Design and Implementation of Fuzzy Control on a Two-Wheel Inverted Pendulum

Cheng-Hao Huang, Wen-June Wang, *Fellow, IEEE*, and Chih-Hui Chiu

***Abstract*—This paper introduces the design and implementation of a two-wheel inverted pendulum (TWIP) system with a fuzzy control scheme and the system-on-a-programmable-chip (SoPC) technology. The control scheme includes three kinds of fuzzy controls which are the fuzzy balanced standing control (FBSC), the fuzzy traveling and position control (FTPC), and the fuzzy yaw steering control (FYSC). Based on the Takagi–Sugeno fuzzy model of the TWIP, the FBSC is a structure of a parallel distributed compensator solved by the linear matrix inequality approach. Based on the motion characteristic of the TWIP, the FTPC and the FYSC are designed with Mamdani architecture if-then rules. Furthermore, the fuzzy control scheme for the real TWIP is implemented into an SoPC development board with an embedded reduced-instruction-set-computer soft-core processor and user intellectual property modules. Both the computer simulations and practical experiments demonstrate the effectiveness of the proposed control scheme.**

***Index Terms*—Field-programmable gate array (FPGA), fuzzy control, parallel distributed compensation (PDC), system on a programmable chip (SoPC), two-wheel inverted pendulum (TWIP).**

I. Introduction

A TWO-WHEEL INVERTED PENDULUM (TWIP) is a personal pendulum vehicle with two wheels, and the two wheels are attached on two sides of its chassis, respectively. Numerous studies have developed models and moving and balance control designs for a TWIP [1], [4]–[7], [17], [19], [20], [22], [30]. Baloh and Parent [1] concentrated on the dynamic modeling and model identification of a TWIP for an urban transportation system. Chiu [4] presented an adaptive output recurrent cerebellar model articulation controller to achieve the balanced standing control for the TWIP. Fiacchini *et al.* [5] proposed a physical dynamic model of a personal pendulum vehicle and developed two controllers (linear and nonlinear) for the balanced standing control. Grasser *et al.* [6] derived another dynamic model by using a Newtonian approach and designed two decoupled state-space controllers for stabilizing the pendulum and acting the yaw rotation, respectively. Ha and Yuta [7] presented a dynamic model of a TWIP using the Lagrange equation and designed a linear state feedback and feedforward controller for posture control and velocity control. Furthermore, the controller and the model of the TWIP proposed by Ha and Yuta [7] were continuously developed to perform baggage transportation and navigation tasks [22]. Nawawi *et al.* [19] described a hardware design of the TWIP and implemented a pole-placement controller for its stabilization. Tsai *et al.* [30] presented an adaptive control using radial basis-function neural networks for a two-wheeled self-balancing scooter. Additionally, Li and Luo [17] proposed the adaptive robust dynamic balance and motion controls for the TWIP. Pathak *et al.* [20] analyzed the dynamics of a TWIP and then applied the partial feedback linearization concept for its velocity and position control design. Unfortunately, there were only theoretical simulations in the aforementioned two papers [17] and [20], but the real TWIP practical performance was absent.

In the rapid development of the silicon industry, the field-programmable gate array (FPGA) chip and the system-on-a-programmable-chip (SoPC) technology have gradually emerged as a major revolution of mechanical and electrical integration for robotic applications [8]–[10], [12], [13], [15], [19], [21], [23], [31]. A SoPC development board, which contains a high capacity of the FPGA chip, can efficiently integrate the embedded soft-core processor and user intellectual property (IP) modules; therefore, the SoPC technology combines the high-performance hardware user IP modules for implementing the signal processing of peripherals and the computation-intensive software programs for realizing the control laws in the embedded processor [8], [9], [12], [15], [21]. With the benefit of high design flexibility, the SoPC technology significantly provides a feasible and powerful solution to develop the robotic systems.

In recent years, fuzzy control designs based on the Takagi–Sugeno (T–S) fuzzy model have been applied to diverse applications, such as the pendubots [2], [16], robot manipulators [18], trailers [24], [26], [27], hovercrafts [25], and helicopters [28]. Apparently, the fuzzy control has been widely and successfully applied to many nonlinear systems. Generally, a nonlinear system can be transformed into a T–S fuzzy model, and then, the parallel distributed compensation (PDC) fuzzy controller design is accomplished using linear matrix inequality (LMI) approaches [11], [29].

This paper applies the SoPC technology to implement a practical TWIP. A real TWIP is fabricated, and three fuzzy controllers are designed. The first controller is for the balanced standing control, the second is for the traveling and position control, and the last is for the yaw steering control. First, the balanced standing control is constructed as a PDC controller by

Manuscript received February 5, 2010; revised May 12, 2010 and June 28, 2010; accepted August 1, 2010. Date of publication August 23, 2010; date of current version June 15, 2011. This work was supported by National Science Council, Taiwan, under Grant NSC 98-2221-E-008-117-MY3.

C.-H. Huang and W.-J. Wang are with the Department of Electrical Engineering, National Central University, Jhongli 32001, Taiwan (e-mail: chhuang@ee.ncu.edu.tw; wjwang@ee.ncu.edu.tw).

C.-H. Chiu is with the Department of Electrical Engineering, Yuan Ze University, Jhongli 32003, Taiwan (e-mail: chchiu@saturn.yzu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2010.2069076

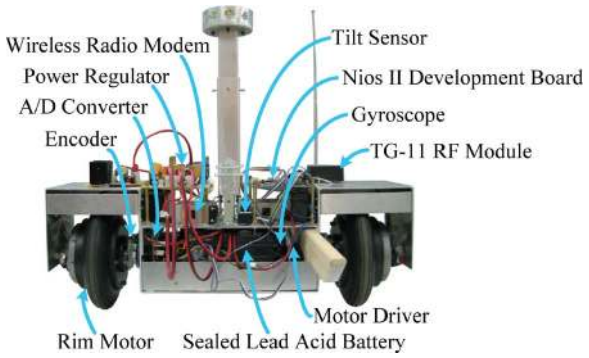

Fig. 1. Photograph of the TWIP.

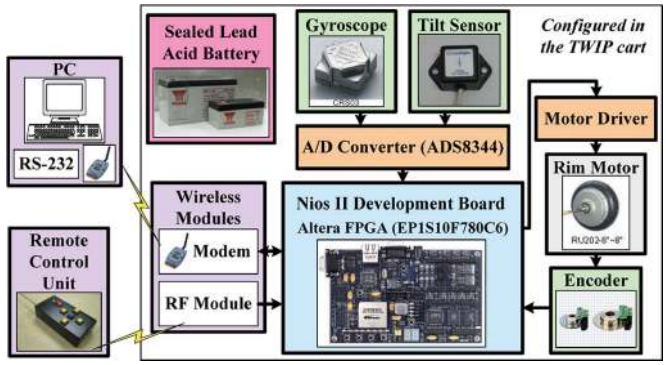

Fig. 2. Hardware configuration of the TWIP system.

using the T–S fuzzy approach [11], [29]. Based on the motion characteristic of the TWIP, this paper uses the Mamdani architecture if-then rules as in [3] and [14] to realize the traveling and position control and the yaw steering control. Here, the computer simulations are given to illustrate the control design ideas. Emphatically, the method in this paper is a hybrid control of the T–S fuzzy PDC approaches and Mamdani fuzzy control. As to the authors' best understanding, although the previous papers [1], [4]–[7], [17], [19], [20], [22], [30] have studied the control of the TWIP, the hybrid fuzzy control scheme proposed in this paper is still an innovative idea.

The proposed control scheme, which includes the aforementioned three fuzzy controllers, is embedded into the soft-core processor on the SoPC development board. The hardware/software codesign process of the SoPC is proposed to establish the fabricated TWIP such that the TWIP can achieve the mentioned three control goals. Finally, several practical experiments are done to verify the effectiveness of the proposed control scheme.

The rest of this paper is organized as follows. Section II introduces the SoPC-based hardware architectures of the real TWIP. Section III then describes the dynamics of the TWIP and the equivalent T–S fuzzy model. Next, Section IV summarizes the fuzzy control scheme along with associated simulations for the TWIP. In Section V, the software program design of the embedded processor is illustrated, and the practical experimental results are given to evaluate the proposed control scheme. Section VI finally concludes this paper.

II. SoPC-Based Hardware Architecture of the TWIP

This section introduces the SoPC-based hardware design to establish a practical TWIP system. Fig. 1 shows a photograph of the fabricated TWIP in which the hardware configuration is described in Fig. 2. The main hardware designs are introduced in the following section.

A. Hardware Construction

Fig. 3(a) shows the 3-D frame of the TWIP. The pendulum pole length is adjustable between 30 and 60 cm. The size of the cart platform is 63.2-cm long, 15-cm wide, and 18-cm high. The mechanism is made of aluminum sheets. The SoPC development board, circuit boards, dynamic sensors, wireless modules, and batteries are placed on the TWIP platform.

The TWIP actuators are two 24-V dc rim motors (RU202-8″) which are manufactured by the Elebike Company, Ltd. A rim motor is characterized by the coils that are mounted around the axle bearing and sealed as a wheel. Compared with common motor mechanisms, this design can decrease the shear strength effort. Moreover, the rotation of the rim motor is controlled by a pulse-width-modulation (PWM) signal from the SoPC development board.

Table I lists three dynamic sensors which are connected to the SoPC development board. The tilt sensor and the gyroscope are used to measure the inclination angle and angular velocity of the TWIP platform, respectively. Since the rim motor has no encoder function, one rotary encoder is installed in each rim motor to calculate the rotary speed of the wheels.

As shown in Fig. 2, the auxiliary equipment in the TWIP are wireless radio modems and a wireless radio frequency (RF) kit. The two wireless radio modems facilitate the communication between the TWIP and a personal computer (PC). Additionally, the TWIP system can also be controlled by a four-button remote control unit.

B. FPGA-Based SoPC Development Board

The adopted SoPC platform in this paper is the Nios II development board in which the FPGA chip is the Altera Stratix EP1S10F780C6. The FPGA provides 10 570 logic elements (LEs), 426 user I/O pins, 6 DSP blocks, 920-kb random-access memory (RAM), 6 phase-locked loops (PLLs), and a Nios II 32-b embedded reduced-instruction-set-computer (RISC) soft-core processor. A 50-MHz quartz oscillator is used as the system clock to supply the FPGA chip in the Nios II development board. Fig. 4 shows the internal architecture of the FPGA which comprises user IP modules and a Nios II embedded processor IP. The user IP modules (custom logic) mainly include an analog-to-digital converter (ADC) transformation circuit, a quadrature encoder pulse (QEP) detection circuit, a recommended standard 232 (RS-232) communication circuit, and a PWM generation circuit. They are all developed by very high speed integrated circuit hardware description language (VHDL) under the SoPC environment. The functions of the user IP modules are described in detail in Appendix A. Furthermore, the main control algorithm can be implemented by

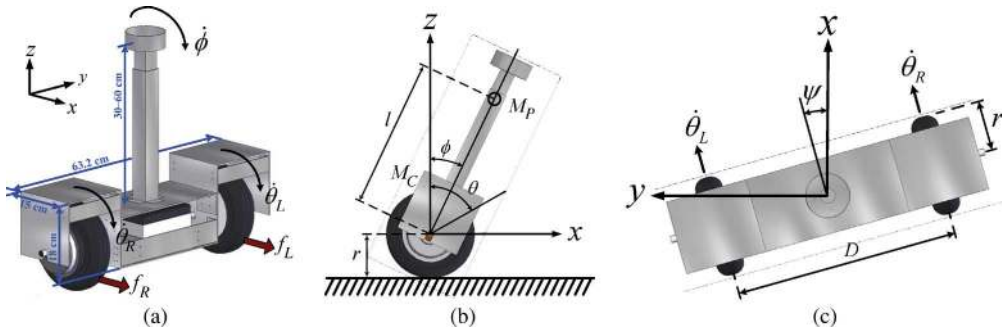

Fig. 3. TWIP coordinate with geometric parameters. (a) Slanted view. (b) Side view. (c) Top view.

TABLE I
DYNAMIC SENSORS INSTALLED IN THE TWIP

	Tilt sensor	Gyroscope	Encoder
Manufacturer (Model No.)	Crossbow Technology (CXTA01)	Silicon Sensing System Japan (CRS03-05)	Agilent Technologies (HEDL-6540#B13)
Supply voltage	9V	5V	5V
Measurable range	[−75°, 75°]	[−80°/sec, 80°/sec]	N/A
Corresponding output voltage range	[0.5V, 4.5V]	[0.5V, 4.5V]	N/A
Resolution	0.05°	0.05°/sec	2048 cpr*

* cpr: counts per revolution.

the programs coded in C language for the Nios II embedded processor IP. Consequently, the TWIP hardware architecture can be completed, and the resource usage of the proposed FPGA design consists of 3730 LEs (35% of the total number of LEs) and a 34 688-b memory (4% of the total number of RAM bits).

III. MODEL OF THE TWIP

The TWIP (shown in Fig. 1) has two driving rim motors attached on the right and left sides of its chassis. Since there is no other supporting wheel for standing, the TWIP cannot stand without the balanced standing controller. The premier task in this paper is to compute a proper driving torque for both side rim motors such that the TWIP can stand in balance. Furthermore, the traveling and position control and the yaw steering control are also designed for the TWIP. Fig. 3 shows the coordinate system of the TWIP with geometric parameters, where ϕ is the pendulum inclination angle, ψ is the yaw steering angle of the cart, and θ is the mean value of the rim motors' rotary angles, i.e., $\theta = (\theta_R + \theta_L)/2$. Here, θ_R and θ_L are the rotary angles of the right and left rim motors, respectively. Furthermore, l denotes the length between the wheel axle and the gravitational center of the pendulum, r denotes the wheel radius, D is the distance between the right and left wheels, and g is the gravitational acceleration. The masses of the pendulum and the cart are represented by M_P and M_C, respectively. According to physical concepts, the resultant of the forces on the x-axis can be obtained as follows [5]:

$$(M_P + M_C)r\ddot{\theta} + M_P l\ddot{\phi}\cos\phi = f_x \tag{1}$$

where f_x is the exerted force by the rim motors along the x-axis. Furthermore, $f_x = f_R + f_L$, where f_R and f_L denote the forces from the right and left rim motors, respectively. Moreover, the moments around the rotating point of the pendulum yield the following equation

$$M_P lr\ddot{\theta}\cos\phi + M_P l^2\ddot{\phi} - M_P gl\sin\phi = 0. \tag{2}$$

The pendulum angle ϕ and its angular velocity $\dot{\phi}$ can be detected by the tilt sensor and the gyroscope, respectively. Additionally, the angular velocity $\dot{\theta}$ of the rim motors is measured by the encoders. Let the state vector be defined as $\boldsymbol{x}(t) = [\,\phi(t) \quad \dot{\phi}(t) \quad \dot{\theta}(t)\,]^T$. For the sake of T–S fuzzy modeling, the inclination angle of the pendulum is limited within the region as $\phi(t) \in [-\pi/6, \pi/6]$. Let the control input be the driving torque of the rim motors, i.e.,

$$u_f(t) = f_x(t) \times r. \tag{3}$$

Therefore, based on (1)–(3), the T–S fuzzy model of the TWIP system can be presented as a set of if-then rules as follows:

$$\begin{aligned} &Model\ rule\ i: \\ &\text{If } \phi(t) \text{ is } \mu_i, \text{ then } \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_i\boldsymbol{x}(t) + \boldsymbol{B}_i u_f(t), \qquad (i = 1, 2) \end{aligned} \tag{4}$$

where the system matrices are

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 & 0 \\ \frac{(M_P+M_C)g}{M_C l} & 0 & 0 \\ \frac{-M_P g}{M_C r} & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{B}_1 = \begin{bmatrix} 0 \\ \frac{-1}{M_C lr} \\ \frac{1}{M_C r^2} \end{bmatrix}$$

$$\boldsymbol{A}_2 = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\alpha(M_P+M_C)g}{((1-\beta^2)M_P+M_C)l} & 0 & 0 \\ \frac{-\alpha\beta M_P g}{((1-\beta^2)M_P+M_C)r} & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{B}_2 = \begin{bmatrix} 0 \\ \frac{-\beta}{((1-\beta^2)M_P+M_C)lr} \\ \frac{1}{((1-\beta^2)M_P+M_C)r^2} \end{bmatrix}$$

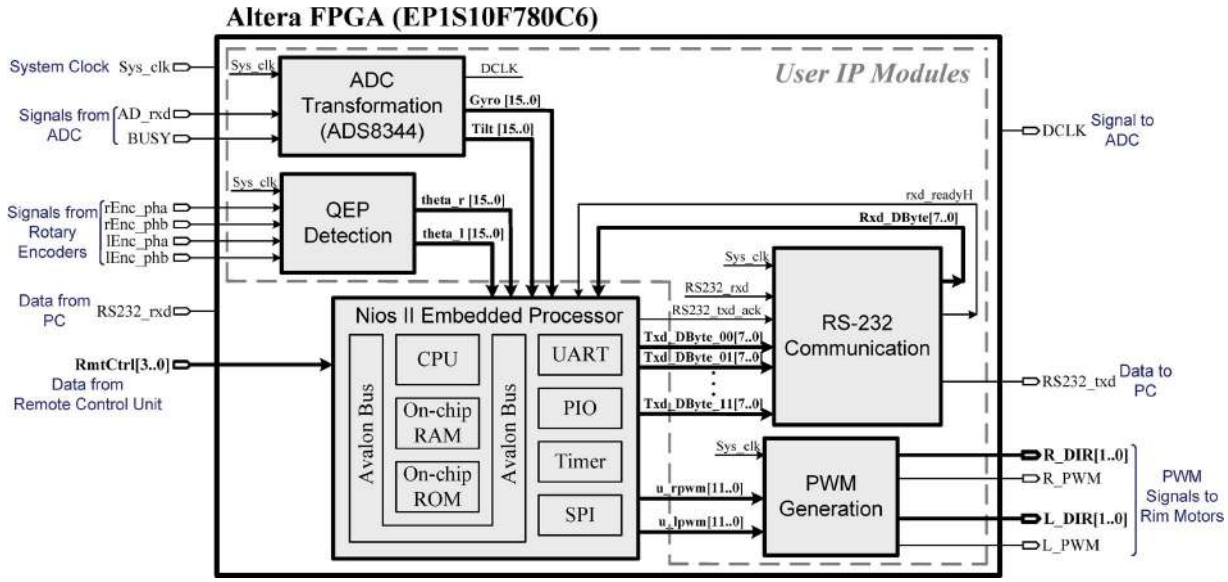

Fig. 4. Internal architecture of the FPGA for TWIP application.

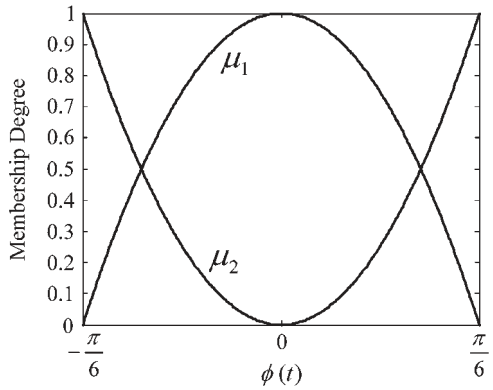

Fig. 5. Membership functions for the fuzzy modeling.

where $\alpha = \sin(\pi/6)/(\pi/6)$ and $\beta = \cos(\pi/6)$. The membership functions of ϕ (shown in Fig. 5) are presented by

$$\mu_1(\phi(t)) = \begin{cases} 1, & \phi(t) = 0 \\ \frac{\sin(\phi(t)) - \alpha\phi(t)}{\phi(t)(1-\alpha)}, & \text{otherwise} \end{cases} \quad (5)$$

$$\mu_2(\phi(t)) = \begin{cases} 0, & \phi(t) = 0 \\ \frac{\phi(t) - \sin(\phi(t))}{\phi(t)(1-\alpha)}, & \text{otherwise.} \end{cases} \quad (6)$$

Therefore, the defuzzification of the aforementioned fuzzy model (4) can be derived as

$$\dot{\boldsymbol{x}}(t) = \sum_{i=1}^{2} \mu_i(\phi(t)) \left[\boldsymbol{A}_i \mathbf{x}(t) + \boldsymbol{B}_i u_f(t)\right]. \quad (7)$$

Consequently, the system (7) will be considered as the model to be controlled in the next section.

IV. Fuzzy Control Design and Simulations

A. FBSC Design

Let the desired state of the TWIP be defined as $\boldsymbol{x}_d(t) = [\,\phi_d(t) \quad \dot{\phi}_d(t) \quad \dot{\theta}_d(t)\,]^T$, where $\phi_d(t)$, $\dot{\phi}_d(t)$, and $\dot{\theta}_d(t)$ are the desired values of $\phi(t)$, $\dot{\phi}(t)$, and $\dot{\theta}(t)$, respectively. Suppose that the TWIP is instructed to stand still and upright, i.e., $\phi_d = 0$, $\dot{\phi}_d = 0$, and $\dot{\theta}_d = 0$. Based on human common sense, if the TWIP's pole has a positive (or negative) angle initially, a positive (or negative) driving force to the rim motors is required to return the pendulum to the upright state; otherwise, the pendulum falls down. The first task in this section is to achieve the goal of balanced standing for the TWIP.

According to the T–S fuzzy model, the following fuzzy PDC controller is employed [29]

$$\begin{aligned} &\textit{Control rule } i: \\ &\text{If } \phi(t) \text{ is } \mu_i, \text{then } u_f(t) = \boldsymbol{F}_i \boldsymbol{x}(t), \qquad (i = 1, 2) \end{aligned} \quad (8)$$

where $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ are the local feedback gains. Finally, the resulting overall fuzzy controller is represented as

$$u_f(t) = \sum_{i=1}^{2} \mu_i(\phi(t)) \boldsymbol{F}_i \boldsymbol{x}(t). \quad (9)$$

Now, each local feedback gain $\boldsymbol{F}_i$ in a consequent part must be determined such that the state $\boldsymbol{x}(t)$ approaches to zero asymptotically. By substituting (9) into (7), the close-loop fuzzy model can be obtained as follows:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \sum_{i=1}^{2} \sum_{j=1}^{2} \mu_i(\phi(t)) \mu_j(\phi(t)) \{\boldsymbol{A}_i + \boldsymbol{B}_i \boldsymbol{F}_j\} \boldsymbol{x}(t) \\ &= \sum_{i=1}^{2} \mu_i^2(\phi(t)) \boldsymbol{G}_{ii} x(t) \\ &\quad + 2\mu_1(\phi(t)) \mu_2(\phi(t)) \left\{\frac{\boldsymbol{G}_{12} + \boldsymbol{G}_{21}}{2}\right\} \boldsymbol{x}(t) \end{aligned} \quad (10)$$

where $\boldsymbol{G}_{ij} = \boldsymbol{A}_i + \boldsymbol{B}_i \boldsymbol{F}_j$.

Lemma 1 [11, Theorem 7]: The equilibrium ($\phi = 0$, $\dot{\phi} = 0$, and $\dot{\theta} = 0$) of the fuzzy control system of (10) is quadratically

stable in the large if there exist symmetric matrices $\boldsymbol{P}$ and $\boldsymbol{W}_{ij}$ such that (11a)–(11d) are satisfied

$$\boldsymbol{P} > \boldsymbol{0} \tag{11a}$$

$$\boldsymbol{G}_{ii}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{G}_{ii} + \boldsymbol{W}_{ii} < \boldsymbol{0}, \qquad (i = 1, 2) \tag{11b}$$

$$\left(\frac{\boldsymbol{G}_{12} + \boldsymbol{G}_{21}}{2}\right)^T \boldsymbol{P} + \boldsymbol{P}\left(\frac{\boldsymbol{G}_{12} + \boldsymbol{G}_{21}}{2}\right) + \boldsymbol{W}_{12} \leq \boldsymbol{0} \tag{11c}$$

$$\tilde{\boldsymbol{W}} \equiv \begin{bmatrix} \boldsymbol{W}_{11} & \boldsymbol{W}_{12} \\ \boldsymbol{W}_{12} & \boldsymbol{W}_{22} \end{bmatrix} > \boldsymbol{0}. \tag{11d}$$

□

Herein, by defining the three matrices $\boldsymbol{X} = \boldsymbol{P}^{-1}$, $\boldsymbol{N}_i = \boldsymbol{F}_i\boldsymbol{X}$, and $\boldsymbol{Y}_{ij} = \boldsymbol{X}\boldsymbol{W}_{ij}\boldsymbol{X}$ and then applying them into the LMI method of [11, Theorem 11], the feedback gains $\boldsymbol{F}_i$, a common $\boldsymbol{P}$, and the symmetric matrices $\boldsymbol{W}_{ij}$ can be obtained.

Lemma 2 [29, Theorems 11 and 13]: For the fuzzy control system of (10), suppose that the initial error vector $\boldsymbol{x}(0)$ is unknown but its upper bound ε is known, i.e., $\|\boldsymbol{x}(0)\| \leq \varepsilon$. Then, the control input (9) can be enforced to satisfy the constraint $\|u_f(t)\| \leq \rho$ if the following conditions are added into the LMIs used in Lemma 1

$$\varepsilon^2 \boldsymbol{I} \leq \boldsymbol{X} \tag{12a}$$

$$\begin{bmatrix} \boldsymbol{X} & \boldsymbol{N}_i^T \\ \boldsymbol{N}_i & \rho^2\boldsymbol{I} \end{bmatrix} \geq \boldsymbol{0}, \qquad (i = 1, 2) \tag{12b}$$

where ε and ρ are predefined positive scalars. □

Therefore, based on Lemmas 1 and 2, the PDC control gains $\boldsymbol{F}_i$ are obtained for guaranteeing $\lim_{t\to\infty}\boldsymbol{x}(t) = \boldsymbol{0}$. In other words, the pendulum will approach to stand still and upright finally.

B. Simulation Results of FBSC

One simulation is first given to demonstrate the performance of the fuzzy balanced standing control (FBSC). Table II lists the geometric parameters and values of the TWIP. Substituting the parameters into the fuzzy model (4) yields the following system matrices

TABLE II
GEOMETRIC PARAMETERS AND VALUES OF THE TWIP

Item	Symbol	Value	Unit
Mass of the pendulum	M_P	9.1	[kg]
Mass of the cart	M_C	25.2	[kg]
Length between the wheel axle and the gravitational center of the pendulum	l	0.5	[m]
Radius of the wheel	r	0.1	[m]
Distance between the right and left wheels	D	0.44	[m]
Gravitational acceleration	g	9.8	[m/s^2]

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 26.6778 & 0 & 0 \\ -35.3889 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{B}_1 = \begin{bmatrix} 0 \\ -0.7937 \\ 3.9683 \end{bmatrix} \tag{13a}$$

$$\boldsymbol{A}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 23.3660 & 0 & 0 \\ -26.8430 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{B}_2 = \begin{bmatrix} 0 \\ -0.6304 \\ 3.6397 \end{bmatrix}. \tag{13b}$$

Because the TWIP is modeled in the range $\phi(t) \in [-\pi/6, \pi/6]$, the initial state is assumed here to be $\boldsymbol{x}(0) = [\,\phi(t) \quad 0 \quad 0\,]^T$, where $|\phi(0)| \leq \pi/6$. Referring to Lemma 2, the initial state is limited to $\|\boldsymbol{x}(0)\| \leq \pi/6$, i.e., $\varepsilon = \pi/6$. Additionally, the constraint of the control signal is selected as $\|u_f(t)\| \leq 40$, i.e., $\rho = 40$. With the settings of $\varepsilon = \pi/6$ and $\rho = 40$, the MATLAB LMI toolbox can be utilized to solve such LMI conditions of Lemmas 1 and 2. The two local controllers (8) of the PDC are constructed, and their local feedback gains $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ are found, respectively, as follows:

$$\boldsymbol{F}_1 = [46.9781 \quad 9.2220 \quad 0.0132] \tag{14a}$$

$$\boldsymbol{F}_2 = [49.7470 \quad 10.2042 \quad 0.0162]. \tag{14b}$$

Then, by using (9), $u_f(t)$ is obtained. The positive definite $\boldsymbol{P}$ and the symmetric matrices $\boldsymbol{W}_{ij}$ are obtained in (15) and (16) as shown at the bottom of the page. Therefore, the stability is ensured, and the control constraints are also satisfied simultaneously.

$$\boldsymbol{P} = \begin{bmatrix} 2.2418 & 0.4524 & 0.0022 \\ 0.4524 & 0.0915 & 0.0004 \\ 0.0022 & 0.0004 & 0.0000 \end{bmatrix} > \boldsymbol{0} \tag{15}$$

$$\tilde{\boldsymbol{W}} \equiv \begin{bmatrix} \boldsymbol{W}_{11} & \boldsymbol{W}_{12} \\ \boldsymbol{W}_{12} & \boldsymbol{W}_{22} \end{bmatrix} = \begin{bmatrix} 4.4694 & 0.9463 & 0.0041 & -0.3628 & -0.0724 & -0.0003 \\ 0.9463 & 0.2010 & 0.0009 & -0.0724 & -0.0145 & -0.0001 \\ 0.0041 & 0.0009 & 0.0000 & -0.0003 & -0.0001 & -0.0000 \\ -0.3628 & -0.0724 & -0.0003 & 3.2763 & 0.6243 & 0.0034 \\ -0.0724 & -0.0145 & -0.0001 & 0.6243 & 0.1196 & 0.0006 \\ -0.0003 & -0.0001 & -0.0000 & 0.0034 & 0.0006 & 0.0000 \end{bmatrix} > \boldsymbol{0} \tag{16}$$

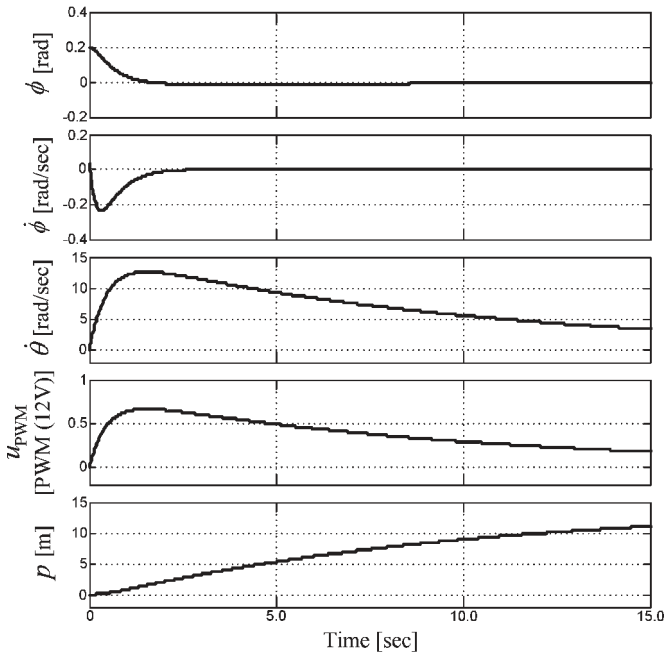

Fig. 6. Simulation results of the FBSC for the TWIP system.

Fig. 6 shows five curves which are the simulation results with the initial state $\boldsymbol{x}(0) = [\,0.2 \quad 0 \quad 0\,]^T$ and which are controlled by the designed FBSC. The first is the pendulum angle $\phi(t)$, the second is the pendulum angular velocity $\dot{\phi}(t)$, the third is the wheel angular velocity $\dot{\theta}(t)$, the fourth is the motor speed control signal $u_{\mathrm{PWM}}(t)$, and the fifth is the position of the TWIP $p(t)$. Since the speed control signal PWM is adopted to manipulate the rim motors of the TWIP system, this paper converts the control torque $u_f(t)$ into the speed control signal $u_{\mathrm{PWM}}(t)$. Thus, based on (1) and (3), the speed control $u_{\mathrm{PWM}}(t)$ can be obtained as follows:

$$u_{\mathrm{PWM}}(t) = \int_0^t \frac{u_f(t) - M_P l r \ddot{\phi}(t) \cos(\phi(t))}{(M_P + M_C) r^2}\, dt \tag{17}$$

where $u_{\mathrm{PWM}}(t)$ applies to the right and left rim motors simultaneously. Hence, $u_{\mathrm{R,PWM}}(t) = u_{\mathrm{L,PWM}}(t) = u_{\mathrm{PWM}}(t)$ will make the TWIP move forward or backward straightly, where $u_{\mathrm{R,PWM}}(t)$ and $u_{\mathrm{L,PWM}}(t)$ represent the speed control signals to the right and left rim motors, respectively. Meanwhile, the TWIP position $p(t)$ can be calculated by the encoders, i.e., $p(t) = \theta(t) r$. Although the TWIP can maintain the inverted pendulum in the upright state by the designed FBSC, it is found that the control response of $\dot{\theta}(t)$ seems tardy and needs over 15 s to slow down (see Fig. 6). It means that the position control problem is not solved by the FBSC; therefore, the TWIP travels over 10 m far away from the origin in order to keep balanced standing.

C. FTPC Design

In this section, the fuzzy traveling and position control (FTPC) is introduced. The FTPC is originally proposed to help the FBSC such that the TWIP is not only balanced standing but also meets the position request. Based on human common sense, when the standing TWIP tries to move forward (or backward) from an almost zero initial condition ($\phi \approx 0$, $\dot{\phi} \approx 0$, and $\dot{\theta} \approx 0$), the starting action gives the pendulum a positive (or negative) angle. Thus, the TWIP must move forward (or backward) to avoid the pendulum falling down. Now, the objective of this section is to make the TWIP travel to and keep at the desired position, but the pendulum cannot fall down. Therefore, designing an adequate driving torque for the rim motors to achieve the aforementioned objective is the main task.

Let the position error of the TWIP be denoted as

$$e_p(t) = p(t) - p_d \tag{18}$$

where p_d represents the desired position of the TWIP. Furthermore, let the desired state of the TWIP be $\boldsymbol{x}_d(t) = [\,\phi_d(t) \quad 0 \quad 0\,]^T$. Then, an alternative state $\boldsymbol{z}(t)$ is defined as the state error

$$\begin{aligned} \boldsymbol{z}(t) &= \boldsymbol{x}(t) - \boldsymbol{x}_d(t) \\ &= \left[\phi(t) - \phi_d(t) \quad \dot{\phi}(t) - \dot{\phi}_d(t) \quad \dot{\theta}(t) - \dot{\theta}_d(t)\right]^T \\ &= \left[e_\phi(t) \quad \dot{\phi}(t) \quad \dot{\theta}(t)\right]^T \end{aligned} \tag{19}$$

where $e_\phi(t)$ represents the pendulum angle error. For the fuzzy model (4), the fuzzy PDC controller is employed

$$\begin{aligned} &\textit{Control rule } i: \\ &\text{If } \phi(t) \text{ is } \mu_i, \text{ then } u_f(t) = \boldsymbol{F}_i \boldsymbol{z}(t), \qquad (i = 1, 2). \end{aligned} \tag{20}$$

Therefore, by combining (7), (19), and (20), the alternative dynamic equation is

$$\begin{aligned} \dot{\boldsymbol{z}}(t) = \sum_{i=1}^{2}\sum_{j=1}^{2} \mu_i(\phi(t))\, \mu_j(\phi(t)) \{\boldsymbol{A}_i + \boldsymbol{B}_i \boldsymbol{F}_j\} \boldsymbol{z}(t) \\ + \sum_{i=1}^{2} \mu_i(\phi(t))\, \boldsymbol{A}_i \boldsymbol{x}_d(t). \end{aligned} \tag{21}$$

Substitute $\boldsymbol{A}_i$ and $\boldsymbol{x}_d(t)$ into (21) and rewrite it as

$$\begin{aligned} \dot{\boldsymbol{z}}(t) = \underbrace{\sum_{i=1}^{2}\sum_{j=1}^{2} \mu_i(\phi(t))\, \mu_j(\phi(t)) \{\boldsymbol{A}_i + \boldsymbol{B}_i \boldsymbol{F}_j\} \boldsymbol{z}(t)}_{T_1} \\ + \underbrace{\sum_{i=1}^{2} \mu_i(\phi(t)) \begin{bmatrix} 0 \\ a_{i,21} \\ a_{i,31} \end{bmatrix} \phi_d(t)}_{T_2} \end{aligned} \tag{22}$$

where $a_{i,21} > 0$ and $a_{i,31} < 0$ are the entries at positions (2, 1) and (3, 1) of the matrix $\boldsymbol{A}_i$, respectively. Let the first term on the right side of (20) be denoted by T_1 and the second term be denoted by T_2. Supposing that the desired pendulum angle is $\phi_d = 0$, the term T_2 can disappear to make the system (22) have the same form as (10). However, our main task is to achieve $e_p(t) \to 0$ and $\|\boldsymbol{z}(t)\| < \delta$ (the TWIP travels to and stands still

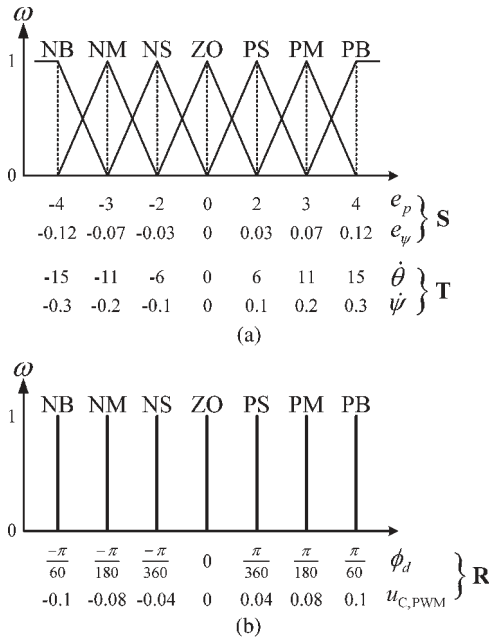

Fig. 7. (a) Membership functions of two antecedents. (b) Membership functions of the consequent. For FTPC with $\mathbf{S} = e_p$, $\mathbf{T} = \dot{\theta}$, and $\mathbf{R} = \phi_d$ and for FYSC with $\mathbf{S} = e_\psi$, $\mathbf{T} = \dot{\psi}$, and $\mathbf{R} = u_{\mathrm{C,PWM}}$.

at the desired position), where δ is a small constant. If $\phi_d = 0$ always, T_2 disappears and only the FBSC works so that the TWIP stands still at the origin always. Consequently, the TWIP will not travel to the desired position p_d. Therefore, we need another controller FTPC working with the FBSC together to push the TWIP traveling to and standing still at the desired position.

The FTPC is a Mamdani if-then fuzzy rule base. Let the position error e_p and the wheel speed $\dot{\theta}$ be the antecedents of each rule of the FTPC and the desired pendulum angle ϕ_d be the consequent. Fig. 7 shows the membership functions of the antecedents (e_p and $\dot{\theta}$) and consequent (ϕ_d). They all are decomposed into seven fuzzy sets: negative big (NB), negative medium (NM), negative small (NS), zero (ZO), positive small (PS), positive medium (PM), and positive big (PB). Therefore, there are 49 fuzzy rules totally in the FTPC, which are tabulated in Table III [3], [14]. The minimum inference engine and a weighted average method derive the resulting crisp output given as follows [14]:

$$\phi_d = \frac{\sum_{k=1}^{49} \min\left(\omega^k(e_p), \omega^k(\dot{\theta})\right) \cdot \gamma_{\mathrm{FTPC}}^k}{\sum_{k=1}^{49} \min\left(\omega^k(e_p), \omega^k(\dot{\theta})\right)} \tag{23}$$

where $\omega^k(e_p)$ and $\omega^k(\dot{\theta})$ are the kth-rule fired membership degrees and γ_{FTPC}^k represents the singleton value of the consequent (ϕ_d) of the kth rule. Based on the FTPC (23), the term T_2 in (22) plays a key role in controlling the TWIP to travel to and stand still at the desired position. The particular idea will be explained in detail in the following paragraphs.

When the initial state of the TWIP is standing still at the origin, i.e., $\boldsymbol{x}(0) = [\,0 \quad 0 \quad 0\,]^T$ and $p(0) = 0$, the desired position command is $p_d = c > 0$, where c is a constant. According

TABLE III
FUZZY RULE TABLE FOR FTPC WITH $\mathbf{S} = e_p$, $\mathbf{T} = \dot{\theta}$, AND $\mathbf{R} = \phi_d$ AND FOR FYSC WITH $\mathbf{S} = e_\psi$, $\mathbf{T} = \dot{\psi}$, AND $\mathbf{R} = u_{\mathrm{C,\,PWM}}$

Consequent: R	Antecedent: T						
Antecedent: S	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PB	PB	PM	PS	ZO
NM	PB	PB	PB	PM	PS	ZO	NS
NS	PB	PB	PM	PS	ZO	NS	NM
ZO	PB	PM	PS	ZO	NS	NM	NB
PS	PM	PS	ZO	NS	NM	NB	NB
PM	PS	ZO	NS	NM	NB	NB	NB
PB	ZO	NS	NM	NB	NB	NB	NB

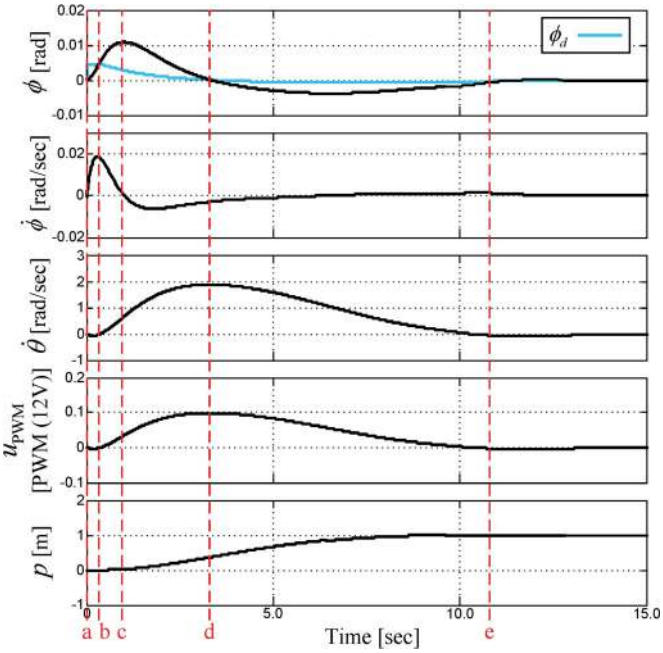

Fig. 8. Simulation results of FBSC and FTPC combined together for the traveling and position control of the TWIP system.

to (22) and the FTPC (23), the following cases are discussed. The simulation result (see Fig. 8) is helpful in explaining the following cases.

Case 1: When $p \ll p_d$, $\dot{\theta} \approx 0$, and $\phi < \phi_d$.

Here, $e_p = p - p_d \ll 0$ and $\dot{\theta} \approx 0$, and according to the FTPC (23) and Table III, ϕ_d is positive and large. Notably, the term T_1 causes $\boldsymbol{z}(t)$ to approach zero (i.e., $\phi \to \phi_d$, $\dot{\phi} \to 0$, and $\dot{\theta} \to 0$). The second element of T_2 is very positive to increase ϕ largely, and the third element is very negative for $\dot{\theta}$. Therefore, by combining the works of T_1 and T_2, the TWIP moves backward suddenly (see the responses $\dot{\theta}$ and p inside the interval [a, b] in the third and fifth figures of Fig. 8, respectively); however, the pendulum angle ϕ will slant forward very rapidly and approach ϕ_d (see the response ϕ inside the interval [a, b] in the first figure of Fig. 8). Since the FBSC works continuously, in order to keep the pendulum from not falling down, the TWIP will

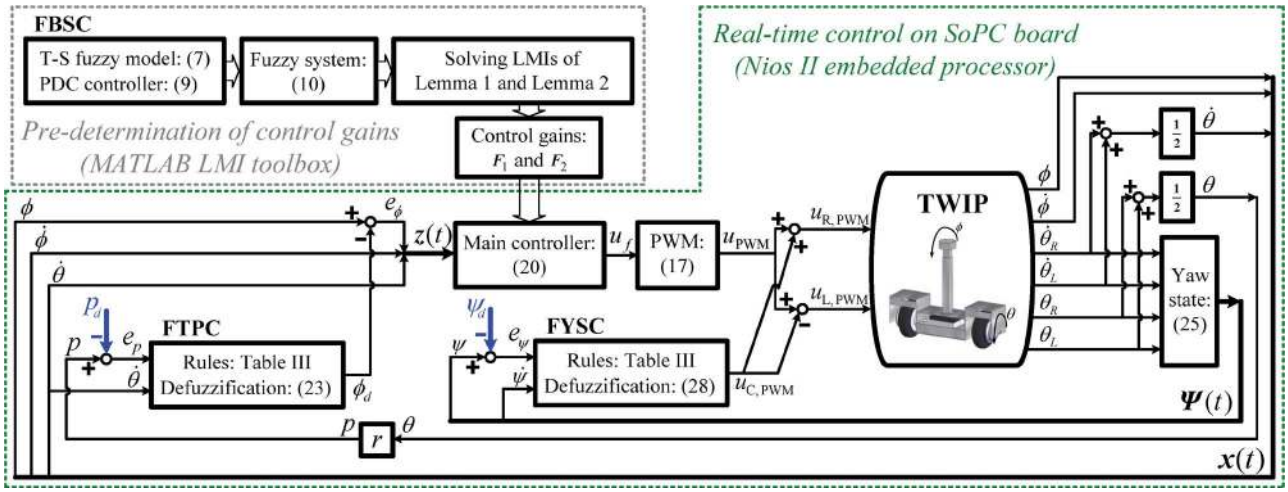

Fig. 9. Overall control diagram of the TWIP.

then move forward from backward (from $\dot{\theta} < 0$ to $\dot{\theta} > 0$) on the moment. Therefore, Case 2 occurs.

Case 2: When $p \ll p_d$, $\dot{\theta} > 0$, and $\phi \approx \phi_d$.

Here, $e_p = p - p_d \ll 0$ and $\dot{\theta} > 0$, and according to the FTPC (23) and Table III, ϕ_d is positive but not large. Notably, the term T_1 causes $\boldsymbol{z}(t)$ to always approach zero (i.e., $\phi \to \phi_d$, $\dot{\phi} \to 0$, and $\dot{\theta} \to 0$). The second element of T_2 is positive to increase ϕ slightly (an effort opposite to the work of T_1), and the third element is negative for $\dot{\theta}$ (an effort opposite to the work of T_1). Therefore, by combining the works of T_1 and T_2, the TWIP moves forward with increasing speed to approach the desired position (see the responses $\dot{\theta}$ and p inside the interval [b, c] in the third and fifth figures of Fig. 8, respectively), and the pendulum angle ϕ increases to exceed ϕ_d, i.e., $\phi > \phi_d$ (see the response ϕ inside the interval [b, c] in the first figure of Fig. 8). Since the FBSC works continuously, in order to keep the pendulum from not falling down, the pendulum angle ϕ will not be increasing always but will be limited at a certain moment. Thus, Case 3 occurs.

Case 3: When $p < p_d$, $\dot{\theta} > 0$, and $\phi > \phi_d$ (ϕ is at the limit point, but $\dot{\phi} = 0$).

Here, $e_p = p - p_d < 0$ and $\dot{\theta} > 0$, and according to the FTPC (23) and Table III, ϕ_d is slightly positive. Similar to Case 2, by combining the works of T_1 and T_2, the TWIP still moves forward with increasing speed to approach the desired position (see the responses $\dot{\theta}$ and p inside the interval [c, d] in the third and fifth figures of Fig. 8, respectively), but the pendulum angle ϕ decreases to approach ϕ_d simultaneously (see the response ϕ inside the interval [c, d] in the first figure of Fig. 8). Since the FBSC works continuously, the pendulum angle ϕ indeed approaches ϕ_d, but the TWIP already accelerates to a very fast speed. However, the TWIP needs to keep approaching to the desired position and to slow down its speed to prepare for stopping at the desired position. Consequently, Case 4 occurs.

Case 4: When p is smaller than but almost equal to p_d, $\dot{\theta} \gg 0$ (but $\ddot{\theta} = 0$), and $\phi \approx \phi_d$.

Here, $e_p = p - p_d < 0$ and $\dot{\theta} \gg 0$, and according to the FTPC (23) and Table III, ϕ_d is very small and negative. Notably, the term T_1 causes $\boldsymbol{z}(t)$ to always approach zero (i.e., $\phi \to \phi_d$, $\dot{\phi} \to 0$, and $\dot{\theta} \to 0$). The second element of T_2 is negative to decrease ϕ slightly (an effort opposite to the work of T_1), and the third element is slightly positive for $\dot{\theta}$ (an effort opposite to the work of T_1). By combining the work of T_1 and T_2, the TWIP slows down and approaches the desired position finally, i.e., $\dot{\theta} \to 0$ and $p \to p_d$ (see the responses $\dot{\theta}$ and p inside the interval [d, e] in the third and fifth figures of Fig. 8, respectively), and the pendulum angle ϕ also approaches ϕ_d finally (see the response ϕ inside the interval [d, e] in the first figure of Fig. 8).

After finishing the aforementioned analysis, this paper must explain the fuzzy rules in detail in terms of the relationship between the antecedents (e_p and $\dot{\theta}$) and the consequent (ϕ_d). For the purpose of traveling and position control, the TWIP should move with a positive (or negative) speed if the position error $e_p < 0$ (or $e_p > 0$); otherwise, the TWIP moves away from the desired position. Based on common sense, the TWIP cannot keep a constant speed $\dot{\theta}$ and a constant angle $\phi_d \neq 0$ simultaneously due to the existence of gravity. If a constant speed $\dot{\theta}$ is kept, ϕ_d should be zero. If a constant angle $\phi_d = \text{constant} > 0$ (or $\phi_d < 0$) is maintained, the speed should increase (or decrease) continuously; otherwise, the pendulum falls down. More specifically, a positive (or negative) angle ϕ_d can cause a positive (or negative) acceleration to increase the moving speed of the TWIP. Therefore, the fuzzy rules for the antecedents (e_p and $\dot{\theta}$) and consequent (ϕ_d) of the proposed FTPC (see Table III) can be obtained. During the traveling and position motion, the desired angle ϕ_d varies the moving speed $\dot{\theta}$ to derive the state $\boldsymbol{x}(t)$ and the position error $e_p(t)$ to reach the stable equilibrium state ($\boldsymbol{x}(t) = [\,0 \quad 0 \quad 0\,]^T$ and $e_p(t) = 0$).

Fig. 9 depicts the proposed control scheme. By solving the LMIs of Lemmas 1 and 2, one can find the control feedback gains $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ to accomplish the balanced standing control

first. Then, the desired angle $\phi_d(t)$ is tuned by the FTPC such that the purpose of traveling and position control is achieved.

D. Simulation Results of FTPC

This paper provides three simulations to evaluate the performance of the proposed FTPC. The control gains $\boldsymbol{F}_1$ (14a) and $\boldsymbol{F}_2$ (14b) are again used, the fuzzy rule table (see Table III) for the antecedents (e_p and $\dot{\theta}$) and consequent (ϕ_d) is added, and they are used together. Consequently, the FTPC (23) can tune the desired angle ϕ_d for the traveling and position control of the TWIP.

Fig. 8 shows the simulation results which are the responses of the desired position control of the TWIP system. In the simulation, standing still at the origin is the initial state of the TWIP, i.e., $\boldsymbol{x}(0) = [\,0 \quad 0 \quad 0\,]^T$ and $p(0) = 0$. The desired position command is $p_d = 1$ m. In the fifth figure of Fig. 8, the TWIP travels forward and approaches the target position within 10 s. The first, third, and fifth figures of Fig. 8 indicate that the combination of the FTPC and FBSC works very well. The detailed analysis of Fig. 8 has been discussed in Section IV-C.

Fig. 10 shows the second simulation results to demonstrate that the proposed FTPC can enhance the performance of the FBSC. The TWIP is again initialed at $\boldsymbol{x}(0) = [\,0.2 \quad 0 \quad 0\,]^T$ and $p(0) = 0$. The behavior in concern is that the TWIP can not only keep balanced standing but also hold the position return to the origin, i.e., the desired conditions of the TWIP should be $\boldsymbol{x}_d = [\,0 \quad 0 \quad 0\,]^T$ and $p_d = 0$. In Fig. 10, the TWIP slows down rapidly and accurately approaches the origin in 10 s. Comparing the responses of Fig. 10 (the combination of the FBSC and FTPC for balanced standing and position control) with the responses of Fig. 6 (with only the FBSC for the balanced standing control), it is found that both the FBSC and FTPC working together can keep the TWIP standing stably and returning to and fixing at the origin position simultaneously.

Fig. 11 is the simulation result for the case where an additional control disturbance occurs with $\boldsymbol{x}_d = [\,0 \quad 0 \quad 0\,]^T$ and $p_d = 0$. The disturbance Δu_f, as shown in (24), is added into the control input u_f to disturb the balance of the TWIP

$$\Delta u_f(t) = \begin{cases} 2[\text{Nt-m}], & \text{when } 5.0 \le t \le 5.5 \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

In Fig. 11, the positive disturbance Δu_f causes the TWIP to move forward and to slant negatively. For the purpose of the FBSC, the TWIP moves backward quickly to avoid falling down. Then, the TWIP travels forward to reach the desired position due to the effort of the FTPC.

E. FYSC Design

In addition to the balanced standing control and the traveling and position control, there is one more control for the TWIP to be discussed, i.e., the yaw steering control. The yaw steering control of the TWIP is to make the TWIP turn to the left or to the right. Let the yaw state of the TWIP be defined as $\boldsymbol{\Psi}(t) = [\,\psi(t) \quad \dot{\psi}(t)\,]^T$ in which $\psi(t)$ and $\dot{\psi}(t)$ are the yaw angle and the yaw angular velocity, respectively. Referring

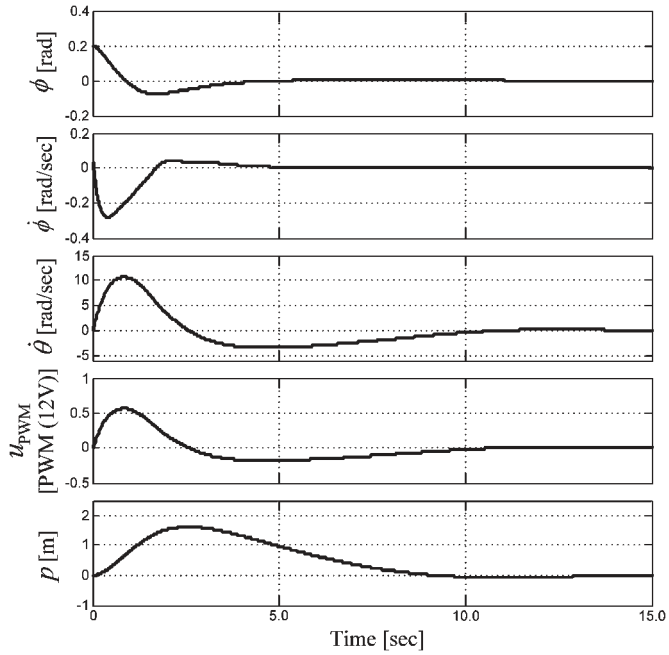

Fig. 10. Simulation results of FBSC and FTPC combined together for the balanced standing control of the TWIP system.

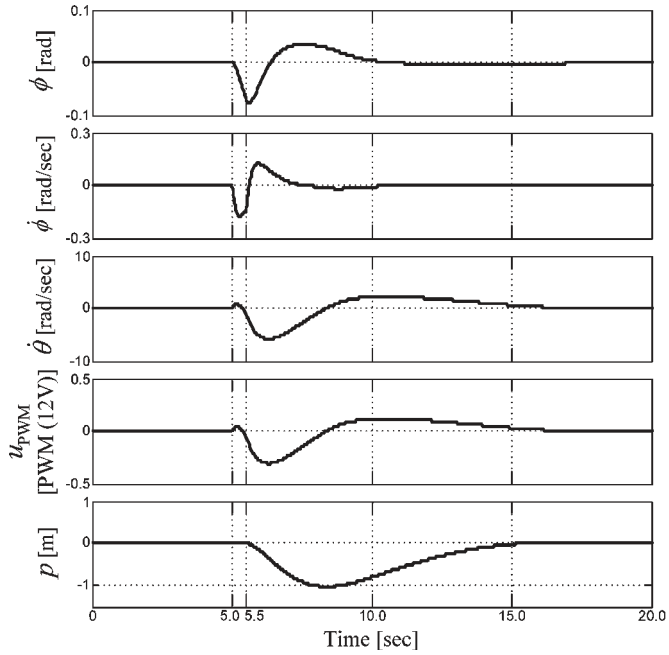

Fig. 11. Simulation results of FBSC and FTPC combined together against the control disturbance of the TWIP system.

to the TWIP model of Fig. 3(c), the kinematics of the yaw steering state can be represented as

$$\boldsymbol{\Psi}(t) = \begin{bmatrix} \psi(t) \\ \dot{\psi}(t) \end{bmatrix} = \begin{bmatrix} \frac{r}{D}\left(\theta_R(t) - \theta_L(t)\right) \\ \frac{r}{D}\left(\dot{\theta}_R(t) - \dot{\theta}_L(t)\right) \end{bmatrix}. \tag{25}$$

Furthermore, the desired steering state is defined as $\boldsymbol{\Psi}_d(t) = [\,\psi_d(t) \quad \dot{\psi}_d(t)\,]^T$, where $\psi_d(t)$ and $\dot{\psi}_d(t)$ are the desired values of $\psi(t)$ and $\dot{\psi}(t)$, respectively. Notably, for the yaw steering

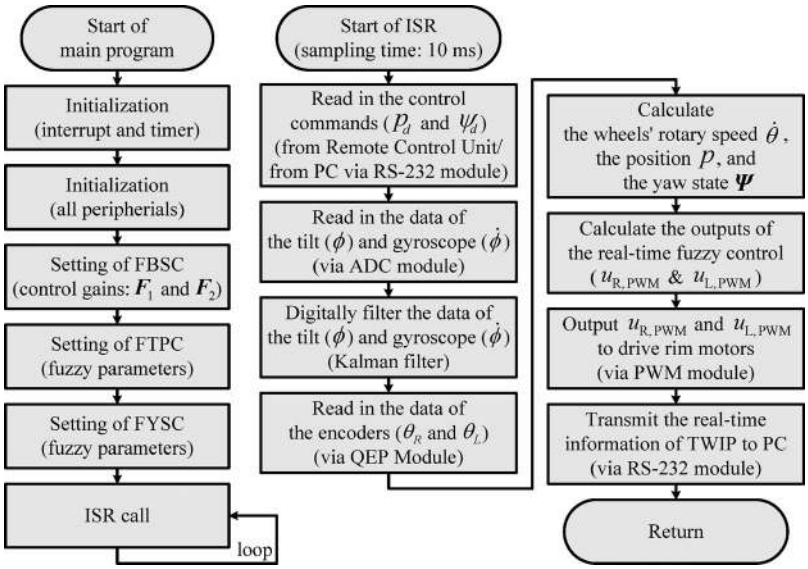

Fig. 12. Flowchart of the main program and the ISR executed by Nios II embedded processor.

action of the TWIP, $\dot{\psi}_d$ should always be zero. Consequently, the yaw error vector is obtained as

$$\begin{aligned} \boldsymbol{e}_{\boldsymbol{\Psi}}(t) &= \boldsymbol{\Psi}(t) - \boldsymbol{\Psi}_d(t) \\ &= \begin{bmatrix} \psi(t) - \psi_d(t) & \dot{\psi}(t) - \dot{\psi}_d(t) \end{bmatrix}^T \\ &= \begin{bmatrix} e_\psi(t) & \dot{\psi}(t) \end{bmatrix}^T \end{aligned} \tag{26}$$

where $e_\psi(t)$ represents the yaw angle error. The proposed fuzzy yaw steering control (FYSC) is also a Mamdani if-then fuzzy rule base, which takes the yaw angle error e_ψ and the yaw angular velocity $\dot{\psi}$ as the antecedents. In the meantime, the consequent of the FYSC is a compensated speed control signal $u_{\mathrm{C,PWM}}$, which is positive or negative, to be added to the right or left rim motors as indicated in (27) (shown in Fig. 9). Hence, the speed control signals applied to the right and left rim motors are obtained as follows:

$$\begin{cases} u_{\mathrm{R,PWM}}(t) = u_{\mathrm{PWM}}(t) + u_{\mathrm{C,PWM}}(t) \\ u_{\mathrm{L,PWM}}(t) = u_{\mathrm{PWM}}(t) - u_{\mathrm{C,PWM}}(t). \end{cases} \tag{27}$$

Thus, the $u_{\mathrm{C,PWM}}$ causes an unequal speed of the right and left wheels such that the TWIP can rotate in the yaw plane. Fig. 7 shows the membership functions of the antecedents (e_ψ and $\dot{\psi}$) and consequent ($u_{\mathrm{C,PWM}}$). Table III is applied again in the FYSC, which also uses the minimum inference engine and weighted average method to derive the resulting crisp output

$$u_{\mathrm{C,PWM}} = \frac{\sum_{k=1}^{49} \min\left(\omega^k(e_\psi), \omega^k(\dot{\psi})\right) \cdot \gamma_{\mathrm{FYSC}}^k}{\sum_{k=1}^{49} \min\left(\omega^k(e_\psi), \omega^k(\dot{\psi})\right)} \tag{28}$$

where $\omega^k(e_\psi)$ and $\omega^k(\dot{\psi})$ are the kth-rule fired membership degrees and γ_{FYSC}^k represents the singleton value of the consequent ($u_{\mathrm{C,PWM}}$) of the kth rule. During the yaw steering motion, the compensated speed signal $u_{\mathrm{C,PWM}}$ derives the yaw error $\boldsymbol{e}_{\boldsymbol{\Psi}}(t)$ to reach the stable equilibrium state ($\boldsymbol{e}_{\boldsymbol{\Psi}}(t) = [0 \quad 0]^T$). Because the design idea of the FYSC has been clearly explained, the computer simulation of the FYSC is omitted.

Referring to Fig. 9, the completed control scheme consists of the proposed FBSC, FTPC, and FYSC. The Mamdani architecture-based FTPC and FYSC are presented to enhance the performance of the FBSC. Finally, the TWIP can achieve the control goals of balanced standing, traveling and position, and yaw steering.

V. Embedded Processor Design and Experimental Results

A. Program Design of Nios II Embedded Processor

Recalling the SoPC hardware architecture of the TWIP (see Fig. 4; as introduced in Section II), this section presents the software program design of the Nios II embedded processor IP which is employed to execute the fuzzy control scheme (see Fig. 9) incorporating with the user IP modules in the FPGA. Fig. 12 shows the flowchart of the control program, where the main program and the interrupt service routine (ISR) for the fuzzy control scheme (see Fig. 9) are coded in C language. In the ISR, the user can easily issue motion commands to the TWIP by using the remote control unit or a PC. The real-time inclination data of the tilt sensor (ϕ) and gyroscope ($\dot{\phi}$) are read in via the ADC module. To eliminate the jumping effect and zigzag noise of the signals from the tilt sensor and the gyroscope, two efficient Kalman filters (KFs) [32] are used for ϕ and $\dot{\phi}$, respectively, such that the real-time inclination of the TWIP can be determined precisely. The KF application details are elaborated in Appendix B. With the QEP module, the embedded processor calculates the wheels' rotary speed $\dot{\theta}$, the position p, and the yaw state $\boldsymbol{\Psi}$ of the TWIP. Finally, the outputs of the fuzzy control ($u_{\mathrm{R,PWM}}$ and $u_{\mathrm{L,PWM}}$) are obtained to drive the rim motors via the PWM module. For monitoring and recording the experimental results, the real-time TWIP information is transmitted to the PC in each ISR in which its interrupt interval is designed to be 10 ms.

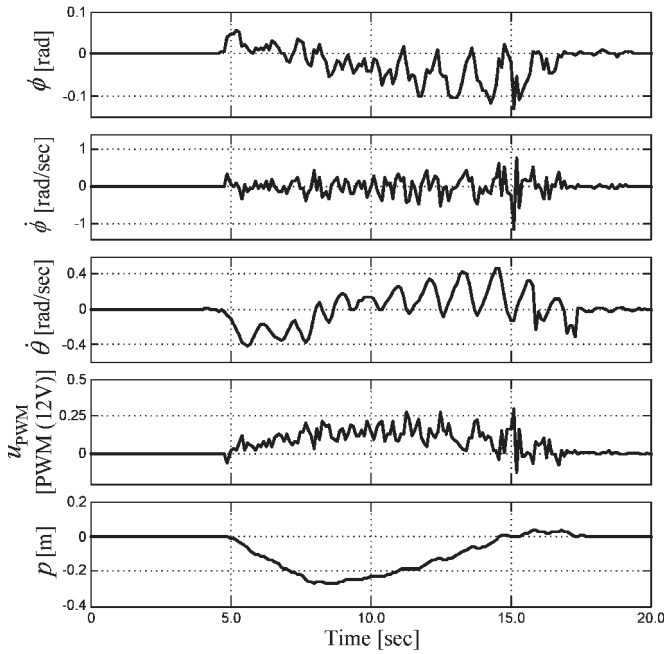

Fig. 13. Experiment results of the FTPC for the real TWIP.

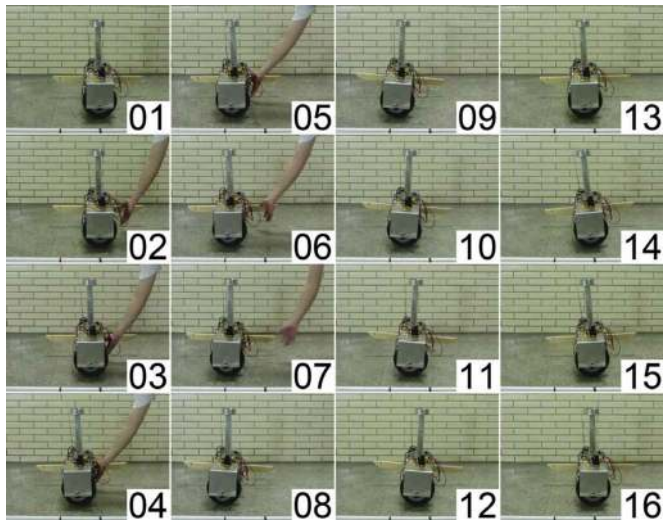

Fig. 14. Sixteen sequential images for the real experiment of the FTPC.

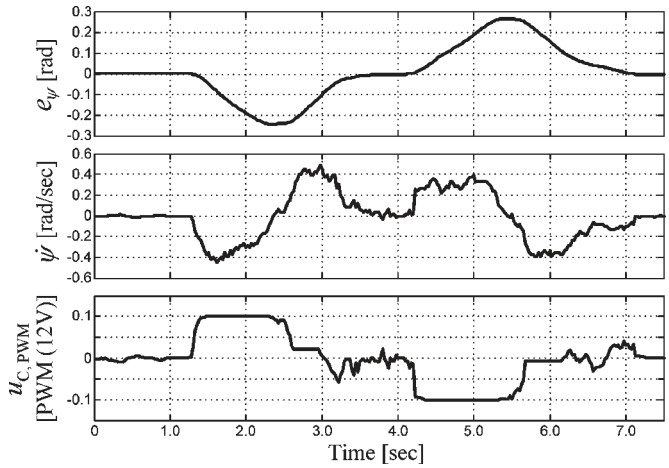

Fig. 15. Experiment results of the FYSC for the real TWIP.

B. Experimental Results

A real TWIP (see Fig. 1) is established practically in the laboratory to illustrate the effectiveness of the proposed control scheme. Using the same parameter setting in the simulations, two real experiments are implemented to verify the feasibility of the proposed FTPC and FYSC, respectively. The experiments are executed in an indoor place.

Fig. 13 presents the first experiment results which test the function of the FTPC. The given initial conditions are $\boldsymbol{x}(0) = [\,0 \quad 0 \quad 0\,]^T$ and $p(0) = 0$; then, the TWIP is shifted by a hand to a distance away from the origin. Let the desired position be at the origin, i.e., $p_d = 0$. The signals $\phi(t)$, $\dot{\phi}(t)$, and $\dot{\theta}(t)$ are detected by the tilt sensor, gyroscope, and encoders, respectively; meanwhile, the position $p(t)$ can be obtained. The speed control signal $u_{\mathrm{PWM}}(t)$ is computed by the Nios II embedded processor. Fig. 14 shows the sequential images of the first experiment. The TWIP is at an upright posture initially (see the response during the time interval [0, 4.8] in Fig. 13 and the still image 01 in Fig. 14). Then, we pushed the TWIP, by a hand, to the left about 30 cm away from the origin (see the response during the time interval [4.8, 8.0] in Fig. 13 and the images 02–07 in Fig. 14). The TWIP is controlled to slant positively (right) and to move forward (right) to approach the origin (see the response during the time interval [8.0, 14.0] in Fig. 13 and the images 08–13 in Fig. 14). Finally, the TWIP is controlled to slant negatively (left) and slow down at the origin stably (see the response during the time interval [14.0, 20.0] in Fig. 13 and the images 14–16 in Fig. 14). Although additional hand shifting disturbs the balance of the TWIP and shifts the TWIP about 30 cm away from the origin, the TWIP can return to the origin accurately and stand still (see Figs. 13 and 14). It is seen that the response in Fig. 11 is smoother than that in Fig. 13 because the former is the simulation result with constant step disturbance and the latter is the practical experiment with human hand shifting.

Fig. 15 shows the second experiment results which demonstrate the performance of the FYSC. The initial steering state is at $\boldsymbol{\Psi}(0) = [\,0 \quad 0\,]^T$; then, a hand turning changes the steering angle twice while the desired steering angle is kept at $\psi_d = 0$. The yaw states (e_ψ and $\dot{\psi}$) are calculated from the encoders, and the compensated speed control signal $(u_{\mathrm{C,PWM}})$ can be obtained. Fig. 16 shows the sequential images of the second experiment. The TWIP is at an upright posture initially (see the response during the time interval [0, 1.25] in Fig. 15 and the still image 01 in Fig. 16). Then, we used hands to turn the TWIP's steering angle clockwise to a -0.24 radian (see the response during the time interval [1.25, 2.5] in Fig. 15 and the images 02–05 in Fig. 16). The TWIP is controlled to rotate counterclockwise to the original steering angle rapidly (see the response during the time interval [2.5, 3.75] in Fig. 15 and the images 06–09 in Fig. 16). Next, we used hands to turn the TWIP counterclockwise to a 0.26 radian (see the response during the time interval [3.75, 5.5] in Fig. 15 and the images 10–14 in Fig. 16). Finally, the TWIP is controlled to rotate clockwise to the original steering angle stably (see the response

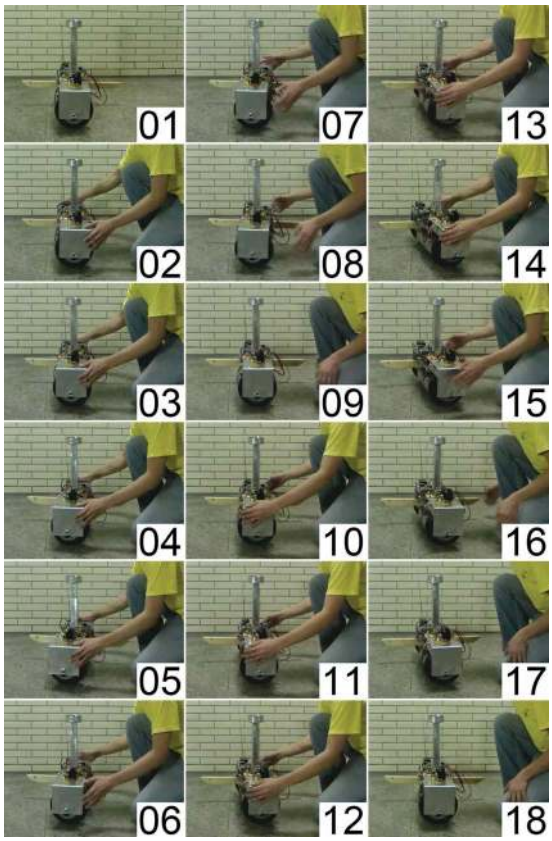

Fig. 16. Eighteen sequential images for the real experiment of the FYSC.

during the time interval [5.5, 7.5] in Fig. 15 and the images 15–18 in Fig. 16). When we used hands to turn the TWIP to a negative (or positive) steering angle, the TWIP can rotate counterclockwise (or clockwise) to the desired steering angle quickly (see Figs. 15 and 16).

The aforementioned two experiments have demonstrated that the TWIP's behaviors under the balanced standing control, traveling and position control, and steering control are implemented successfully in the real TWIP.

VI. Conclusion

This paper has implemented the SoPC developmental hardware/software establishment on a TWIP system. The completed control scheme (containing the FBSC, FTPC, and FYSC) has been proposed to control the TWIP achieving the desired control behaviors. The fuzzy PDC-based FBSC was constructed to attain the balanced standing control. The FTPC or FYSC have been proposed to realize the traveling and position control or the yaw steering control, respectively. Both the simulation and practical experiment results indeed have verified that the proposed control schemes are effective for the real TWIP systems.

Appendix A
User IP Modules

As shown in Fig. 4, the user IP modules (custom logic) operate at the 50-MHz system clock (Sys_clk), and their task is to handle the real-time signal processing of the peripheral devices in the TWIP.

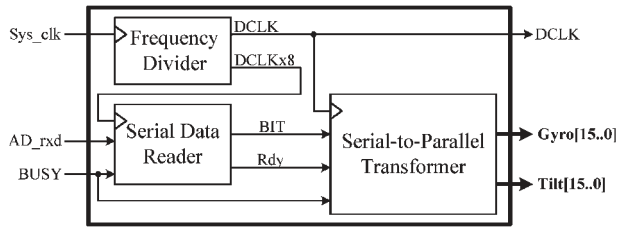

Fig. 17. Block diagram of the ADC transformation circuit module.

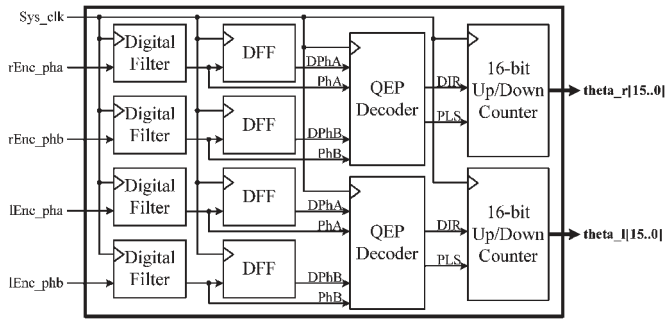

Fig. 18. Block diagram of the QEP detection circuit module.

1) ADC Transformation Circuit Module: An ADC (ADS8344) is adopted as the interface between the analog-signal sensors (including the tilt sensor and the gyroscope) and the digital-signal SoPC board. Since the ADS8344 is a 16-b ADC with a synchronous serial interface, this paper builds the ADC module for the serial signal processing of the ADS8344 (see Figs. 4 and 17). In the ADC module, a frequency divider generates a 100-kHz clock (DCLK) as the sampling rate. The binary-serial signal (AD_rxd) can be read in and then transformed into two 16-b parallel data (**Tilt[15..0]** and **Gyro[15..0]**). Consequently, the circuit module achieves the inclination measurement of the tilt sensor and the gyroscope for the TWIP.

2) QEP Detection Circuit Module: Referring to [9] and [12], the QEP module is designed as a direct interface with the two rotary encoders to obtain their rotary angles and directions (see Figs. 4 and 18). Fig. 18 shows the QEP module which is implemented for detecting the right-encoder phases (rEnc_pha/rEnc_phb) and left-encoder phases (lEnc_pha/lEnc_phb) to obtain the right and left wheel rotations (**theta_r[15..0]** and **theta_l[15..0]**). Once the two rim motors run, the encoder output QEP signals are digitally filtered, decoded, and calculated as two 16-b counting values to represent the real-time feedback wheels' rotations.

3) RS-232 Communication Circuit Module: As shown in Fig. 2, one modem is installed in the TWIP and the other is connected to a PC to make a wireless communication between the TWIP and the PC via the RS-232 protocol. The RS-232 module (see Figs. 4 and 19) is therefore constructed as the binary-serial communication interface, which provides the FPGA a duplex transmission function such that the TWIP can receive a control command byte (RS232_rxd to **Rxd_DByte[7..0]**) from the PC and transmit 12-B real-time data (**Txd_DByte_01[7..0]**–**Txd_DByte_11[7..0]** to RS232_txd) to the PC simultaneously. Here, the 12-B real-time

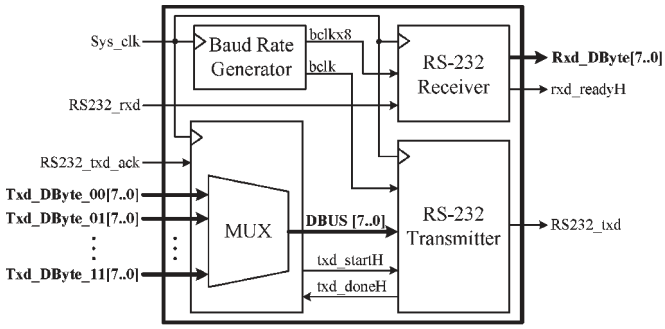

Fig. 19. Block diagram of the RS-232 communication circuit module.

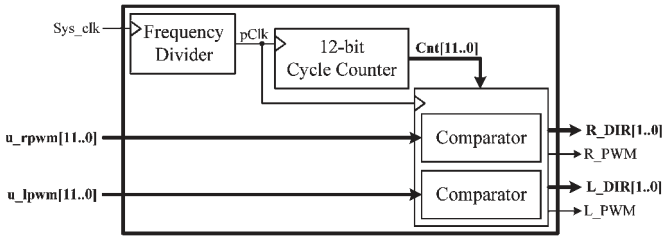

Fig. 20. Block diagram of the PWM generation circuit module.

data are formed as a data package which consists of two check bytes, 2-B tilt sensor data, 2-B gyroscope data, 4-B encoder data, and two default empty bytes. This design shows that the user can not only give control command to the TWIP but also easily monitor its real-time information by using a PC. Furthermore, based on the RS-232 protocol, the baud rate is selected as 9600 b/s (i.e., bclk $= 9600$ Hz) to meet the transmission requirement of the TWIP.

4) PWM Generation Circuit Module: According to the well-known PWM technique, this paper designs the PWM module to control the rim motors (see Figs. 4 and 20). In the PWM module, a frequency divider generates a 4-MHz counting clock (pClk) to modulate the PWM signal cycle rate as 10 kHz. Then, a cycle counter repeatedly generates a 12-B up-count wave (**Cnt[11..0]**) to compare with the motor control commands (**u_rpwm[11..0]** and **u_lpwm[11..0]**) for modulating the duty widths of the PWM signals (R_PWM and L_PWM). Thus, the rim motors can be controlled by the SoPC board via the PWM module.

APPENDIX B
KALMAN FILTER [32]

In the TWIP application, two KFs are used for the signal processing of the tilt sensor and the gyroscope, respectively. Based on (B.1)–(B.5) in Fig. 21, z_k is the actual measurement from the tilt sensor or the gyroscope. The initial conditions and the KF parameters are given as follows:

$$\begin{aligned} &\widehat{x}_0^- = 0,\ P_0^- = 1,\\ &R = 0.01,\ Q = 0.001,\\ &A = 1,\ B = 1,\ H = 1,\ u = 0. \end{aligned}$$

After the recursive computation, the z_k can be estimated as smoother $\widehat{x}_k$ for ϕ or $\dot{\phi}$ for eliminating the jumping effect and zigzag noise of the signals from the tilt sensor or the gyroscope.

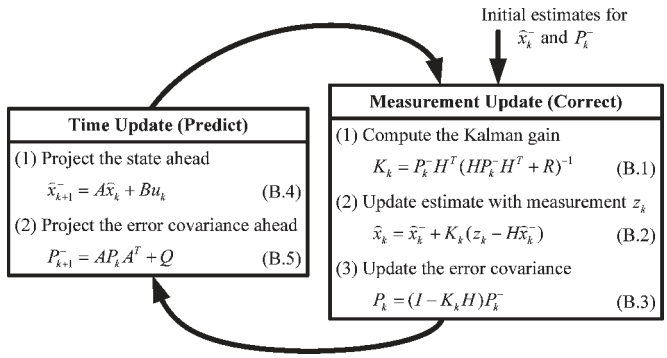

Fig. 21. Operation of the KF.

REFERENCES

[1] M. Baloh and M. Parent, "Modeling and model verification of an intelligent self-balancing two-wheeled vehicle for an autonomous urban transportation system," in *Proc. Conf. Comput. Intell. Robot. Auton. Syst.*, Singapore, Dec. 2003.

[2] O. Begovich, E. N. Sanchez, and M. Maldonado, "Takagi–Sugeno fuzzy scheme for real-time trajectory tracking of an underactuated robot," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 1, pp. 14–20, Jan. 2002.

[3] C.-Y. Chang, "Adaptive fuzzy controller of the overhead cranes with nonlinear disturbance," *IEEE Trans. Ind. Informat.*, vol. 3, no. 2, pp. 164–172, May 2007.

[4] C.-H. Chiu, "The design and implementation of a wheeled inverted pendulum using an adaptive output recurrent cerebellar model articulation controller," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1814–1822, May 2010.

[5] M. Fiacchini, A. Viguria, R. Cano, A. Prieto, F. R. Rubio, J. Aracil, and C. Canudas-de-Wit, "Design and experimentation of a personal pendulum vehicle," in *Proc. 7th Portuguese Conf. Autom. Control*, Lisboa, Portugal, Sep. 2006.

[6] F. Grasser, A. D'Arrigo, S. Colombi, and A. C. Rufer, "JOE: A mobile, inverted pendulum," *IEEE Trans. Ind. Electron.*, vol. 49, no. 1, pp. 107–114, Feb. 2002.

[7] Y.-S. Ha and S. Yuta, "Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot," *Robot. Auton. Syst.*, vol. 17, no. 1/2, pp. 65–80, Apr. 1996.

[8] T. S. Hall and J. O. Hamblen, "System-on-a-programmable-chip development platforms in the classroom," *IEEE Trans. Educ.*, vol. 47, no. 4, pp. 502–507, Nov. 2004.

[9] H.-C. Huang and C.-C. Tsai, "FPGA implementation of an embedded robust adaptive controller for autonomous omnidirectional mobile platform," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1604–1616, May 2009.

[10] S. Jung and S. s. Kim, "Hardware implementation of a real-time neural network controller with a DSP and an FPGA for nonlinear systems," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 265–271, Feb. 2007.

[11] E. Kim and H. Lee, "New approaches to relaxed quadratic stability condition of fuzzy control systems," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 523–534, Oct. 2000.

[12] Y.-S. Kung, R.-F. Fung, and T.-Y. Tai, "Realization of a motion control IC for X–Y table based on novel FPGA technology," *IEEE Trans. Ind. Electron.*, vol. 56, no. 1, pp. 43–53, Jan. 2009.

[13] T.-H. S. Li, S.-J. Chang, and Y.-X. Chen, "Implementation of human-like driving skills by autonomous fuzzy behavior control on an FPGA-based car-like mobile robot," *IEEE Trans. Ind. Electron.*, vol. 50, no. 5, pp. 867–880, Oct. 2003.

[14] T.-H. S. Li, S.-J. Chang, and W. Tong, "Fuzzy target tracking control of autonomous mobile robots by using infrared sensors," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 491–501, Aug. 2004.

[15] T.-H. S. Li, Y.-C. Yeh, J.-D. Wu, M.-Y. Hsiao, and C.-Y. Chen, "Multifunctional intelligent autonomous parking controllers for carlike mobile robots," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1687–1700, May 2010.

[16] W. Li, K. Tanaka, and H. O. Wang, "Acrobatic control of a pendubot," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 549–552, Aug. 2004.

[17] Z. Li and J. Luo, "Adaptive robust dynamic balance and motion controls of mobile wheeled inverted pendulums," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 1, pp. 233–241, Jan. 2009.

[18] Y.-W. Liang, S.-D. Xu, D.-C. Liaw, and C.-C. Chen, "A study of T–S model-based SMC scheme with application to robot control," *IEEE Trans. Ind. Electron.*, vol. 55, no. 11, pp. 3964–3971, Nov. 2008.
[19] S. W. Nawawi, M. N. Ahmad, and J. H. S. Osman, "Real-time control of a two-wheeled inverted pendulum mobile robot," in *Proc. World Acad. Sci. Eng. Technol.*, May 2008, vol. 29, pp. 214–220.
[20] K. Pathak, J. Franch, and S. K. Agrawal, "Velocity and position control of a wheeled inverted pendulum by partial feedback linearization," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 505–513, Jun. 2005.
[21] S. Sánchez-Solano, A. J. Cabrera, I. Baturone, F. J. Moreno-Velo, and M. Brox, "FPGA implementation of embedded fuzzy controllers for robotic applications," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1937–1945, Aug. 2007.
[22] T. Takei, R. Imamura, and S. Yuta, "Baggage transportation and navigation by a wheeled inverted pendulum mobile robot," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 3985–3994, Oct. 2009.
[23] H. Tanaka, K. Ohnishi, H. Nishi, T. Kawai, Y. Morikawa, S. Ozawa, and T. Furukawa, "Implementation of bilateral control system based on acceleration control using FPGA for multi-DOF haptic endoscopic surgery robot," *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 618–627, Mar. 2009.
[24] K. Tanaka, S. Hori, and H. O. Wang, "Multiobjective control of a vehicle with triple trailers," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 3, pp. 357–368, Sep. 2002.
[25] K. Tanaka, M. Iwasaki, and H. O. Wang, "Switching control of an R/C hovercraft: Stabilization and smooth switching," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 6, pp. 853–863, Dec. 2001.
[26] K. Tanaka and T. Kosaki, "Design of a stable fuzzy controller for an articulated vehicle," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 27, no. 3, pp. 552–558, Jun. 1997.
[27] K. Tanaka, T. Kosaki, and H. O. Wang, "Backing control problem of a mobile robot with multiple trailers: Fuzzy modeling and LMI-based design," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 28, no. 3, pp. 329–337, Aug. 1998.
[28] K. Tanaka, H. Ohtake, and H. O. Wang, "A practical design approach to stabilization of a 3-DOF RC helicopter," *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 2, pp. 315–325, Mar. 2004.
[29] K. Tanaka and H. O. Wang, *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. New York: Wiley, 2001.
[30] C.-C. Tsai, H.-C. Huang, and S.-C. Lin, "Adaptive neural network control of a self-balancing two-wheeled scooter," *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1420–1428, Apr. 2010.
[31] L. Vachhani, K. Sridharan, and P. K. Meher, "Efficient FPGA realization of CORDIC with application to robotic exploration," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4915–4929, Dec. 2009.
[32] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*. Chapel Hill, NC: Dept. Comput. Sci., Univ. North Carolina at Chapel Hill.

Cheng-Hao Huang was born in Taichung, Taiwan, in 1982. He received the B.S. degree from the Department of Electrical Engineering, Tamkang University, Tamsui, Taiwan, in 2004, and the M.S. degree from the Department of Electrical Engineering, National Central University, Jhongli, Taiwan, in 2006, where he is currently working toward the Ph.D. degree.

His research interests include fuzzy systems, intelligent control theories, embedded systems, and mobile robots.

Wen-June Wang (S'85–M'87–SM'00–F'08) was born in Hsinchu, Taiwan, in 1957. He received the B.S. degree in control engineering from National Chiao Tung University, Hsinchu, in 1980, the M.S. degree in electrical engineering from Tatung University, Taipei, Taiwan, in 1984, and the Ph.D. degree from the Institute of Electronics, National Chiao Tung University, in 1987.

He is currently a Chair Professor with the Department of Electrical Engineering, National Central University, Jhongli, Taiwan. He has published more than 120 journal papers and 120 conference papers. His research interests include fuzzy theory and systems, robust control, neural networks, and pattern recognition.

Dr. Wang was the recipient of the Distinguished Research Award from the National Science Council of Taiwan in 1999, 2001, and 2003. He serves as a member of the editorial board of numerous journals, including the *Asian Journal of Control*, the *International Journal of Electrical Engineers*, the IEEE TRANSACTIONS ON FUZZY SYSTEMS, and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS. He is also the Editor in Chief of the *International Journal of Fuzzy Systems*.

Chih-Hui Chiu was born in Taiwan. He received the B.S. degree in electrical engineering from the Tatung Institute of Technology, Taipei, Taiwan, in 1994, and the M.S. and Ph.D. degrees in electrical engineering from the National Central University, Jhongli, Taiwan, in 1996 and 2000, respectively.

He is currently an Assistant Professor with the Department of Electrical Engineering, Yuan Ze University, Jhongli. His research interests include fuzzy logic theory, intelligent control, robotics, and control theory applications.