r]^T$ and $\mu_r = [v_r\ \omega_r]^T$. We define the errors between the actual and desired postures $\tilde{x},\ \tilde{y},\ \tilde{\theta}$ as:

$$\begin{cases} \tilde{x} = x - x_r \\ \tilde{y} = y - y_r \\ \tilde{\theta} = \theta - \theta_r \end{cases} \cdot \tag{37}$$

The tangential error $e_1(t)$, the normal error $e_2(t)$ and the orientation error $e_3(t)$ can be written as the following matrix:

$$\begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{bmatrix} \tag{38}$$

Differentiating the error with respect to time obtains:

$$\begin{cases} \dot{e}_1 = \omega e_2 - v + v_r \cos e_3 \\ \dot{e}_2 = -\omega e_1 + v_r \sin e_3 \\ \dot{e}_3 = \omega_r - \omega \end{cases} \tag{39}$$

Notice that all the transformed errors, $e_1(t)$ $e_2(t)$ and $e_3(t)$, are continuous and bounded, the original errors, $\tilde{x}$, $\tilde{y}$, $\tilde{\theta}$ are bounded and continuous. This means that

$$\begin{cases} \lim_{t\to\infty} e_1(t) = 0 \\ \lim_{t\to\infty} e_2(t) = 0 \\ \lim_{t\to\infty} e_3(t) = 0 \end{cases}, \text{ if and only if } \begin{cases} \lim_{t\to\infty} \tilde{x}(t) = 0 \\ \lim_{t\to\infty} \tilde{y}(t) = 0 \\ \lim_{t\to\infty} \tilde{\theta}(t) = 0 \end{cases}.$$

Define a new auxiliary variable $\bar{e}_3(t)$:

$$\bar{e}_3(t) = e_3 + \alpha e_2 \tag{40}$$

, where $\alpha \neq 0$ .Taking the time derivative of $\bar{e}_3(t)$ yields:

$$\begin{aligned} \dot{\bar{e}}_3(t) &= \dot{e}_3 + \alpha \dot{e}_2 \\ &= \omega_r - \omega + \alpha(-\omega e_1 + v_r \sin e_3) \end{aligned} \tag{41}$$

To stabilize the system, the control law v and ω are selected as follows:

$$\begin{cases} v = v_{ref} = k_1 e_1 + v_r \cos e_3 \\ \omega = \omega_{ref} = \dfrac{1}{1+\alpha e_1}(k_2 \bar{e}_3 + v_r \dfrac{\sin e_3}{\bar{e}_3} e_2 + \alpha v_r \sin e_3 + \omega_r) \end{cases} \tag{42}$$

The closed-loop error system can be stated as follows:

$$\begin{aligned} \dot{e}_1 &= -k_1 e_1 + \omega e_2 \\ \dot{e}_2 &= -\omega e_1 + v_r \sin e_3 \\ \dot{\bar{e}}_3 &= -k_2 \bar{e}_3 - v_r \frac{\sin e_3}{\bar{e}_3} e_2 \end{aligned} \tag{43}$$

To show the asymptotical stability of the closed-loop error system, we find the following racially unbounded Lyapunov function candidate:

$$V = \frac{1}{2}(e_1^2 + e_2^2 + \bar{e}_3^2) \tag{44}$$

which has it's time derivative along its motion trajectory:

$$\begin{aligned} \dot{V} &= e_1(\omega e_2 - k_1 e_1) + e_2(-\omega e_1 + v_r \sin e_3) + \bar{e}_3(-k_2 \bar{e}_3 - v_r \frac{\sin e_3}{\bar{e}_3} e_2) \\ &= -k_1 e_1^2 - k_2 \bar{e}_3^2 \le 0 \end{aligned} \tag{45}$$

Hence, $\dot{V}$ is semi-negative definite. According to Barbalat's lemma and the LaSalle's principle, $e_1(t)$ and $\bar{e}_3(t)$ approach zero as time goes to infinity. Here $\dot{\bar{e}}_3 = -k_2 \bar{e}_3 - v_r \frac{\sin e_3}{\bar{e}_3} e_2$ and $\lim_{t\to\infty} \bar{e}_3(t) = 0$, if we can prove $\lim_{t\to\infty} \frac{\sin e_3}{\bar{e}_3} \neq 0$ then $e_2(t)$ approach to zero when time goes to infinite. Then, we show:

TABLE 1.
ALL THE PARAMETERS USED FOR SIMULATIONS.

Symbol (unit)	Parameter and variable name	Value
I_{xx}, I_{yy}, I_{zz}	Moment of inertia of the pendulum body with respect to the axis, respectively.	I_{xx}=0.1586 I_{yy}=0.1586 I_{zz} =0.0135
v_r (*m/sec*)	Reference linear velocity	0.25
ω_r (*m/sec*)	Reference angular velocity	0.5
R (*m*)	Radius of both wheels	0.1
c_x, c_z	The center of mass of the vehicle is at Coordinate	c_z=0.2414
τ_R, τ_L	Input torque applied to the right motor and the left motor	
I_{wa}, I_{wd} ($Kg \cdot m^2$)	Moment of inertia of a wheel about its axis and about a diameter respectively	I_{wa}=0.003 I_{wd}=0.0012
ϕ_R, ϕ_L (*rad*)	Angles of the right and left wheels	
θ (*rad*)	Yaw angle	
M_b (*Kg*)	Mass of the pendulum body	2.6
M_w (*Kg*)	Mass of the each wheel	0.023
α (*rad*)	Tilt angle of the wheeled inverted pendulum	
b (*m*)	Half of the distance between the two driving wheels	0.1

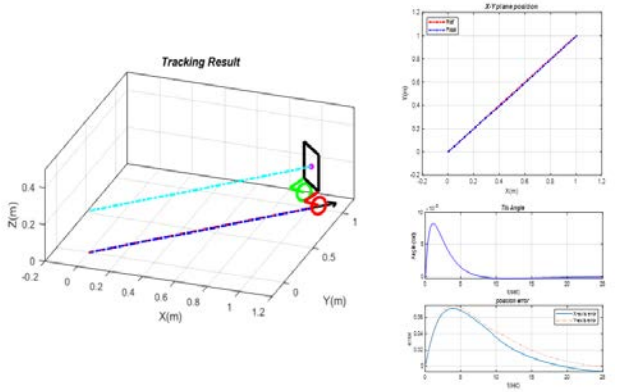

Fig.5. Simulation results of the line trajectory tracking.

$$\begin{aligned} \lim_{t\to\infty} \frac{\sin e_3}{\bar{e}_3} &= \lim_{t\to\infty} \frac{-k_2(e_3 + \alpha e_2) - v_r \dfrac{\sin e_3}{e_3 + \alpha e_2} e_2 + \alpha\omega e_1 - \alpha v_r \sin e_3}{-k_2(e_3 + \alpha e_2) - v_r \dfrac{\sin e_3}{e_3 + \alpha e_2} e_2} \\ &= \frac{k_2 + \alpha v_r}{k_2} \neq 0, \quad if \ k_2 \neq -\alpha v_r \end{aligned}$$

The proof assumes that $\lim_{t\to\infty} e_2(t)$ approaches zero fast that $e_3(t)$ approaches zero. This assumption always holds. Then $\lim_{t\to\infty} e_3(t) = 0$.

IV. SIMULATIONS AND EXPERIMENTAL RESULTS DISCUSSION

Four simulations, including sliding-mode yaw rate control, sliding-mode speed control, motion command generator and overall motion control system, are conducted to examine the effectiveness and merit of the proposed motion control method in this section. Table 1 shows the coefficients of the two-wheeled self-balancing robot for simulations. Fig. 5 presents the simulated results of the overall system tracking a straight line, and Fig.6 presents the simulated results of the

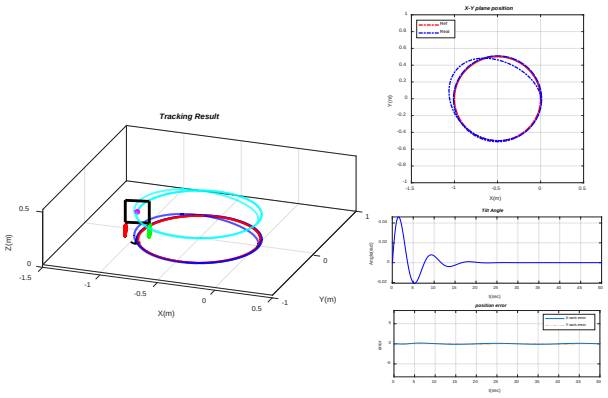

Fig.6. Simulation results of the circle trajectory tracking.

overall system tracking a circular trajectory. As can be seen in Figs. 5 and 6, the tilt angles of the robot for both simulations are eventually maintained at zeros, namely that the robot move without falling down. The simulations results confirm the feasibility and effectiveness of the proposed overall system controller.

V. Conclusions and Future Work

This paper has presented a motion control method of a mixing self-balancing two-wheeled mobile platform used for autonomous service robots. The mathematical modeling of the platform has been well established in a state-space framework. With the model, the motion commands generator generates linear and angular speed commands for the platform. By decomposing the overall system of the robot into the corresponding two subsystems: yaw motion and mobile inverted pendulum, we have synthesized sliding-mode speed and yaw rate controllers to achieve expected linear and angular speeds tracking. The simulations results show that the proposed controllers is useful and effective in providing proper control actions to steer the robot at desired speeds. Future work will be conducting several experiments to show the applicability of the proposed attitude and velocity and yaw rate controllers.

Acknowledgements

The authors deeply acknowledge financial support from the MOST, Taiwan, ROC, under contract MOST 109-2221-E-005-066-MY2.

Rerferences

[1] D. Voth, “Segway to the future,” Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications] vol.20, no.3, pp.5 – 8, May-June 2005.

[2] Y.Hosoda, S. Egawa, J. Tamamoto, K. Yamamoto, R.Nakamura and M.Togami “Basic design of human-symbiotic robot EMIEW,” in Pro. IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, pp.5079-5084, Oct. 9 - 15, 2006.

[3] S.C. Lin, System design, modeling and control of self-balancing human transportation vehicles, Ph.D Dissertation, Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan, July 2008.

[4] M. Sasaki, N. Yanagihara, O. Matsumoto, and K. Komoriya, “Steering control of the personal riding-type wheeled mobile platform (PMP),” in Proc. 2005 IEEE International Conference on Intelligent Robots and Systems, pp.1697-1702, 2005.

[5] F. Grasser, A.D’Arrigo, and S. Colombi, “JOE: A Mobile, Inverted Pendulum,” IEEE Trans. Industrial Electronics, vol.49, no.1, pp.107-114, February 2002.

[6] K. Pathak, J. Franch, and S. K. Agrawal, “Velocity and position control of a wheeled inverted pendulum by partial feedback linearization,” IEEE Transactions on Robotics, vol.21, no.3, pp.505-513, June 2005.

[7] Y.-S. Ha and S. Yuta, “Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot,” Robotics and Autonomous Systems, vol.17, pp. 65-80, 1996.

[8] C. C. Tsai, C. K. Chan and S. H. Wang , “Nonlinear Slide-Mode Motion Control of a Self-balancing Autonomous Service Robot,” Proc. of 2009 International Conference on Service and Interactive Robots, Taipei, Taiwan, August 6-7, 2009.

[9] A. Salerno, and J. Angeles, “The control of semi-autonomous two-wheeled robots undergoing large payload-variations,” in Proc. ICRA’04, vol.2, pp.1740-1745, Apr 26-May 1, 2004.

[10] C. C. Tsai, C. K. Chan, and Y. H. Fan, “Planned Navigation of a self-balancing autonomous service robot,” International Conference on Advanced Robots and its Social Impacts, Taipei, Taiwan, Aug. 2008.

[11] S. Y. Ju, *Trajectory tracking of a self-balancing two-wheeled robot using backstepping sliding-mode control and fuzzy basis function networks.* M. S. Thesis, Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan, July, 2010.