IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 1, 2019, accepted December 18, 2019, date of publication December 30, 2019, date of current version January 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963110

Adaptive Nonlinear Control Algorithm for a Self-Balancing Robot

YUN SU[1,2,3], TING WANG[1,2], KAI ZHANG[1,2], CHEN YAO[1,2], AND ZHIDONG WANG[4]

[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[2]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China
[3]Shenyang Institute of Automation, University of Chinese Academy of Sciences, Beijing 100049, China
[4]Department of Advanced Robotics, Chiba Institute of Technology, Chiba 275-0016, Japan

Corresponding author: Yun Su (robosu12@gmail.com)

This work was supported by the National Key Research and Development Program of China under Grant 2017YFC0822201 and Grant 2017YFC0822203.

ABSTRACT We previously developed a novel composite wheel-leg-track explosive ordnance disposal (EOD) robot with high mobility, able to switch between a track or a self-balancing motion mode according to environmental conditions, named Scorpio. In this paper, we propose an adaptive nonlinear control algorithm for improving the stability of the robot in the self-balancing mode. First, a model of the dynamics of the robot was established, with which we designed the nonlinear cascade controller for combined balance and motion control. With our system, the attitude of the robot is estimated using a Kalman filtering algorithm. Based on this, an adaptive adjustment algorithm amends the parameters of the controller in real time according to the state of the robot, for improved stability. In addition, we formulated an adaptive zero-offset angle identification algorithm to compensate for deviations caused by changes to the robot's center of gravity (due to changes to its mechanical structure), ensuring that this stability could be maintained. Results of experiments conducted to verify their operation show that self-balancing control of Scorpio can be achieved with the proposed algorithms.

INDEX TERMS Adaptive control, center of gravity compensation, nonlinear controller, parameter adjustment, self-balancing robot.

I. INTRODUCTION

EOD (explosive ordnance disposal) robots have a wide range of applications in the fields of security, disaster relief, anti-terrorism, and hazardous environment navigation. Most existing EOD robots operate in tracked or wheeled modes [1]–[4], resulting in the adoption of over-constrained structures. The limitations of such structures means that EOD robots need to overcome large frictions when making a turn, especially when engaging in point-turn motion, causing damage to mechanical structures, and posing high requirements on motor performance. In addition, owing to the serious internal power consumption of the track, the operating efficiency of robots featuring this motion is relatively low. To solve this problem, we developed a novel composite wheel-leg-track EOD robot with crawler and self-balancing motion modes, named Scorpio. In the first mode, Scorpio uses crawler tracks to overcome obstacles in complex environments, and to climb stairs. The second mode is selected in situations when high speed, flexibility, and low energy consumption are required, such as when travelling on flat urban roads. Although Scorpio's environmental adaptability and flexibility of motion is advantageous, a more complex control algorithm is required for effective operation of this hybrid motion. Therefore, this paper focuses on control of Scorpio's self-balancing motion mode, where it effectively operates as a two-wheeled self-balancing robot (TWSBR).

TWSBRs are multi-variable, nonlinear, and strongly coupled underactuated systems, widely studied by a range of researchers. For instance, Felix Grasser and co-researchers at the Swiss Federal Institute of Technology developed JOE, a TWSBR in which basic balance and motion control functions were realized [5]. Researchers at the Georgia Institute of Technology created Golem Krang, a more complex humanoid robot with a TWSBR chassis, and a body consisting of a two-degree-of-freedom torso and a robotic arm [6]. A cascade PID

The associate editor coordinating the review of this manuscript and approving it for publication was Ning Sun.

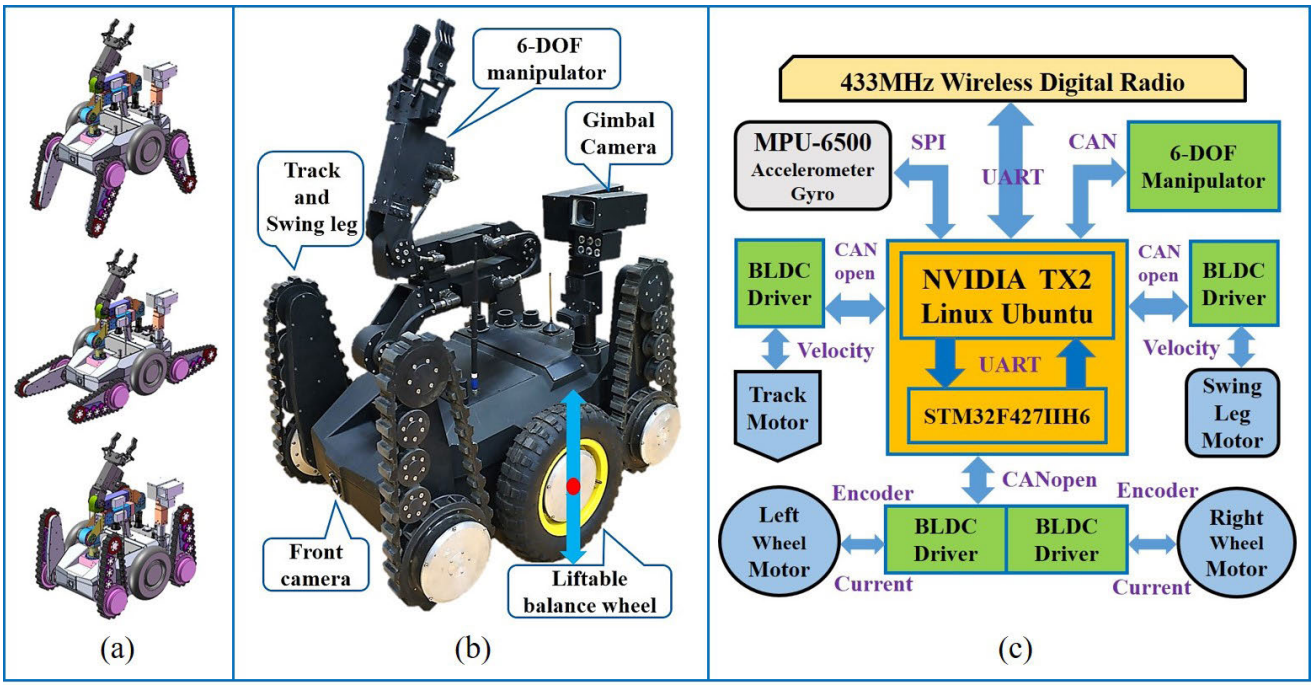


FIGURE 1. Overview of the Scorpio robot system. (a) Different robot postures in track mode. (b)Self-balancing mode. The balance wheel can be lifted vertically to return to track mode. (c) Control system structure.

algorithm is used to control the motion of the robot. Since the robot's center of gravity shifts according to the movement of its arms, this compensates for such deviations by calculating the dynamics of arm movement. However, it is unable to provide compensation for unknown deviations. Several alternative control algorithms have been studied to address similar problems in other TWSBRs. Iwendi *et al.* devised a PD-PI-P control algorithm, configured such that the attitude of the robot is adjusted by a PD controller, speed by a PI controller, and steering by a P controller. Park *et al.* [8] used a control moment gyroscope to adjust the balance of a robot, resulting in a small translation on application of external force. Unluturk and Aydogdu [9] employed a neural-network based technique using target angle deviation, target displacement deviation, and controller output data, for adaptive control of robot's balance on different surfaces. However, this technique requires advanced collection of data on the robot's operation on different surfaces, for classification. Similar adaptive techniques for control of TWSBRs were discussed in [10]–[13], each with associated drawbacks. From study of the existing research, we note the following problems associated with TWSBR control:

(1) Since TWSBRs are strongly nonlinear, linear controllers cannot guarantee their stability for large deviations from the linearization point [14]–[16]. Hence, different control strategies are required for a TWSBR in different states of operation.

(2) Self-balancing control of a TWSBR is affected by changes to its structure or load, which change its center of gravity, resulting in a random deviation of the angle of the balance point.

To address these problems, in this paper, we present an adaptive nonlinear control algorithm (ANC). Based on a model of the robot's dynamics, we developed a nonlinear cascade controller for self-balancing and motion control. A Kalman filter algorithm [17] is used to estimate the attitude of the robot, for classification of the current state of the robot. Controller parameters are subsequently modified according to relevant input and output data from the robot using an adaptive adjustment algorithm, for improved stability. Finally, deviations caused by changes to the robot's center of gravity, due to changes to its structure or dynamic components, are compensated using an adaptive zero-offset angle identification algorithm, ensuring that the robot can maintain its stability.

The rest of this paper is organized as follows. An overview of the Scorpio robot is provided in Section 2. A dynamic model of the robot is derived in Section 3. Section 4 introduces details of the proposed nonlinear controller, adaptive regulator, and center of gravity deviation compensator. Results of experiments conducted to verify the proposed algorithm are discussed in Section 5. Finally, the paper is summarized in Section 6.

II. ROBOT SYSTEM DESIGN

The structure of the Scorpio robot system, featuring a composite wheel-leg-track moving mechanism, is shown in Fig. 1. The chassis consists of four independently driven tracks and

two independently driven wheels. The swing legs at the front and rear are arranged coaxially, and are rotated around the concentric axis by an internal driving mechanism, to adjust their angle. An internal lifting mechanism adjusts the position of the balance wheel in the vertical direction.

When required to cross an obstacle or climb stairs, the robot is switched to crawler motion mode and its wheels are raised. In this mode, the robot can adjust the angle of the swing leg to different environments, as shown in Fig. 1(a). In addition, the robot is able to climb a staircase with a 35° incline, a 40° slope, and get across a 40 cm high vertical obstacle and a 50 cm wide trench. The robot can be switched to the self-balancing mode (Fig. 1(b)), for use when the road is flat, by retracting the swing legs and lowering the balance wheels. At this time, a maximum speed of 2 m/s can be achieved, with low energy consumption and flexible steering. The modes of motion can be switched at will, according to the operational environment, making Scorpio robots both environmentally adaptable and flexible.

An overview of the Scorpio control system is shown in Fig. 1(c). The control core consists of a main processor (NVIDIA jetson TX2) executing the motion control and planning algorithms, and a coprocessor (STM32F427IIH6) directing communication between the underlying hardware devices (i.e., drivers and sensors). Wheels and tracks are both driven by DC brushless hub motors, while an MPU-6500 inertial measurement unit (IMU) is employed to measure the angle and angular velocity of the robot. A CAN bus is used for real-time communication with the 6-DOF arms and the 2-DOF PTZ camera, with instructions relayed to the console remotely using a 433 MHz data transmission station and a 1.2 GHz image transmission station.

III. DYNAMIC MODEL

The Scorpio robot in the self-balancing mode can be modeled as an inverted pendulum, as shown in Fig. 2 (Fig. 2(a) is a side view and Fig. 2(b) is a top view of the robot). The top sphere of the pendulum represents the position of the robot's center of gravity, while θ_o is the value measured by the attitude sensor when the robot is at the balance point.

Based on [18]–[20], we present an analysis of the dynamics of the robot disturbed by external forces. A definition of the physical parameters used in modeling is included in Table 1. In the following, the " $\dot{}$ " and " $\ddot{}$ "operators refer to the first and second time derivatives, respectively. The resultant force on each wheel in the horizontal and vertical directions can be represented as follows:

$$m\ddot{x}_W = f_{HW} - H_W, \tag{1}$$

$$m\ddot{y}_W = f_{VW} - V_W - mg, \tag{2}$$

where H_W and V_W are applied force components, $\ddot{x}_W$ and $\ddot{y}_W$ are linear acceleration components, and f_{HW} and f_{VW} are friction components in the horizontal and vertical directions, respectively, and g is acceleration due to gravity. The resultant

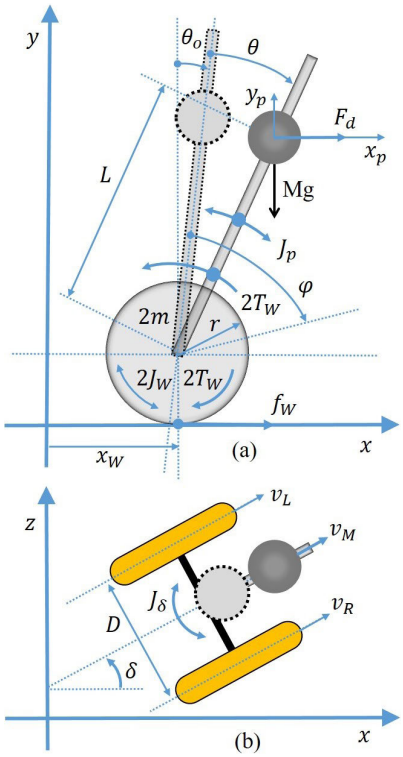


FIGURE 2. Diagram of the robot's coordinate system.

TABLE 1. Physical parameters of scorpio.

Parameter	Value	Definition
M	175 kg	Robot weight
m	8.9 kg	Balance wheel weight
r	0.19 m	Balance wheel radius
L	0.21 m	Distance between wheel's center and robot's center of gravity
D	0.45 m	Distance between left and right balance wheel
J_W	0.039 kg·m^2	Moment of inertia of balancing wheel
J_p	25.5 kg·m^2	Moment of inertia of the robot around the z axis
J_δ	20.9 kg·m^2	Moment of inertia of the robot around the y axis
K_m	1.59 N·m/A	Torque constant of hub motor
K_e	0.26 V/(r/min)	Opposing electromotive force constant of hub motor
r_a	0.37 Ω	Armature resistance of hub motor

torque for rotation of a wheel about its axle is given as:

$$\frac{J_W}{r}\ddot{x}_W = T_W - f_{HW}r, \tag{3}$$

where T_W is the applied torque.

The resultant force on the robot's body in the horizontal and vertical directions is given by,

$$M\ddot{x}_p = F_d + 2H_W, \tag{4}$$
$$M\ddot{y}_p = 2V_W - Mg. \tag{5}$$

The resultant torques for rotation of the robot's body around the horizontal (z) and the vertical (y) axis are as follows:

$$J_p\ddot{\theta} = 2V_W L\sin\theta - 2H_W L\cos\theta - 2T_W, \tag{6}$$
$$J_\delta\ddot{\delta} = \frac{H_{WL} - H_{WR}}{2}D, \tag{7}$$

where θ is the pitch angle, δ is the yaw angle, and H_{WL} and H_{WR} are applied force components on the left and right wheels, respectively, in the horizontal direction.

Using (1)–(7), the following equations can be obtained by eliminating intermediate variables.

$$\ddot{x}_W\left(2m+M+\frac{2J_W}{r^2}\right)+ML\left(\ddot{\theta}\cos\theta-\dot{\theta}^2\sin\theta\right)=\frac{2T_W}{r}, \tag{8}$$

$$J_p\ddot{\theta} + ML^2\ddot{\theta} - ML\cos\theta\ddot{x}_W = 2T_W - MgL\sin\theta, \tag{9}$$

$$\left(Dm + \frac{2J_\delta}{D} + \frac{DJ_W}{r^2}\right)\ddot{\delta} = \frac{T_{WL} - T_{WR}}{r}, \tag{10}$$

where T_{WL} and T_{WR} are applied torque components on the left and right wheels, respectively. By ignoring higher order terms and considering perturbations close to the balance point such that $\theta \approx 0$, $\sin\theta \approx \theta$, $\cos\theta \approx 1$, $\theta^2 \approx 0$, the model of the robot's dynamics can be linearized.

The operating principle of DC motors dictates the following relationship between electromagnetically-controlled torque (T_W), and armature voltage (U) and motor speed (n):

$$T_W = \frac{K_m}{r_a}U - \frac{K_m K_e}{r_a}n. \tag{11}$$

With $[x_W\ \dot{x}_W\ \theta\ \dot{\theta}\ \delta\ \dot{\delta}]^T$ selected as the state vector and substituting (11) for the wheel's torque, the state update equation is defined as below:

$$\begin{bmatrix}\dot{x}_W\\ \ddot{x}_W\\ \dot{\theta}\\ \ddot{\theta}\\ \dot{\delta}\\ \ddot{\delta}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\\ 0 & a_{22} & a_{23} & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & a_{42} & a_{43} & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & a_{66}\end{bmatrix}\begin{bmatrix}x_W\\ \dot{x}_W\\ \theta\\ \dot{\theta}\\ \delta\\ \dot{\delta}\end{bmatrix} + \begin{bmatrix}0 & b_{21} & 0 & b_{41} & 0 & b_{61}\\ 0 & b_{22} & 0 & b_{42} & 0 & b_{62}\end{bmatrix}^T\begin{bmatrix}U_L\\ U_R\end{bmatrix}, \tag{12}$$

where $K_x = 2m + M + \frac{2J_W}{r^2}$, $K_\theta = J_p + ML^2$, and the remaining factors in (12) are defined as follows:

$$a_{22} = \frac{2\,K_m K_e\left(MLr - ML^2 - J_p\right)}{\left(K_x K_\theta + M^2L^2\right)r_a r}$$
$$a_{23} = \frac{M^2L^2 g}{K_x K_\theta + M^2L^2}$$
$$a_{42} = -\frac{2\,K_m K_e\left(K_x r + ML\right)}{K_x K_\theta + M^2L^2}$$
$$a_{43} = -\frac{MgLK_x}{K_x K_\theta + M^2L^2}$$
$$a_{66} = -\frac{DK_m K_e}{\left(Dm + \frac{2\,J_\delta}{D} + \frac{DJ_W}{r^2}\right)r^2 r_a}$$
$$b_{21} = b_{22} = \frac{K_m\left(K_\theta - MLr\right)}{\left(K_x K_\theta + M^2L^2\right)r_a r}$$
$$b_{41} = b_{42} = \frac{K_m\left(K_x r + ML\right)}{\left(K_x K_\theta + M^2L^2\right)r_a r}$$
$$b_{61} = -b_{62} = \frac{DK_m}{\left(Dm + \frac{2\,J_\delta}{D} + \frac{DJ_W}{r^2}\right)r_a r}$$

Using the actual physical properties of the robot, (12) dictates how self-balancing control is achieved with the voltage of the left and right hub motors as system inputs. In practice, these voltages are not modified directly; rather, a cascade PID controller is used to modify the current of the motor, for better self-balancing control.

IV. ANC ALGORITHM

Changes to the robot's physical parameters (such as from changes to its load) and interference from external forces affect the quality of self-balancing control. As traditional PID algorithms cannot adjust controller parameters according to changes of state [21], we devised an adaptive nonlinear control algorithm for regulation of Scorpio's self-balance.

A block diagram of the control algorithm is shown in Fig. 3. Our technique makes use of a Kalman filter and a sliding window filter to estimate the attitude and velocity of the robot, for characterization of its current state. The core control system consists primarily of a balance subsystem comprising a speed loop and an attitude loop in the form of a cascade controller, and a steering subsystem. An extended state observer (ESO) is included outside the attitude loop, for estimation of disturbances to the system [22], [23]. The total output of the system is dictated by the combination of the balance and steering controllers, the parameters of which are adjusted by the adaptive controller according to relevant input and output data. Hence, the robot can obtain better control quality in different states. In addition, a center of gravity compensation mechanism is also included, to ensure random changes to the robot's center of gravity caused by changes to its mechanical structure or load can be corrected.

A. ROBOT POSE ESTIMATION

In this study, the Kalman filter combines measurements from the accelerometer and the gyroscope to provide an accurate estimate of the robot's attitude. The discrete state equation for this block is defined as follows:

$$\begin{bmatrix}\theta_k\\ \omega_k\\ bg_k\end{bmatrix} = \begin{bmatrix}1 & \Delta t & -\Delta t\\ 0 & 1 & -\Delta t\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\theta_{k-1}\\ \omega_{k-1}\\ bg_{k-1}\end{bmatrix} + \varsigma_{k-1} \tag{13}$$

$$Z_k = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\theta_k & \omega_k & bg_k\end{bmatrix}^T + \sigma_k \tag{14}$$

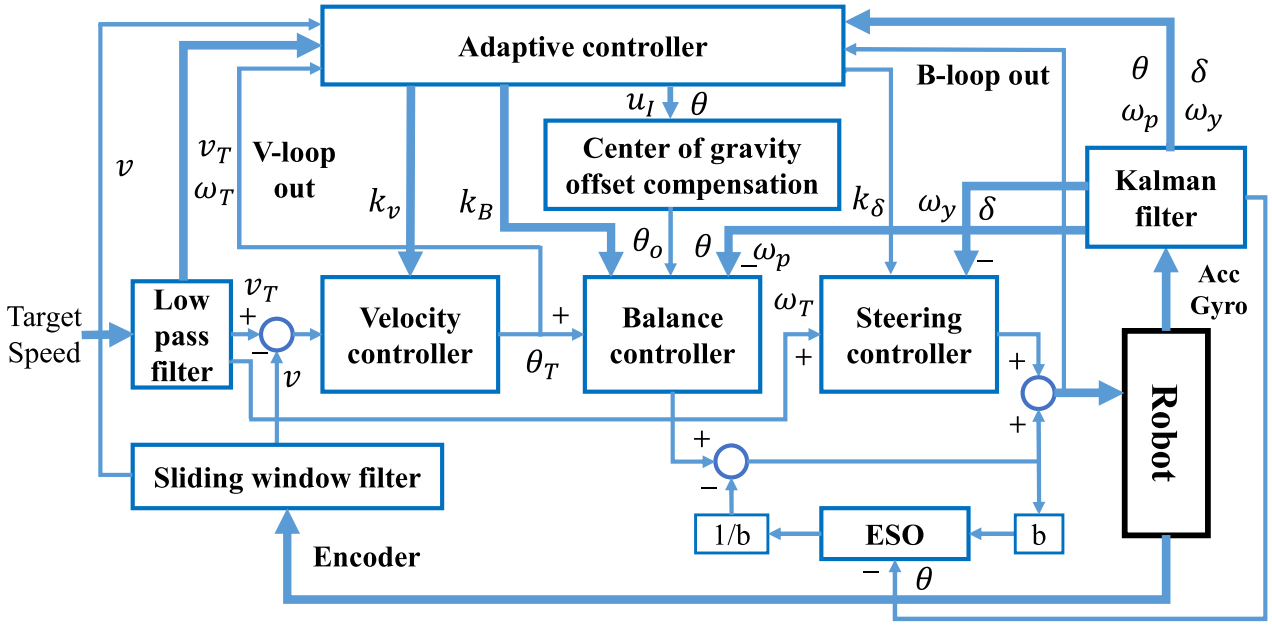


FIGURE 3. Block diagram of the control algorithm. Core control consists primarily of the velocity, balance, and steering controllera. The adaptive controller adjust parameters of each individual controller according to the robot's state. Center of gravity offset compensation is provided for adjusting the balance point in response to random changes to the center of gravity (e.g., when the mechanical structure or load on the robot changes).

where θ_k is the pitch angle, ω_k is the pitch angular velocity, bg_k is the gyro bias in the pitch direction, and Δt is the sampling period (5 ms).

The algorithm first makes an initial estimate of the current state of the system and error covariance based on historical information as follows:

$$\widehat{X}_{k|k-1} = A\widehat{X}_{k-1}, \tag{15}$$

$$\widehat{P}_{k|k-1} = A\widehat{P}_{k-1}A^{\mathrm{T}} + Q. \tag{16}$$

This estimate is subsequently modified according to observations from the sensors to obtain an updated estimate of the current state as below:

$$K_k = P_{k|k-1}H_k^{\mathrm{T}}\left(H_kP_{k|k-1}H_k^{\mathrm{T}} + R\right), \tag{17}$$

$$P_k = (I - K_kH_k)\,P_{k|k-1}, \tag{18}$$

$$\widehat{X}_{k|k} = \widehat{X}_{k|k-1} + K_k\left(Z_k - H_k\widehat{X}_{k|k-1}\right). \tag{19}$$

B. NONLINEAR CONTROLLER

Although linear controllers can balance the attitude of a TWSBR with appropriate definition of control parameters, their control domain is small as TWSBRs are fundamentally non-linear. Hence, this balance cannot be maintained for large changes to the robot's state [24], [25], and a nonlinear controller is required.

The basis of this control system is attitude control, provided by the balance and steering controllers in the inner loop of Fig. 3. The outputs of both these controllers are in the form shown in (20).

$$u_0(k) = k_1 fal(\theta, \lambda_1, h_1) + k_2 fal\left(\omega_p, \lambda_2, h_2\right), \tag{20}$$

$$fal(x, \lambda, h) = \begin{cases} \dfrac{x}{h^{1-\lambda}}, & |x| \le h \\ |x|^{\lambda}\, sign(x), & |x| > h, \end{cases} \tag{21}$$

where, in (21), λ is in the range [0, 1]. These functions fit high gains to low errors, and low gains to high errors, conforming to the nonlinear model of the robot. Hence, reasonable controller outputs are obtained for the different states of the robot.

To enhance the system's resistance to interference, we regard unmodeled dynamics and unknown external disturbances as the total disturbance of the system, and employ an ESO to compensate for these [26]–[28]. The state equation for this ESO is given below,

$$\begin{cases} e = z_1(k) - y(k) \\ fe = fal(e, \lambda_1, h) \\ fe_1 = fal(e, \lambda_2, h) \\ z_1(k+1) = z_1(k) + h\left(z_2(k) - \beta_{01}e\right) \\ z_2(k+1) = z_2(k) + h\left(z_3 - \beta_{02}fe + bu\right) \\ z_3(k+1) = z_3(k) + h\left(-\beta_{03}fe_1\right). \end{cases} \tag{22}$$

where $z_1(k)$ and $z_2(k)$ are the pitch angle and the pitch angular velocity of the robot, respectively, both estimated by the Kalman filter, and $z_3(k)$ is a real-time estimate of the total disturbance. By compensating it to the output, this function replaces the effect of integral control while avoiding the side effects of integral feedback. Thus, the total output of the attitude loop is given as,

$$u_b(k) = u_0(k) - \frac{z_3(k)}{b}. \tag{23}$$

The aim of the steering loop is to ensure the target angular velocity is maintained. However, when this velocity is zero, random deviations of the gyroscope (which senses angular velocity) mean that the rotation of the robot cannot be eliminated simply through the control of the angular

velocity feedback. Therefore, data from a magnetic compass is combined with the data from the gyroscope, to estimate the yaw angle. The steering loop is thus controlled using this angle and angular velocity as feedback, eliminating unwanted rotation.

As the outer loop, the speed loop regulates the linear velocity of the robot, with its output being used as the target angle of the attitude loop. However, as the balance control loop aims to return the robot's center of gravity to its balance point, the speed loop can be considered an interference to the attitude loop. Dramatic changes to the output of the speed loop can thus cause the robot's attitude to oscillate or diverge from the control point. Hence, the speed loop control law was defined using the non-linear combination below,

$$u_v(k) = k_1 \cosh(e_0, \omega_1) + k_2 \cosh(e_1, \omega_2) + k_3 \cosh(e_2, \omega_3), \quad (24)$$

$$\cosh(e, \omega) = \frac{\exp(\omega e) + \exp(-\omega e)}{2}. \quad (25)$$

As the cosh function fits low gain to low errors, and high gain to high errors, the output of the speed loop is small close to the origin, and increases rapidly as disturbances cause larger deviations from the origin. This function thus ensures the stability of the robot close to the origin, as well as a rapid response to disturbances.

C. ADAPTIVE REGULATOR

While the nonlinear controller described above is able to provide motion and balance control for the robot, with fixed parameters, this regulation is not satisfactory when changes to the robot's state or its physical parameters are encountered, due to the coupling of speed and attitude control. Therefore, we designed an adaptive regulator to dynamically adjust the parameters of the controller. The adjustment laws for the speed loop and attitude loop are shown in (26) and (27), respectively.

$$k_v(t+1) = \frac{k_v(t)}{\cosh(\theta, \omega_1)}, \quad (26)$$

$$k_B(t+1) = \cosh(v, \omega_2)\, k_B(t). \quad (27)$$

It can be noted that the velocity loop parameter is decreased as function of the hyperbolic secant of the pitch angle (θ), and the balance loop parameter is increased as a function of the hyperbolic cosine of the linear velocity (v). This adjustment law weights the response of the controller towards the output of the attitude loop, thus conforming to the principle that attitude control is the basis of the self-balancing robot. From other investigation, the value of ω in (26) should be selected such that when the pitch angle of the robot is half of its maximum value, the slope of the hyperbolic secant function is maximized. It should be pointed out that this adjustment law can significantly reduce the amplitude of the motor current when the robot is switched from the free state to the self-balancing mode, making the robot start more smoothly.

Adaptive control of the steering loop requires consideration of the yaw angle, obtained using a combination of data from the magnetic compass and the gyroscope. As compass readings are affected by magnetic fields (which is affected by operation of the electric motor), it can only be used to reduce the robot's rotation in a fixed position. When the target linear velocity or angular velocity is not zero, only the angular velocity is used for closed-loop control. Therefore, the steering loop adjustment law is given as,

$$k_\delta = \begin{cases} k_0 & |v_T| + |\omega_T| = 0 \\ 0 & |v_T| + |\omega_T| > 0. \end{cases} \quad (28)$$

D. COMPENSATOR FOR CENTER OF GRAVITY DEVIATION

An additional consideration for the control algorithm is its response to random variations to the robot's center of gravity (caused by changes to the load or mechanical structure of the robot), which subsequently affect the location of the balance point. Although the integral output of the speed loop can compensate for some center of gravity deviation, balance control deteriorates close to the integral limit, which is set to one-third of the maximum output to reduce the spring effect (when disturbed, the robot moves back due to the integration effect). Hence, we included a separate compensator for center of gravity deviations capable of identifying and adjusting for modifications to the balance point in real time.

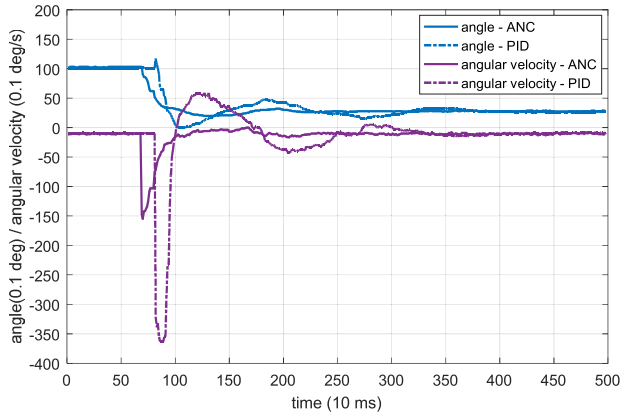
The coupled set of equations dictating the output of the center of gravity deviation compensator is as in (29). Here, $u_I(k)$ is the integral output of the speed loop, which is converted into an angle by dividing by d_0, while h is the convergence factor, calculated using h_1 and h_2, the information values of the angle variables ($\theta_I(k)$ and $\theta(k)$, respectively). Average filtering is conducted on $\theta_I(k)$ and $\theta(k)$ by passing them through a sliding window with a set length. $\sum_1$ is variance of $\theta_I(k)$ in this sliding window and $\sum_2$ is the variance of $\theta(k)$. Finally, $\theta_C(k)$is the compensation angle. Inspection of the equations for the information values shows that these parameters are inversely proportional to variance. Hence, if the variance of the data is large, the information value is small, and $\theta_C(k)$ updates slowly or does not update.

$$\begin{cases} \theta_I(k) = \dfrac{u_I(k)}{d_0} \\ h = \mu h_1 h_2, \quad h_1 = \dfrac{1}{\sum_1}, h_2 = \frac{1}{\sum_2} \\ \theta_\sum(k) = \dfrac{\sum_2}{\sum_1 + \sum_2}\theta_I(k) + \dfrac{\sum_1}{\sum_1 + \sum_2}\theta(k) \\ \theta_C(k+1) = (1-h)\,\theta_C(k) + h\theta_\sum(k) \end{cases} \quad (29)$$

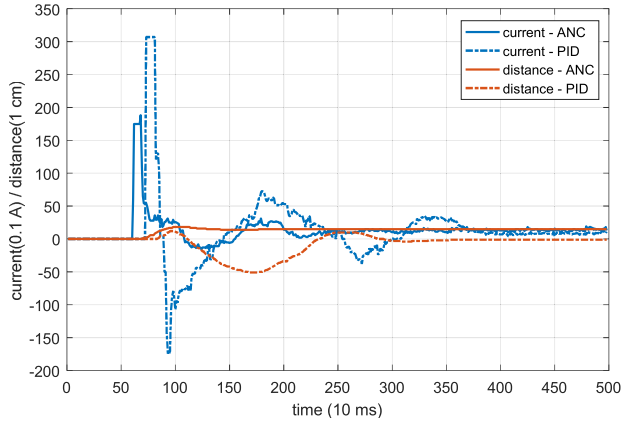
To prevent the robot from oscillating as a result of sudden changes to the attitude, a first-order low-pass filter function is applied to the compensation angle. Hence, the output to the attitude loop is as below:

$$\theta_o(k+1) = (1-c_0)\,\theta_0(k) + c_0\theta_C(k). \quad (30)$$

In addition, compensator update is turned off (by zeroing μ in the convergence factor) when the difference between the pitch angle and the balance point angle of the robot is below a certain threshold, or when the robot is moving under remote



(a)



(b)

FIGURE 4. Comparison of the robot's change in (a) attitude, and (b) displacement and motor current when self-balancing mode is activated.

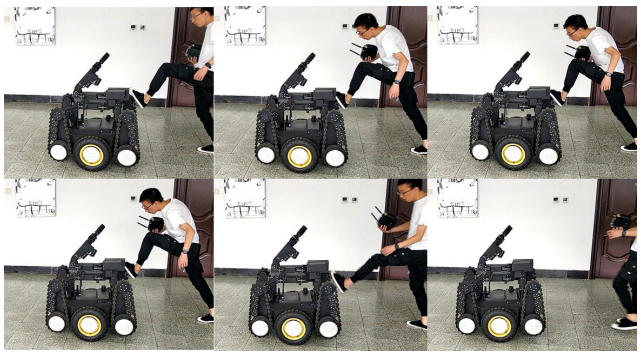
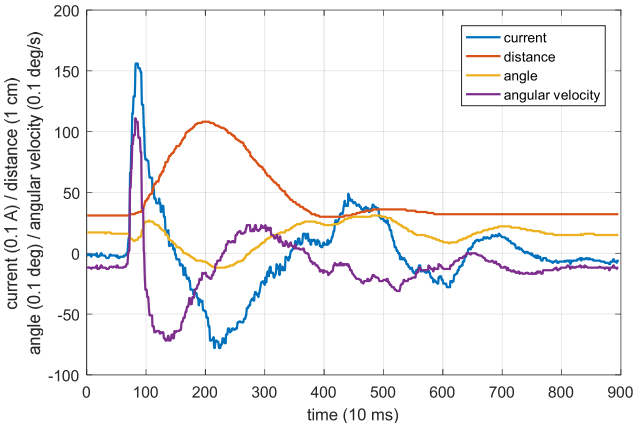
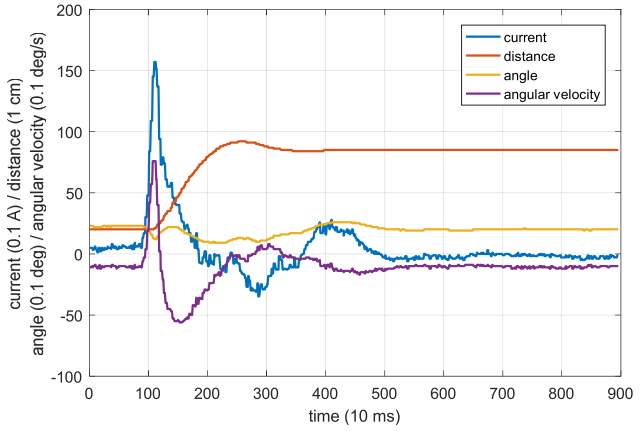


FIGURE 5. Application of external force to robot.

control. In the former case, smaller differences between the pitch angle of the robot and the balance point, indicate that deviations to the robot's center of gravity are minor, and no compensation is needed. Similarly, as the robot's attitude can change frequently under remote control, compensation should be stopped to avoid errors.



(a)



(b)

FIGURE 6. Response of the robot's state variables to external force interference when controlled using the (a) PID algorithm, and (b) ANC algorithm.

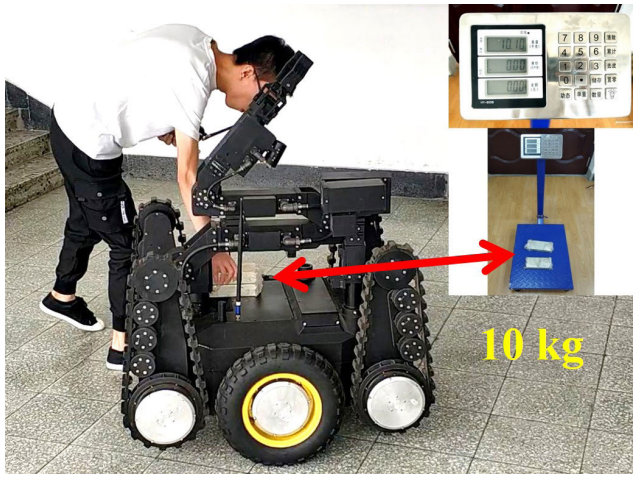


FIGURE 7. Add random weights to the robot and change the center of gravity of the robot.

V. EXPERIMENTAL RESULTS AND ANALYSIS

We performed a series of experiment to verify the effect of the proposed algorithm on control of Scorpio.

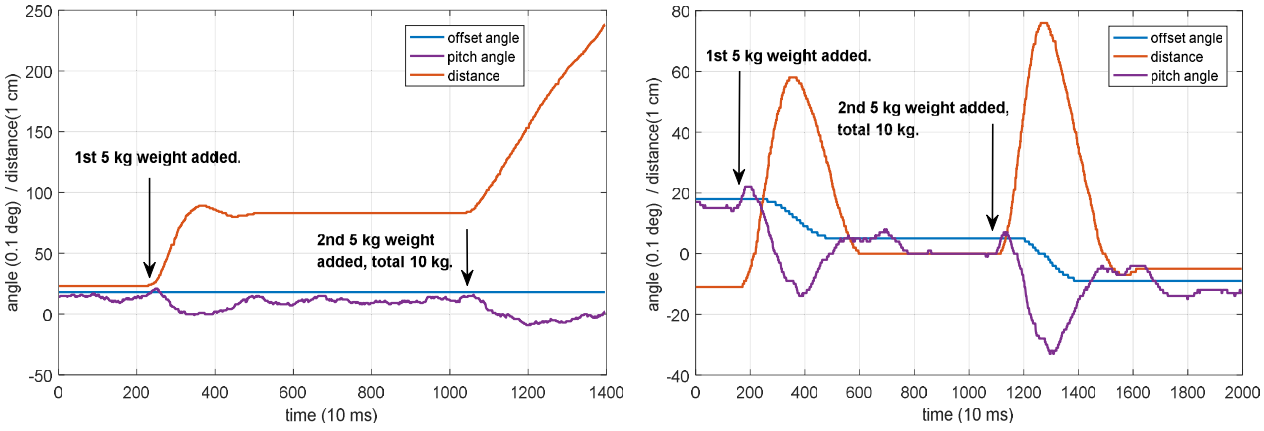


FIGURE 8. Effect of deviations to the robot's center of gravity on pitch angle and moving distance, when controlled using the (a) PID algorithm, and (b) ANC algorithm.

Using measurements of individual state variables, we evaluated the robot's response to switches to the mode of motion, interference from external forces, and changes to the center of gravity. In addition, we compared these results to the responses obtained using a standard PID algorithm.

A. MOTION MODE SWITCH

Switching experiments were conducted to verify the rapidity and stability of the robot's motion response on selection of the self-balancing mode from static. Using a range of metrics, we observe that the ANC algorithm provides the robot with a smoother start, and reduces the displacement in position caused by changing the mode of motion. Fig. 4(a) shows the change in the pitch and the angular velocity in the pitch direction during the mode switch. The solid lines depict the results obtained with the proposed ANC algorithm, while the dotted lines depict the results obtained with the linear PID algorithm. A significant reduction in the angular velocity of the robot is observed when the ANC algorithm is adopted, and the angle change is smoother, so that the overall attitude change is more stable. Moreover, the ordinary PID algorithm overshoots the control quantity, and oscillations in robot attitude are noted.

Curves depicting the displacement of the robot and changes to the motor current are shown in Fig. 4(b), for further characterization of the responses of the different algorithms. Oscillations in current can be observed with the PID algorithm, which are reflected in the oscillations in the displacement, pitch, and angular velocity. In contrast, with the ANC algorithm the initial current is reduced, and there is only a small displacement in position when the mode is switched.

B. EXTERNAL FORCE INTERFERENCE

To verify the stability of the control provided by the ANC algorithm, we conducted experiments introducing interference from an external force, up to a maximum of 150 N, during self-balancing, as shown in Fig. 5. Curves depicting the displacement of the robot, and the changes in motor current and robot attitude are shown in Fig. 6.

From the above, it can be seen that while the peak motor current is similar with both algorithms, there are only slight changes to the robot's attitude using the ANC algorithm, which can be stabilized quickly. In addition, as this algorithm employs a smaller integral limit in the speed loop, the rebound displacement of the robot is small, i.e., the "spring effect" is weak. In contrast, the linear PID algorithm requires a larger integral limit for realization of the anti-interference ability and to resist the center of gravity deviation, creating a stronger "spring effect."

C. CENTER OF GRAVITY CHANGE

To verify the operation of the center of gravity compensator included in the control algorithm, we conducted additional experiments as follows. With the robot in self-balancing mode, two 5 kg weights were placed on its chassis at random, at different times, to change its center of gravity. A variety of individual state variables were subsequently measured, to characterize the effect of this system. Fig. 7 depicts an image of the center of gravity deviation experiment being conducted.

Curves depicting the displacement of the robot, and changes to the robot attitude and zero-offset angle at balance point are shown in Fig. 8. It can be seen that, using either the ANC or PID algorithms, the robot is able to regain its balance after the first 5 kg weight is placed. However, while the compensation provided by the ANC algorithm gradually modifies the balance point to reflect the new center of gravity, such that the displacement of the robot to its original position is minimized, no such provision is made with the PID algorithm. Hence, the original position of the robot is not recovered.

With the addition of the second 5 kg weight, the integral limit of the PID algorithm is reached. Hence, with this

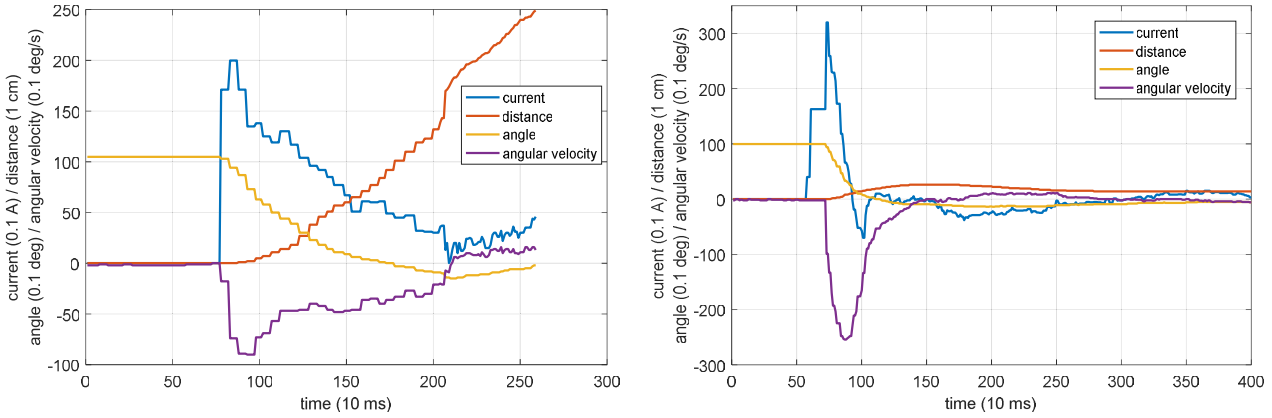


FIGURE 9. Response of the robot's state variables when switching motion mode after the robot's center of gravity changed randomly, when controlled using the (a) PID algorithm, and (b) ANC algorithm.

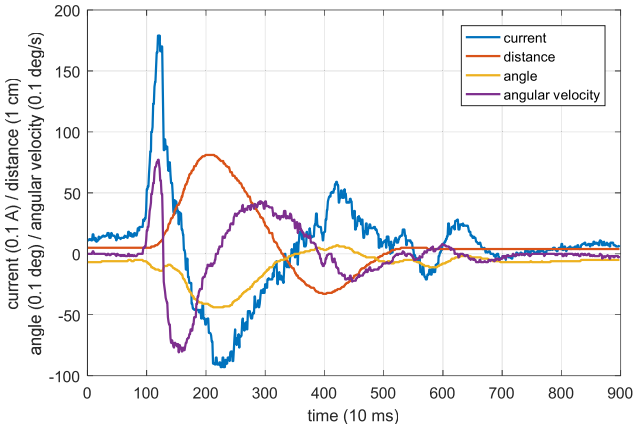


FIGURE 10. Response of the robot's state variables to external force interference after the robot's center of gravity changed randomly.

algorithm, the robot accelerates until attitude divergence, as the deviation of the center of gravity is excessive. In contrast, with the ANC algorithm, the robot regains its balance due to the active compensation provided.

The mode switch experiment was repeated following modification of the robot's center of gravity (i.e., the robot was loaded with a 5 kg weight). The results of this experiment are shown in Fig. 9, where it can be noted that during the switch from static to self-balancing mode, the robot experiences a large displacement to its position under the control of the linear PID algorithm. In contrast, while there is a displacement in position with the ANC algorithm, this is almost equal to the one experienced without a deviation to the center of gravity. However, there is an increase to the initial motor current on switching the motor modes, indicating that the ability to compensate for deviations to the robot's center of gravity depends on the maximum driving current and the peak torque of the motor. Finally, the external force interference test was also repeated, with the results of these experiments indicating that the ANC algorithm can keep the robot balanced after its center of gravity is modified, as shown in Fig. 10.

VI. CONCLUSION

In this paper we presented an adaptive nonlinear control algorithm for control of a two-wheel self-balancing robot. Based on a nonlinear model of robot dynamics, we designed a nonlinear controller for balance and steering control of the robot. In addition, we designed an adaptive regulator, which dynamically adjusts the parameters of the controller according to the current state of the robot and the controller output, for improved stability. An additional compensator for modifying the zero-offset angle for the balance point is also included, in consideration of changes to the mechanical structure or other components of the robot that can lead to random variations to the location of its center of gravity. In this way, the robot can achieve consistent control during its initialization and operation, and after its center of gravity has been modified. Experiments conducted to verify the effect of the proposed algorithm on the Scorpio robot indicated improved performance when compared with the control provided by a linear PID algorithm.

REFERENCES

[1] J. Ma, X. Li, and C. Yao, "Dynamic modeling and analysis for obstacle negotiation of ground mobile robot," *Robot*, vol. 30, no. 3, pp. 273–278, 2008.

[2] J. Xin, X. Li, and Z. Wang, "Performance analysis of track-leg mobile robot in unstructured environment," *Robot*, vol. 26, no. 1, pp. 35–39, 2004.

[3] Y. Liu and G. Liu, "Track–stair interaction analysis and online tipover prediction for a self–reconfigurable tracked mobile robot climbing stairs," *IEEE/ASME Trans. Mechatron.*, vol. 14, no. 5, pp. 528–538, Oct. 2009.

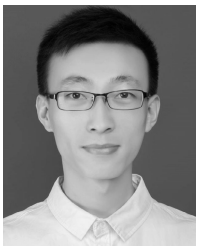
[4] Y. Chang, "Method of kinematic modeling of wheeled mobile robot," *J. Mech. Eng.*, vol. 46, no. 5, p. 30, 2010.

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

[6] M. Stilman, J. Olson, and W. Gloss, "Golem Krang: Dynamically stable humanoid robot for mobile manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 3304–3309.

[7] C. Iwendi, M. A. Alqarni, J. H. Anajemba, A. S. Alfakeeh, Z. Zhang, and A. K. Bashir, "Robust navigational control of a two–wheeled self–balancing robot in a sensed environment," *IEEE Access*, vol. 7, pp. 82337–82348, 2019.

[8] J.-H. Park and B.-K. Cho, "Development of a self-balancing robot with a control moment gyroscope," *Int. J. Adv. Robot. Syst.*, vol. 15, no. 2, Mar. 2018, Art. no. 172988141877086.
[9] A. Unluturk and O. Aydogdu, "Adaptive control of two-wheeled mobile balance robot capable to adapt different surfaces using a novel artificial neural network-based real-time switching dynamic controller," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 2, Apr. 2017, Art. no. 172988141770089.
[10] 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.
[11] S.-C. Lin, C.-C. Tsai, and H.-C. Huang, "Adaptive robust self–balancing and steering of a two–wheeled human transportation vehicle," *J. Intell. Robot. Syst.*, vol. 62, no. 1, pp. 103–123, Apr. 2011.
[12] F. Dai, X. Gao, S. Jiang, W. Guo, and Y. Liu, "A two-wheeled inverted pendulum robot with friction compensation," *Mechatronics*, vol. 30, pp. 116–125, Sep. 2015.
[13] C. Yang, Z. Li, and J. Li, "Trajectory planning and optimized adaptive control for a class of wheeled inverted pendulum vehicle models," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 24–36, Feb. 2013.
[14] K. Heong Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. Contr. Syst. Technol.*, vol. 13, no. 4, pp. 559–576, Jul. 2005.
[15] A. Polyakov, "Nonlinear feedback design for fixed–time stabilization of linear control systems," *IEEE Trans. Autom. Control*, vol. 57, no. 8, pp. 2106–2110, Aug. 2012.
[16] C. Wen, J. Zhou, Z. Liu, and H. Su, "Robust adaptive control of uncertain nonlinear systems in the presence of input saturation and external disturbance," *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1672–1678, Jul. 2011.
[17] E. Leffens, F. Markley, and M. Shuster, "Kalman filtering for spacecraft attitude estimation," *J. Guid., Control, Dyn.*, vol. 5, no. 5, pp. 417–429, Sep. 1982.
[18] A. Almeshal, K. Goher, and M. Tokhi, "Dynamic modelling and stabilization of a new configuration of two-wheeled machines," *Robot. Auto. Syst.*, vol. 61, no. 5, pp. 443–472, May 2013.
[19] J.-X. Xu, Z.-Q. Guo, and T. H. Lee, "Design and implementation of integral sliding–mode control on an underactuated two–wheeled mobile robot," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3671–3681, Jul. 2014.
[20] J. Huang, F. Ding, T. Fukuda, and T. Matsuno, "Modeling and velocity control for a novel narrow vehicle based on mobile wheeled inverted pendulum," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 5, pp. 1607–1617, Sep. 2013.
[21] K. Pathak, J. Franch, and S. 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.
[22] J. Han, "From PID to active disturbance rejection control," *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 900–906, Mar. 2009.
[23] W. Chen, J. Yang, L. Guo, and S. Li, "Disturbance-observer-based control and related methods-an overview," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1083–1095, Feb. 2016.
[24] 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.
[25] J. Huang, Z.-H. Guan, T. Matsuno, T. Fukuda, and K. Sekiyama, "Sliding–mode velocity control of mobile–wheeled inverted–pendulum systems," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 750–758, Aug. 2010.
[26] D. Sun, "Comments on active disturbance rejection control," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3428–3429, Dec. 2007.
[27] W. Xue, W. Bai, S. Yang, K. Song, Y. Huang, and H. Xie, "ADRC with adaptive extended state observer and its application to air–fuel ratio control in gasoline engines," *IEEE Trans. Ind. Electron.*, vol. 62, no. 9, pp. 5847–5857, Sep. 2015.
[28] J. Huang, S. Ri, L. Liu, Y. Wang, J. Kim, and G. Pak, "Nonlinear disturbance observer–based dynamic surface control of mobile wheeled inverted pendulum," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 6, pp. 2400–2407, Nov. 2015.



YUN SU received the B.S. degree from the Department of Automation, Chang'an University, China, in 2015. He is currently pursuing the Ph.D. degree with the Shenyang Institute of Automation, University of Chinese Academy of Sciences. His current research interests include robot collaboration control, robot localization and mapping algorithm, and vision-based multisensor fusion slam.

TING WANG received the B.S degree from the Department of Automation, Dalian University of Technology, in 2001, and the Ph.D. degree in pattern recognition and intelligent systems from the Shenyang Institute of Automation, Chinese Academy of Sciences, in 2007. Since then, he has been working with the Shenyang Institute of Automation. His research interests include robot control, special robot technology and pattern recognition, and intelligent systems.

KAI ZHANG received the B.S. and M.S. degrees in mechanical manufacturing and automation from the Nanjing University of Aeronautics and Astronautics, in 2011 and 2014, respectively. Since then, he has been working with the Shenyang Institute of Automation, Chinese Academy of Sciences. His research interests include EOD robot, inspection robot, and mobile mechanical arm.

CHEN YAO received the B.S. degree in engineering from Southeast University, in 1985. Since then, he has been working with the Shenyang Institute of Automation, Chinese Academy of Sciences, where he is currently a Professor. He has presided over several National Natural Science Foundation projects and has long been engaged in research on special robot control and related fields.

ZHIDONG WANG received the B.S. degree in control engineering from the Beijing University of Aeronautics and Astronautics, in 1987, and the Ph.D. degree in mechanical engineering from Tohoku University, Japan, in 1995. He is currently a Professor of the Advanced Robotics Department, Chiba Institute of Technology, Japan. His research interests include intelligent systems and dynamic control theory, multirobot self-coordinated control systems, and implementation of human–computer interaction type cooperative operation.