



# Adaptive robust control of Mecanum-wheeled mobile robot with uncertainties

Veer Alakshendra · Shital S. Chiddarwar

Received: 2 June 2016 / Accepted: 27 October 2016 / Published online: 19 November 2016  
© Springer Science+Business Media Dordrecht 2016

**Abstract** This paper presents a novel implementation of an adaptive robust second-order sliding mode control (ARSSMC) on a mobile robot with four Mecanum wheels. Each wheel of the mobile robot is actuated by separate motors. It is the first time that higher-order sliding mode control method is implemented for the trajectory tracking control of Mecanum-wheeled mobile robot. Kinematic and dynamic modeling of the robot is done to derive an equation of motion in the presence of friction, external force disturbance, and uncertainties. In order to make the system robust, second-order sliding mode control law is derived. Further, adaptive laws are defined for adaptive estimation of switching gains. To check the tracking performance of the proposed controller, simulations are performed and comparisons of the obtained results are made with adaptive robust sliding mode control (ARSMC) and PID controller. In addition, a new and low-cost experimental approach is proposed to implement the proposed control law on a real robot. Experimental results prove that without compromising on the dynamics of the robot real-time implementation is possible in less computational time. The simulation and experimental results obtained confirms the superiority of ARSSMC over

ARSMC and PID controller in terms of integral square error (ISE), integral absolute error (IAE), and integral time-weighted absolute error (ITAE), control energy and total variance (TV).

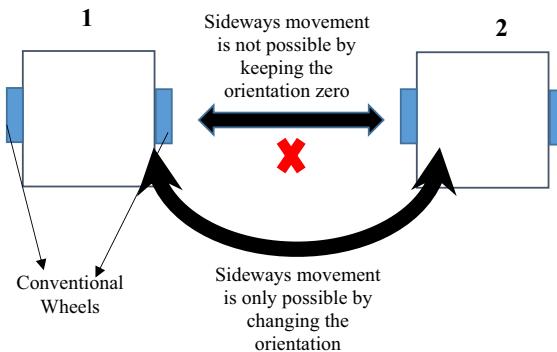
**Keywords** Mecanum wheel · Robust tracking · Adaptive control · Second-order sliding mode control

## 1 Introduction

In recent years, the study of mobile robots (conventional and omnidirectional) subjected to complex environment has gained popularity. Owing to advantages such as better maneuverability, ability to turn in any direction with zero turning radius, and capability to move in confined spaces, omnidirectional wheeled mobile robots are being used in a variety of homes and industrial applications such as omniwheel chairs, fork lifter, manipulators etc. Among various types of omnidirectional mobile robots, mobile robot with four Mecanum wheels is one of them [1]. Each Mecanum wheel is driven by a separate DC motor and has a series of rollers angled at 45° to its hub circumference. Compared to conventional mobile robots, Mecanum-wheeled mobile robots can move in sideways (Figs. 1, 2) and even along a curved path, keeping the orientation about its center of gravity zero (Figs. 3, 4). However, due to the use of four separate motors, its motion control becomes challenging in the presence of uncertainties.

**Electronic supplementary material** The online version of this article (doi:[10.1007/s11071-016-3179-1](https://doi.org/10.1007/s11071-016-3179-1)) contains supplementary material, which is available to authorized users.

V. Alakshendra (✉) · S. S. Chiddarwar  
Department of Mechanical Engineering, Visvesvaraya  
National Institute of Technology, Nagpur 40010, India  
e-mail: alakshendra.veer@gmail.com



**Fig. 1** Conventional wheeled mobile robot sideways movement

The kinematic and dynamic modeling of an omnidirectional wheel platform has been investigated by many researchers [2]. Muir and Neuman [3] derived the kinematic equations for Mecanum-wheeled Uranus mobile robot. A practical approach was proposed by Conceicao et al. [4] to model an omnidirectional mobile robot. Tlale and Villiers [5] presented the dynamics of Mecanum-wheeled mobile robot considering the frictional forces. Later derivation of an equation of motion focusing on the contact point of the wheel was proposed by Villiers and Tlale [6]. Although, the equation of motion of the mobile robot has been already derived [5, 6], the current work required more generalized dynamic equation of motion in a standard form and input as DC motor voltage. Thus, a new set of equations has been proposed in this work.

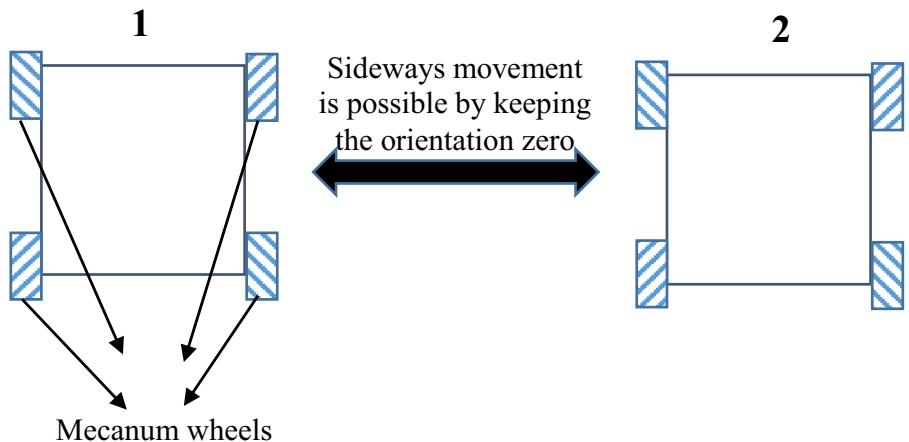
Considering the movement of the mobile robot in an environment subjected to uncertainties, nonlinear controllers (sliding mode control,  $H_\infty$  control, neural network, fuzzy control, etc.) have better efficacy to minimize the tracking error compared to linear controllers (PID). Yang and Kim [7] implemented sliding mode control for trajectory tracking of a wheeled mobile robot without considering the uncertainties. Viet et al. [8] presented a sliding mode controller to track the trajectory of a three wheel omnidirectional mobile manipulator in presence of structured as well as unstructured uncertainties. Fierro and Lewis [9] implemented neural network control of a mobile robot in the absence of friction forces. A trajectory generation and optimal control algorithm for a four-wheel omnidirectional vehicle under limited friction condition was proposed by Purwin and Andrea [10]. Xu et al. [11] utilized neural network concept and built a robust controller

for a wheeled mobile robot in presence of unstructured uncertainties. Ryu and Agrawal [12] designed a kinematic and dynamic controller based on the differential flatness framework. However, none of the papers implemented their proposed control law on a mobile robot with Mecanum wheels.

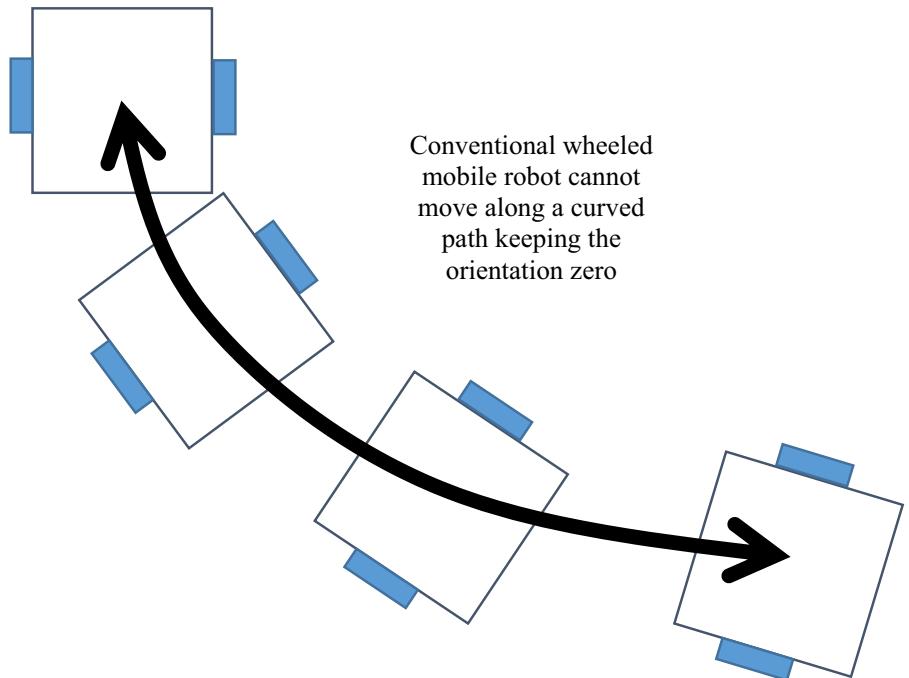
Sliding mode control (SMC) [13–15] is one of the powerful approaches to control a dynamic system subjected to the uncertainties where system trajectories are bought on the sliding surface and a switching function is applied to ensure that these system trajectories stay on the sliding surface after reaching phase. However, high switching gain can lead to oscillations of the state trajectories around the sliding manifold, resulting in undesirable chattering effect [16–18]. Thus, to reduce the chattering effect, efficacy of second-order sliding mode control (SSMC) has already been reported in the earlier research works [19–21]. In first-order sliding mode control approach, the sliding function has one relative degree, i.e., the total control input acts only on the first derivative of the sliding function, whereas in SSMC, the derived control input acts on the second derivative of the sliding function. Salgado and Jouvelcel [22] applied the higher-order sliding mode control law to control the diving of an autonomous underwater vehicle. Later, Mihoub et al. [23] used the approach in a chemical reactor application. Apart, from having the advantage of reducing chattering effect, SSMC also has better error convergence accuracy.

As the mobile robot system is a multiple-input–multiple-output system (MIMO), the effect of uncertainties can be severe if the uncertainties are bounded but unknown. For example, if a constant switching gain is selected using trial-and-error method, this may lead to increase in control energy. Moreover, if the magnitude of uncertainty increases abruptly and there is no provision of auto tuning the switching gains, the robot will deviate from its desired trajectory. One of the best and most common approaches to tackle such situations is to make the controller adaptive. Chen et al. [24, 25] implemented an adaptive sliding mode control on a wheeled mobile robot. An adaptive sliding mode control combined with backstepping technique is presented by Chen et al. [26] to control a mobile manipulator in the presence of disturbances. Cui et al. [27] applied the adaptive sliding mode control on a differential mobile robot for tracking different trajectories in the presence of uncertainties. Huang et al. [28] proposed an adaptive sliding mode control for a three wheel omnidirectional

**Fig. 2** Mecanum-wheeled mobile robot sideways movement



**Fig. 3** Conventional wheeled mobile robot curved movement



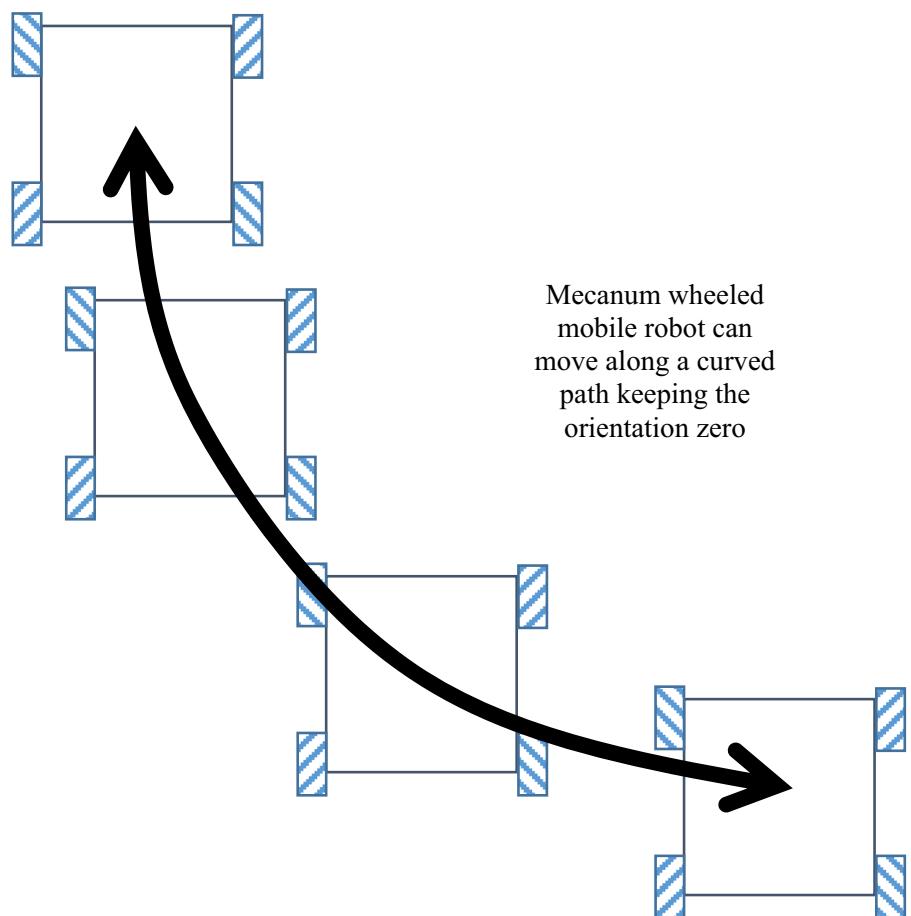
mobile. Recently, Wang et al. [29] further modified the adaptive laws for the estimation of bounded unknown uncertainties and online estimation of control gains.

Thus, motivated by the advantages claimed by previous work about second-order sliding mode control and adaptive laws, the current paper proposes an adaptive robust control to track the desired trajectory in presence of external force disturbance and uncertainties. A generalized equation of motion is derived using Newton–Euler method. Then, an adaptive robust control law is obtained to track the desired trajectory. In order to find the optimal value of few control variables of the control

law, pattern search optimization method has been used. Proposed control law efficacy is verified by presenting simulation results for two different trajectories. The results are compared with existing ARSMC and PID controller. Finally, the proposed algorithm is tested on a real robot by a new low-cost and simple methodology by using robot operating system (ROS) toolbox along with low-cost hardware and sensors.

This paper is organized in the following manner. In Sect. 2, the methodology proposed in this work is explained. Sections 3 and 4 provide kinematic and dynamic modeling of the mobile robot, respectively.

**Fig. 4** Mecanum-wheeled mobile robot curved movement



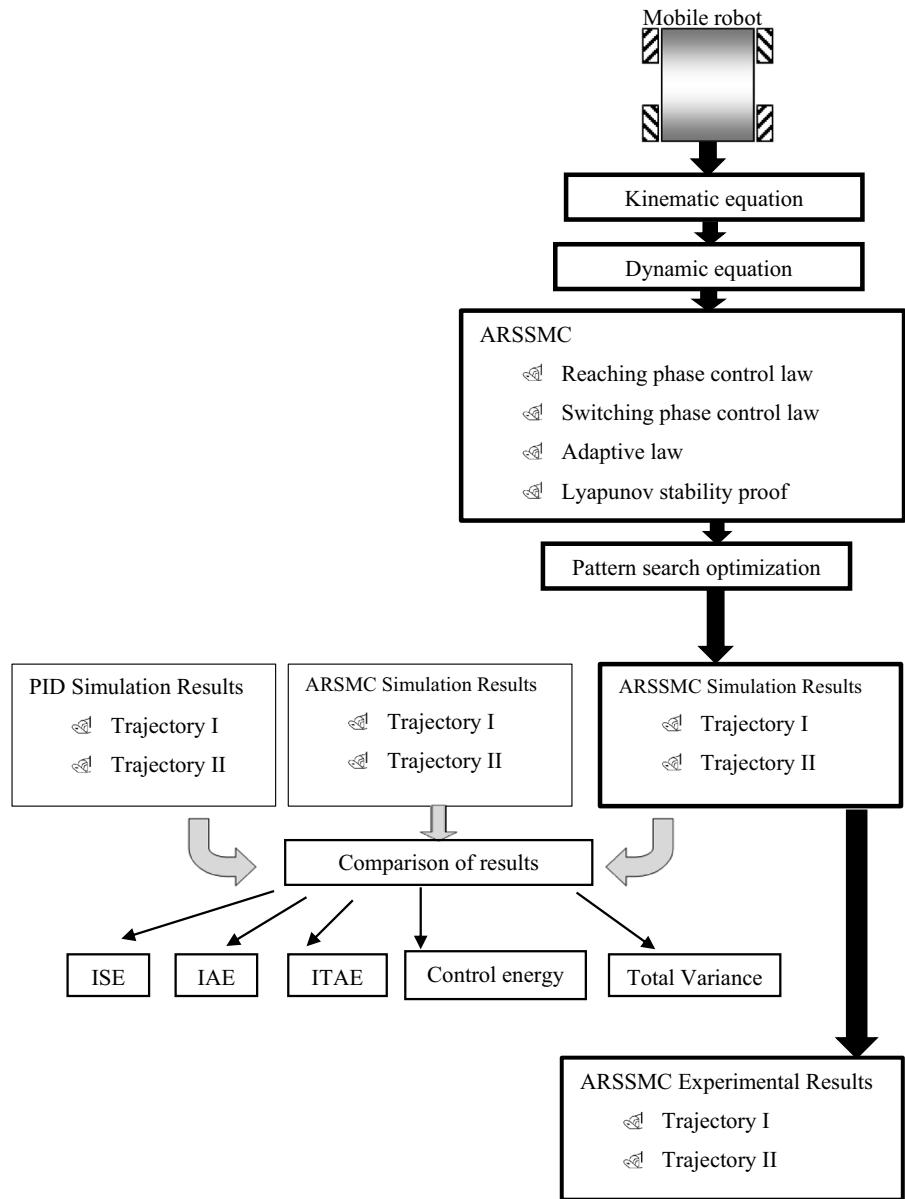
Section 5 presents the derivation of the proposed control law. The pattern search optimization algorithm is briefly explained in Sect. 6. Simulation and experimental results are presented in Sects. 7 and 8 respectively. Finally, conclusions drawn from this study are presented in Sect. 9.

## 2 Proposed methodology

Figure 5 shows a block diagram of the proposed methodology. Mobile robot with four Mecanum wheels is selected to track the desired trajectory in presence of uncertainties. Kinematic and dynamic analysis of the mobile robot is done to obtain the equation of motion using Newton–Euler approach assuming that the mobile robot moves on a plane horizontal surface. The equation of motion is utilized to derive the robust control law which is a combination of reach-

ing phase control law and switching phase control law. The robust controller is modified to adaptive robust controller by replacing the gains of switching function by an adaptive law. The stability of proposed control law is verified by Lyapunov stability theorem. Later, particle search optimization technique is utilized to determine optimal values of variables. Simulations are performed for two desired trajectories and the results obtained are compared with ARSMC and PID controller. To quantify the controller performance, integral square error (ISE), integral absolute error (IAE), and integral time-weighted absolute error (ITAE), control energy and variance of all the three controllers have been presented in a tabular form. Control energy accounts for the amount of energy required and total variance (TV) tells about the smoothness of the control signal. Finally, experiments are conducted to verify the results in a real and restricted environment.

**Fig. 5** Block diagram of proposed methodology



### 3 Kinematics

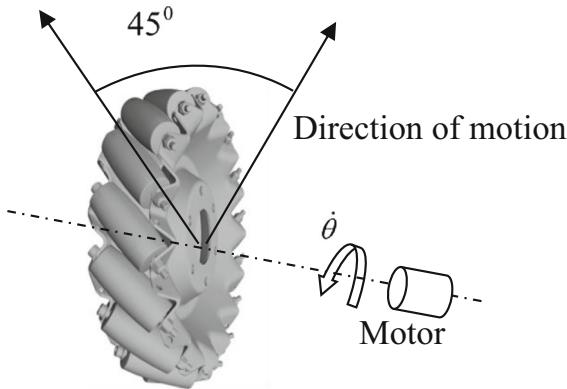
A four-wheel omnidirectional mobile robot consists of four Mecanum wheels (Fig. 6) in which peripheral rollers are inclined at a constant slope angle ( $\psi$ ). In this case  $\psi = 45^\circ$ , hence the wheel moves freely at an angle  $45^\circ$  with the driven motion. The torque required to drive the mobile robot is provided by the DC motor attached to each wheel. Point  $O_r$  (Fig. 7) is the center of gravity of the robot. It is assumed that the robot moves

over an even, horizontal and flat surface. Further, for the derivation of equation of motion, it is also assumed that all the components of robot including wheels are rigid. Figure 7 shows the schematic of a mobile robot with four Mecanum wheels.

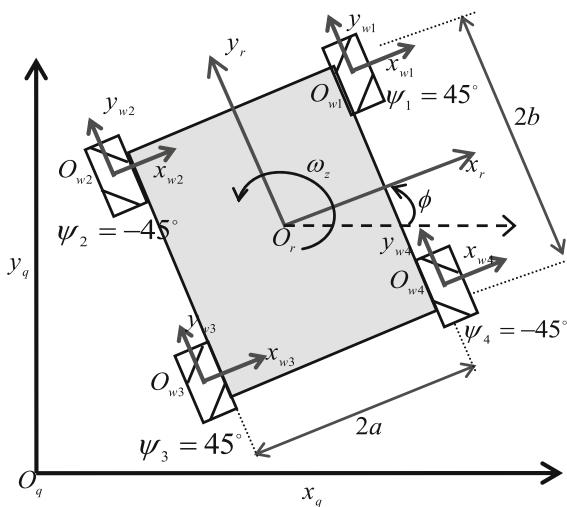
Following are the coordinate frames used in the kinematic modeling:

$O_q$  = Fixed coordinate frame.

$O_r$  = Mobile robot moving coordinate frame.



**Fig. 6** Mecanum wheel



**Fig. 7** Schematic of mobile robot

$O_{wi}$  ( $i = 1, 2, 3, 4$ ) = Wheel coordinate frame.

Let  $\mathbf{P}_{wi}$  ( $i = 1, 2, 3, 4$ ) =  $[x_{wi} \ y_{wi} \ \phi_{wi}]^T$  is the position vector of wheel in  $O_{wi}$ ,  $\dot{\theta}_{ix}$  ( $i = 1, 2, 3, 4$ ) = wheel angular velocity around the hub,  $\dot{\theta}_{ir}$  ( $i = 1, 2, 3, 4$ ) = angular velocity of roller,  $\dot{\theta}_{iz}$  ( $i = 1, 2, 3, 4$ ) = wheel angular velocity about the contact point,  $R_i$  ( $i = 1, 2, 3, 4$ ) = wheel radius,  $\psi_i$  ( $i = 1, 2, 3, 4$ ) = roller slope angle of each wheel, and velocity  $r$  = roller radius, then the robot velocity vector is given as

$$\dot{\mathbf{P}}_{wi} = \begin{bmatrix} \dot{x}_{wi} \\ \dot{y}_{wi} \\ \dot{\phi}_{wi} \end{bmatrix} = \begin{bmatrix} 0 & r \sin(\psi_i) & 0 \\ R_i & -r \cos(\psi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{ix} \\ \dot{\theta}_{ir} \\ \dot{\theta}_{iz} \end{bmatrix} \quad (1)$$

$$\text{Let, } \mathbf{T}_{wi}^r = \begin{bmatrix} \cos(\phi_{wi}^r) & -\sin(\phi_{wi}^r) & d_{wiy}^r \\ \sin(\phi_{wi}^r) & \cos(\phi_{wi}^r) & d_{wix}^r \\ 0 & 0 & 1 \end{bmatrix}$$

where  $\phi_{wi}^r$  ( $i = 1, 2, 3, 4$ ) is the rotational angle of

$O_{wi}$  with respect to  $O_r$ , and  $d_{wiy}^r$  and  $d_{wix}^r$  are the translational distance between two coordinate frames. If  $\mathbf{P}_r = [x_r \ y_r \ \phi_r]^T$  is the position vector of the robot in  $O_r$ , the relation between  $\mathbf{P}_r$  and  $\mathbf{P}_{wi}$  is obtained as  $\mathbf{P}_r = \mathbf{T}_{wi}^r \mathbf{P}_{wi}$ . Further, from Fig. 7 it can be seen that  $\phi_{w1}^r = \phi_{w2}^r = \phi_{w3}^r = \phi_{w4}^r = 0$ . Hence, robot velocity vector is written as

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\phi}_r \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_{wiy}^r \\ 0 & 1 & d_{wix}^r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_{wi} \\ \dot{y}_{wi} \\ \dot{\phi}_{wi} \end{bmatrix} \quad (2)$$

Using (1) and (2), we get

$$\dot{\mathbf{P}}_r = \mathbf{J}_i \dot{\mathbf{q}}_i \quad (3)$$

where  $\mathbf{J}_i \in R^{3 \times 3} = \begin{bmatrix} 0 & r \sin(\psi_i) & d_{wiy}^r \\ R_i & -r \cos(\psi_i) & d_{wix}^r \\ 0 & 0 & 1 \end{bmatrix}$  is  $i^{\text{th}}$  wheel Jacobian matrix and  $\dot{\mathbf{q}}_i = [\dot{\theta}_{ix} \ \dot{\theta}_{ir} \ \dot{\theta}_{iz}]$ .

**Remark 1** For  $\psi_i = 0$ ,  $|\mathbf{J}_i| = 0$ . Hence, singularity is not present in Mecanum wheels.

**Remark 2** Since, rank ( $\mathbf{J}_i$ ) = 3, therefore each wheel has three degrees of freedom (DOF).

Since, all the four wheels are identical, geometric and kinematic parameters for each wheel is taken as,  $R_1 = R_2 = R_3 = R_4 = R$ ,  $d_{w1x}^r = a$ ,  $d_{w1y}^r = b$ ,  $d_{w2x}^r = -a$ ,  $d_{w2y}^r = b$ ,  $d_{w3x}^r = -a$ ,  $d_{w3y}^r = -b$ ,  $d_{w4x}^r = a$ ,  $d_{w4y}^r = -b$ . Thus, Jacobian matrix for each wheel is obtained as

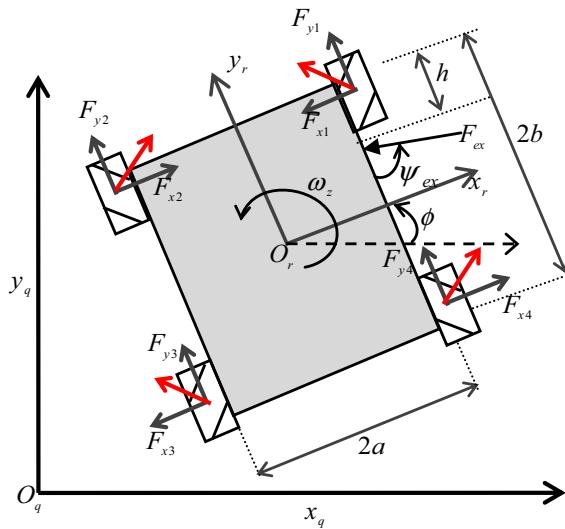
$$\mathbf{J}_1 = \begin{bmatrix} 0 & r/\sqrt{2} & b \\ R & -r/\sqrt{2} & a \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{J}_2 = \begin{bmatrix} 0 & -r/\sqrt{2} & b \\ R & -r/\sqrt{2} & -a \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{J}_3 = \begin{bmatrix} 0 & r/\sqrt{2} & -b \\ R & -r/\sqrt{2} & -a \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{J}_4 = \begin{bmatrix} 0 & -r/\sqrt{2} & -b \\ R & -r/\sqrt{2} & a \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Using (3) and (4), the inverse kinematics solution is obtained as [3]

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\phi}_r \end{bmatrix} = \frac{R}{4} \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 1/a + b & -1/a + b & -1/a + b & 1/a + b \end{bmatrix} \times \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} \quad (5)$$

where  $\dot{\theta}_i$  ( $i = 1, 2, 3, 4$ ) is the angular velocity of each wheel.



**Fig. 8** Free body diagram of mobile robot

**Remark 3** From the obtained kinematic solution, it can be observed that the mobile robot can move along any desired trajectory, even if  $\phi_r = 0$ . For example, if the robot has to move along a curved path, there is no need to change the orientation of the robot, which is one of the major reasons for its use in confined space applications.

Since, the closed-loop feedbacks are position and orientation, the velocity vector in world coordinate frame  $O_q$  is written as

$$\dot{\mathbf{P}}_q = [\dot{x}_q \dot{y}_q \dot{\phi}]^T = \mathbf{R}(\phi) \dot{\mathbf{P}}_r \quad (6)$$

where  $\mathbf{R}(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$  denotes the rotation matrix of  $O_r$  with respect to  $O_q$ .

#### 4 Dynamics

The dynamics of the mobile robot moving on a plane flat surface is derived using Newton–Euler method. For the purpose of derivation of equation of motion, it is assumed that moving coordinate frame  $O_r$  lies on the center of gravity of the robot. Let,  $\mathbf{M}_m$  is the mass matrix of the mobile robot,  $\mathbf{S}_q = [x_q \ y_q]^T$  is the position vector and  $\mathbf{F}_q = [F_{qx} \ F_{qy}]^T$  is the force vector in fixed coordinate frame  $O_q$ . Using Newton's second

law, equation of motion of the robot (Fig. 8) in coordinate frame  $O_q$  is written as

$$\mathbf{M}_m \ddot{\mathbf{S}}_q = \mathbf{F}_q \quad (7)$$

$$\begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \ddot{x}_q \\ \ddot{y}_q \end{bmatrix} = \begin{bmatrix} F_{qx} \\ F_{qy} \end{bmatrix} \quad (8)$$

If  ${}^q\mathbf{R}_r(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}$  is the transformation matrix of  $O_r$  with respect to  $O_q$ , then  $\dot{\mathbf{S}}_q = {}^q\mathbf{R}_r(\phi) \dot{\mathbf{S}}_r$  and  $\mathbf{F}_q = {}^q\mathbf{R}_r(\phi) \mathbf{F}_r$ . Using transformation matrix, (7) is simplified as,

$$\mathbf{M}_m ({}^q\mathbf{R}_r(\phi) \ddot{\mathbf{S}}_r + {}^q\mathbf{R}_r(\phi) \dot{\mathbf{S}}_r) = {}^q\mathbf{R}_r(\phi) \mathbf{F}_r \quad (9)$$

Multiplying both sides by  ${}^q\mathbf{R}_r(\phi)^{-1}$  yields

$$\begin{aligned} \mathbf{M}_m ({}^q\mathbf{R}_r(\phi)^{-1} {}^q\mathbf{R}_r(\phi) \ddot{\mathbf{S}}_r + {}^q\mathbf{R}_r(\phi)^{-1} {}^q\mathbf{R}_r(\phi) \dot{\mathbf{S}}_r) \\ = {}^q\mathbf{R}_r(\phi)^{-1} {}^q\mathbf{R}_r(\phi) \mathbf{F}_r \end{aligned} \quad (10)$$

Since,  ${}^q\mathbf{R}_r(\phi)^{-1} = {}^q\mathbf{R}_r(\phi)^T$ ,  ${}^q\mathbf{R}_r(\phi)^{-1} {}^q\mathbf{R}_r(\phi) (\dot{\phi}) = \dot{\phi} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ . Therefore,

$$\begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \ddot{x}_r - \dot{\phi} \dot{y}_r \\ \ddot{y}_r + \dot{\phi} \dot{x}_r \end{bmatrix} = \begin{bmatrix} F_{xr} \\ F_{yr} \end{bmatrix} - \begin{bmatrix} \beta_x \dot{x}_r \\ \beta_y \dot{y}_r \end{bmatrix} + \begin{bmatrix} -F_{ex} \cos(\psi_{ex}) \\ F_{ex} \sin(\psi_{ex}) \end{bmatrix} \quad (11)$$

where  $F_{xr}$  is the total force in x direction,  $F_{yr}$  is the total force in y direction,  $\beta_x$  is the linear friction coefficient in x direction and  $\beta_y$  is the linear friction coefficient in y direction.  $F_{ex}$  denotes the external force acting on the robot body at a distance  $h$  from the upper edge making an angle  $\psi_{ex}$  with  $y_r$ . Consider  $\tau$  as the moment about robot's c.g.,  $I_q$  as the moment of inertia of the robot about its c.g and  $\beta_z$  as the linear friction coefficient of friction in z direction. Thus, from the free body diagram (Fig. 8) Euler equation is written as

$$\begin{aligned} I_q \ddot{\phi} &= \tau - \beta_z \dot{\phi} + F_{ex} \cos(\psi_{ex}) a \\ &\quad - F_{ex} \sin(\psi_{ex})(b - h) \end{aligned} \quad (12)$$

Considering the dynamics of the DC motor attached to each wheel, the driving force  $F_{di}$  ( $i = 1, 2, 3, 4$ ), generated by the motor, is given as

$$F_{di}(i = 1, 2, 3, 4) = \alpha u_i - \beta R \dot{\theta}_i \quad (13)$$

where,  $u_i$  ( $i = 1, 2, 3, 4$ ) is the input voltage applied at each motors,  $\alpha$  and  $\beta$  are motor coefficients which are calculated using the following formulas

$$\alpha = \frac{k_\tau}{RR_a} \quad (14)$$

$$\beta = \frac{k_\tau k_e n}{R^2 R_a} \quad (15)$$

In (14) and (15),  $k_\tau$  denotes the motor torque coefficient,  $k_e$  is the motor back emf coefficient,  $n$  is the gear ratio, and  $R_a$  is the armature resistance.

From the free body diagram (Fig. 8),  $F_{xr}$ ,  $F_{yr}$  and  $\tau$  in terms of driving force is written as

$$F_{xr} = \frac{1}{2}(-F_{d1} + F_{d2} - F_{d3} + F_{d4}) \quad (16)$$

$$F_{yr} = \frac{1}{2}(F_{d1} + F_{d2} + F_{d3} + F_{d4}) \quad (17)$$

$$\begin{aligned} \tau &= \frac{a}{2}(F_{d1} - F_{d2} - F_{d3} + F_{d4}) \\ &\quad + \frac{b}{2}(F_{d1} - F_{d2} - F_{d3} + F_{d4}) \end{aligned} \quad (18)$$

With the states defined as  $\ddot{\mathbf{x}}(\mathbf{t}) = [\ddot{x}_q \ddot{y}_q \ddot{\phi}_q]^T$  and  $\mathbf{u}(\mathbf{t}) = [u_1 u_2 u_3 u_4]^T$ , and substituting (13) to (18) in (11) and (12), equation motion of the mobile robot is obtained in the form

$$\ddot{\mathbf{x}}(\mathbf{t})_{3 \times 1} = \mathbf{f}(\mathbf{x})_{3 \times 1} + \mathbf{g}(\mathbf{x})_{3 \times 4} \mathbf{u}(\mathbf{t})_{4 \times 1} + \xi(\mathbf{t}, \mathbf{u}(\mathbf{t}))_{3 \times 1} \quad (19)$$

where,

$$\begin{aligned} \mathbf{f}(\mathbf{x})_{3 \times 1} &= \left[ \begin{array}{l} (1/2M)(a_1 \dot{y}_q + a_2 \dot{\phi}_q + 2\beta_y \dot{x}_q + a_3 \dot{x}_q + a_4 + a_5 + 4\beta \dot{y}_q) \\ (1/2M)(b_1 \dot{x}_q + b_2 \dot{\phi}_q + 2\beta_x \dot{y}_q + b_3 \dot{y}_q + b_4 + b_5 + 4\beta \dot{\phi}_q) \\ (-1/2I_q)(c_1 \dot{\phi} + c_2 + c_3 + c_4) \end{array} \right], \\ \mathbf{g}(\mathbf{x})_{3 \times 4} &= \left[ \begin{array}{l} \bar{a}_{11} \bar{a}_{12} \bar{a}_{13} \bar{a}_{14} \\ \bar{b}_{11} \bar{b}_{12} \bar{b}_{13} \bar{b}_{14} \\ \bar{c}_{11} \bar{c}_{12} \bar{c}_{13} \bar{c}_{14} \end{array} \right] \text{ and } \xi(\mathbf{t}, \mathbf{u}(\mathbf{t}))_{3 \times 1} = [\xi_1 \xi_2 \xi_3]^T \end{aligned}$$

is the bounded lumped uncertainty. The bound limit of  $\xi(t, u(t))$  is not known in advance but satisfies  $|\xi(t, u(t))| \leq \xi_{\max}$ .

$\xi_{\max} \in \mathbb{R}^+$ . The variables used in the terms  $\mathbf{f}(\mathbf{x})_{3 \times 1}$  and  $\mathbf{g}(\mathbf{x})_{3 \times 4}$  are as follows:-

$$a_1 = \beta_x \sin(2\phi), a_2 = -\beta_y \sin(2\phi),$$

$$a_3 = 2 \cos^2(\phi)(\beta_x - \beta_y),$$

$$a_4 = 2F_{ex} \sin(\phi) \sin(\psi_{ex}),$$

$$a_5 = 2F_{ex} \cos(\phi) \cos(\psi_{ex}),$$

$$b_1 = -\beta_x \sin(2\phi), b_2 = \beta_y \sin(2\phi),$$

$$b_3 = 2 \cos^2(\phi)(\beta_x - \beta_y),$$

$$b_4 = -2F_{ex} \sin(\phi) \cos(\psi_{ex}),$$

$$b_5 = 2F_{ex} \cos(\phi) \sin(\psi_{ex}),$$

$$\bar{a}_{11} = \bar{a}_{13} = \frac{-\alpha}{2M}(\sin(\phi) + \cos(\phi)),$$

$$\bar{a}_{12} = \bar{a}_{14} = \frac{-\alpha}{2M}(\sin(\phi) - \cos(\phi)),$$

$$\begin{aligned} \bar{b}_{11} &= \bar{b}_{13} = \frac{-\alpha}{2M}(\sin(\phi) - \cos(\phi)), \\ \bar{b}_{12} &= \bar{b}_{14} = \frac{\alpha}{2M}(\sin(\phi) + \cos(\phi)), \text{ and} \\ \bar{c}_{11} &= -\bar{c}_{12} = -\bar{c}_{13} = \bar{c}_{14} \\ &= \frac{\alpha}{2I_q}(a + b). \end{aligned}$$

*Remark 4* From the derived equation of motion it is evident that it is a second-order nonlinear equation. The uncertainty  $\xi(\mathbf{t}, \mathbf{u}(\mathbf{t}))_{3 \times 1}$  is included in the equation to test the robustness of the controller in its presence. However, in real-life scenario with a changing environment, the robot can be hit by a moving obstacle which can make the robot unstable. Hence,  $F_{ex}$  is included in the equation of motion to account for the same. It is assumed that  $F_{ex}$  is measurable throughout the simulation.

Thus, to counter the nonlinearities associated with, bounded lumped uncertainties, and external force disturbance a robust adaptive higher-order sliding mode controller is developed in the next section.

## 5 Adaptive second-order sliding mode control design

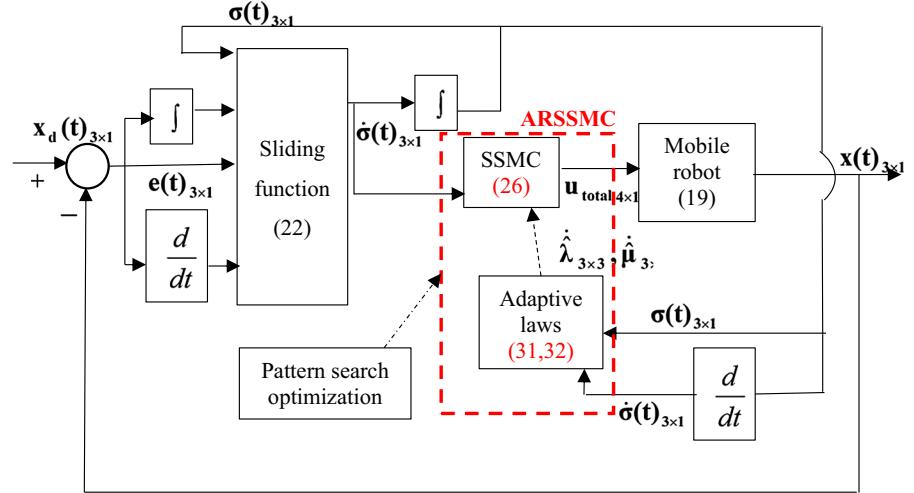
In this section (Fig. 9), a robust adaptive control design is proposed such that the mobile robot is able to track the desired trajectory in presence of uncertainties and external force. To reduce the complexity of the controller design, it is divided into three stages. First stage deals with the derivation of reaching phase control law. Then a switching control law is designed to deal with the bounded unknown uncertainties. Finally, to estimate the uncertainty bounds and switching law control gains, an adaptive law is proposed.

### 5.1 Reaching phase control law

In a conventional first-order sliding mode control, the aim is to force the state trajectories to move along a sliding surface  $\sigma(t) = 0$ . Compared to this, in a second-order sliding mode control, the purpose is not only to move the system states along the sliding surface  $\sigma(t) = 0$  but also its first-order derivative, i.e.,  $\dot{\sigma}(t) = 0$ .

Let the tracking error  $\mathbf{e}(\mathbf{t})_{3 \times 1} = [e_1 e_2 e_3]^T$  between the desired trajectory  $\mathbf{x}_d(\mathbf{t})_{3 \times 1} = [x_d y_d \phi_d]^T$

**Fig. 9** Mobile robot control system diagram



and obtained trajectory  $\mathbf{x}(t)_{3 \times 1} = [x_q \ y_q \ \phi]^T$  is defined as

$$\mathbf{e}(t)_{3 \times 1} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \\ \phi_d \end{bmatrix} - \begin{bmatrix} x_q \\ y_q \\ \phi \end{bmatrix} \quad (20)$$

It is assumed that initially the state trajectories are in the region  $\sigma(t) > 0$ . Conventionally, a sliding surface is a function of error  $e(t)$ . But as per the work of earlier researchers [29] it has been proved that selecting a sliding surface with proportional, integral and derivative action, gives flexibility to the controller to improve the performance in different scenarios. Hence, in order to improve the robustness of the system a PID-type sliding surface is selected which is chosen as

$$\dot{\sigma}(t)_{3 \times 1} = \mathbf{k}_{p_{3 \times 3}} \mathbf{e}(t)_{3 \times 1} + \mathbf{k}_{i_{3 \times 3}} \int \mathbf{e}(t)_{3 \times 1} dt + \mathbf{k}_{d_{3 \times 3}} \dot{\mathbf{e}}(t)_{3 \times 1} \quad (21)$$

$$\text{where, } \dot{\sigma}(t)_{3 \times 1} = \begin{bmatrix} \dot{\sigma}_1(t) \\ \dot{\sigma}_2(t) \\ \dot{\sigma}_3(t) \end{bmatrix}, \mathbf{k}_{p_{3 \times 3}} = \begin{bmatrix} k_{p1} & 0 & 0 \\ 0 & k_{p2} & 0 \\ 0 & 0 & k_{p3} \end{bmatrix}, \mathbf{k}_{i_{3 \times 3}} = \begin{bmatrix} k_{i1} & 0 & 0 \\ 0 & k_{i2} & 0 \\ 0 & 0 & k_{i3} \end{bmatrix}, \mathbf{k}_{d_{3 \times 3}} = \begin{bmatrix} k_{d1} & 0 & 0 \\ 0 & k_{d2} & 0 \\ 0 & 0 & k_{d3} \end{bmatrix}$$

positive control gains and  $\dot{\mathbf{e}}(t)_{3 \times 1} = [\dot{e}_1 \ \dot{e}_2 \ \dot{e}_3]^T$ . However, as the  $\sigma(t)_{3 \times 1}$  should also tend to zero, an additional term is included in (21) which will contribute for the damping of  $\sigma(t)_{3 \times 1}$ . The modified sliding surface is given as

$$\dot{\sigma}(t)_{3 \times 1} = \mathbf{k}_{p_{3 \times 3}} \mathbf{e}(t)_{3 \times 1} + \mathbf{k}_{i_{3 \times 3}} \int \mathbf{e}(t)_{3 \times 1} dt + \mathbf{k}_{d_{3 \times 3}} \dot{\mathbf{e}}(t)_{3 \times 1} - \kappa_{3 \times 3} \sigma(t)_{3 \times 1} \quad (22)$$

where,  $\kappa_{3 \times 3} = \begin{bmatrix} \kappa_1 & 0 & 0 \\ 0 & \kappa_2 & 0 \\ 0 & 0 & \kappa_3 \end{bmatrix}$  is positive constant. Differentiating (22) with respect to time, following equation is obtained

$$\ddot{\sigma}(t)_{3 \times 1} = \mathbf{k}_{p_{3 \times 3}} \dot{\mathbf{e}}(t)_{3 \times 1} + \mathbf{k}_{i_{3 \times 3}} \mathbf{e}(t)_{3 \times 1} + \mathbf{k}_{d_{3 \times 3}} \ddot{\mathbf{e}}(t)_{3 \times 1} - \kappa_{3 \times 3} \dot{\sigma}(t)_{3 \times 1} \quad (23)$$

Which can be further written as

$$\ddot{\sigma}(t)_{3 \times 1} = \mathbf{k}_{p_{3 \times 3}} \dot{\mathbf{e}}(t)_{3 \times 1} + \mathbf{k}_{i_{3 \times 3}} \mathbf{e}(t)_{3 \times 1} + \mathbf{k}_{d_{3 \times 3}} (\ddot{\mathbf{x}}_d(t)_{3 \times 1} - \ddot{\mathbf{x}}(t)_{3 \times 1}) - \kappa_{3 \times 3} \dot{\sigma}(t)_{3 \times 1} \quad (24)$$

Substituting for  $\ddot{\mathbf{x}}(t)_{3 \times 1}$  from equation (19) into equation (24), yields

$$\ddot{\sigma}(t)_{3 \times 1} = \mathbf{k}_{p_{3 \times 3}} \dot{\mathbf{e}}(t)_{3 \times 1} + \mathbf{k}_{i_{3 \times 3}} \mathbf{e}(t)_{3 \times 1} + \mathbf{k}_{d_{3 \times 3}} (\ddot{\mathbf{x}}_d(t)_{3 \times 1} - \mathbf{f}(\mathbf{x})_{3 \times 1} - \mathbf{g}(\mathbf{x})_{3 \times 4} \mathbf{u}(t)_{4 \times 1} - \xi(t, \mathbf{u}(t))_{3 \times 1}) - \kappa_{3 \times 3} \dot{\sigma}(t)_{3 \times 1} \quad (25)$$

As per ideal second-order sliding surface condition, the tracking error  $\mathbf{e}(t)_{3 \times 1}$  reaches to zero if  $\sigma(t)_{3 \times 1} = \dot{\sigma}(t)_{3 \times 1} = \ddot{\sigma}(t)_{3 \times 1} = 0$ . Hence, to obtain the reaching phase control law  $\mathbf{u}_{rp_{4 \times 1}}$  with  $\xi(t, \mathbf{u}(t))_{3 \times 1} = 0$ , the necessary condition is to make  $\ddot{\sigma}(t)_{3 \times 1} = 0$  which yields

$$\mathbf{u}_{rp_{4 \times 1}} = \mathbf{g}(\mathbf{x})_{4 \times 3}^{-1} \left( \frac{\mathbf{k}_{p_{3 \times 3}}}{\mathbf{k}_{d_{3 \times 3}}} \dot{\mathbf{e}}(t)_{3 \times 1} + \frac{\mathbf{k}_{i_{3 \times 3}}}{\mathbf{k}_{d_{3 \times 3}}} \mathbf{e}(t)_{3 \times 1} + \ddot{\mathbf{x}}_d(t)_{3 \times 1} - \mathbf{f}(\mathbf{x})_{3 \times 1} - \frac{\kappa_{3 \times 3}}{\mathbf{k}_{d_{3 \times 3}}} \dot{\sigma}(t)_{3 \times 1} \right) \quad (26)$$

If the uncertainties are considered, then the reaching phase control law  $\mathbf{u}_{rp_{4 \times 1}}$  is written as

$$\begin{aligned}\hat{\mathbf{u}}_{rp_{4 \times 1}} = & \mathbf{g}(\mathbf{x})_{4 \times 3}^{-1} \left( \frac{k_{p_{3 \times 3}}}{k_{d_{3 \times 3}}} \dot{\mathbf{e}}(\mathbf{t})_{3 \times 1} + \frac{k_{i_{3 \times 3}}}{k_{d_{3 \times 3}}} \mathbf{e}(\mathbf{t})_{3 \times 1} \right. \\ & + \ddot{\mathbf{X}}_{d_{3 \times 1}} - \mathbf{f}(\mathbf{x})_{3 \times 1} - \frac{k_{3 \times 3}}{k_{d_{3 \times 3}}} \dot{\sigma}(\mathbf{t})_{3 \times 1} \Big) \\ & + \mathbf{g}(\mathbf{x})_{4 \times 3}^{-1} (\xi(\mathbf{t}, \mathbf{u}(\mathbf{t}))_{3 \times 1})\end{aligned}\quad (27)$$

## 5.2 Switching control law

The reaching phase control law obtained in the last section is not effective if the mobile robot is moving in a real environment which is prone to uncertainties and hence its effect cannot be ignored. Therefore, to make the system robust against these disturbances, an additional switching control law  $\mathbf{u}_{sw_{4 \times 1}}$  is defined as

$$\begin{aligned}\mathbf{u}_{sw_{4 \times 1}} = & \mathbf{g}(\mathbf{x})_{4 \times 3}^{-1} (\lambda_{3 \times 3} \sigma(\mathbf{t})_{3 \times 1} \\ & + \mu_{3 \times 3} \text{sign}(\dot{\sigma}(\mathbf{t}))_{3 \times 1})\end{aligned}\quad (28)$$

where  $\lambda_{3 \times 3} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \in \Re^+$  with  $\lambda_{3 \times 3} > 0$ ,  $\mu_{3 \times 3} = \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{bmatrix} \in \Re^+$  with  $\mu_{3 \times 3} >$

$\xi(\mathbf{t}, \mathbf{u}(\mathbf{t}))_{\max}$  are the switching gains, and  $\text{sign}(\dot{\sigma}(\mathbf{t}))_{3 \times 1} = \begin{bmatrix} \text{sign}(\dot{\sigma}_1(t)) \\ \text{sign}(\dot{\sigma}_2(t)) \\ \text{sign}(\dot{\sigma}_3(t)) \end{bmatrix}$ . Thus, the total feedback control law for the mobile robot trajectory tracking is

$$\begin{aligned}\mathbf{u}_{\text{total}_{4 \times 1}} = & \mathbf{u}_{rp_{4 \times 1}} + \mathbf{u}_{sw_{4 \times 1}} \\ = & \mathbf{g}(\mathbf{x})_{4 \times 3}^{-1} \left( \frac{k_{p_{3 \times 3}}}{k_{d_{3 \times 3}}} \dot{\mathbf{e}}(\mathbf{t})_{3 \times 1} + \frac{k_{i_{3 \times 3}}}{k_{d_{3 \times 3}}} \mathbf{e}(\mathbf{t})_{3 \times 1} + \ddot{\mathbf{X}}_{d_{3 \times 1}} - \mathbf{f}(\mathbf{x})_{3 \times 1} \right. \\ & \left. - \frac{k_{3 \times 3}}{k_{d_{3 \times 3}}} \dot{\sigma}(\mathbf{t})_{3 \times 1} \right) \\ & + \mathbf{g}(\mathbf{x})_{4 \times 3}^{-1} (\lambda_{3 \times 3} \sigma(\mathbf{t})_{3 \times 1} + \mu_{3 \times 3} \text{sign}(\dot{\sigma}(\mathbf{t}))_{3 \times 1})\end{aligned}\quad (29)$$

Using (29) in (25) the second time derivative of sliding surface can be written as

$$\ddot{\sigma}(\mathbf{t})_{3 \times 1} = -k_{d_{3 \times 3}} \xi(\mathbf{t}, \mathbf{u}(\mathbf{t}))_{3 \times 1} - k_{d_{3 \times 3}} \lambda_{3 \times 3} \sigma(\mathbf{t})_{3 \times 1} - k_{d_{3 \times 3}} \mu_{3 \times 3} \text{sign}(\dot{\sigma}(\mathbf{t}))_{3 \times 1} \quad (30)$$

**Remark 5** The switching control law  $\mathbf{u}_{sw_{4 \times 1}}$  in equation (28) ensures that the state trajectories remains on the sliding surface even if the uncertainties exist. Moreover, the speed of convergence and chattering is governed by varying constants  $\lambda_{3 \times 3}$  and  $\mu_{3 \times 3}$ .

## 5.3 Adaptive control law

The mobile robot used in the work is a multiple-input–multiple-output (MIMO) system, due to which it is a tedious job to estimate the switching gain parameters compared to a single input single output system (SISO). Moreover, the environment in which it moves is uncertain where the uncertainties are bounded but their upper bounds are unknown. Now, if these parameters are selected by hit-and-trial method, it may increase the control voltage even when it is not required, which leads to chattering in the control input. Hence, an adaptive law is proposed to estimate the switching gain parameters  $\lambda_{3 \times 3}$  and  $\mu_{3 \times 3}$  which is given as

$$\begin{aligned}\dot{\hat{\lambda}}_{3 \times 3} = & \begin{bmatrix} \dot{\hat{\lambda}}_1 & 0 & 0 \\ 0 & \dot{\hat{\lambda}}_2 & 0 \\ 0 & 0 & \dot{\hat{\lambda}}_3 \end{bmatrix} \\ = & \begin{bmatrix} \rho_1 \sigma_1(t) \dot{\sigma}_1(t) & 0 & 0 \\ 0 & \rho_2 \sigma_2(t) \dot{\sigma}_2(t) & 0 \\ 0 & 0 & \rho_3 \sigma_3(t) \dot{\sigma}_3(t) \end{bmatrix}\end{aligned}\quad (31)$$

$$\begin{aligned}\dot{\hat{\mu}}_{3 \times 3} = & \begin{bmatrix} \dot{\hat{\mu}}_1 & 0 & 0 \\ 0 & \dot{\hat{\mu}}_2 & 0 \\ 0 & 0 & \dot{\hat{\mu}}_3 \end{bmatrix} \\ = & \begin{bmatrix} \gamma_1 |\dot{\sigma}_1(t)| & 0 & 0 \\ 0 & \gamma_2 |\dot{\sigma}_2(t)| & 0 \\ 0 & 0 & \gamma_3 |\dot{\sigma}_3(t)| \end{bmatrix}\end{aligned}\quad (32)$$

where  $\hat{\lambda}_{3 \times 3}$  and  $\hat{\mu}_{3 \times 3}$  are the estimates of  $\lambda_{3 \times 3}$  and  $\mu_{3 \times 3}$  respectively. The adaptation speed of  $\hat{\lambda}_{3 \times 3}$  and  $\hat{\mu}_{3 \times 3}$  are regulated by positive constants  $\rho_i$  ( $i = 1, 2, 3$ ) and  $\gamma_i$  ( $i = 1, 2, 3$ ). Therefore, modified adaptive robust control in presence of bounded, but uncertainties is written as

$$\begin{aligned}\mathbf{u}_{\text{total}_{4 \times 1}} = & \mathbf{u}_{rp_{4 \times 1}} + \mathbf{u}_{sw_{4 \times 1}} \\ = & \mathbf{g}(\mathbf{x})_{4 \times 3}^{-1} \\ & \times \left( \frac{k_{p_{3 \times 3}}}{k_{d_{3 \times 3}}} \dot{\mathbf{e}}(\mathbf{t})_{3 \times 1} + \frac{k_{i_{3 \times 3}}}{k_{d_{3 \times 3}}} \mathbf{e}(\mathbf{t})_{3 \times 1} + \ddot{\mathbf{X}}_{d_{3 \times 1}} - \mathbf{f}(\mathbf{x})_{3 \times 1} \right. \\ & \left. - \frac{k_{3 \times 3}}{k_{d_{3 \times 3}}} \dot{\sigma}(\mathbf{t})_{3 \times 1} \right) \\ & + \mathbf{g}(\mathbf{x})_{4 \times 3}^{-1} (\hat{\lambda}_{3 \times 3} \sigma(\mathbf{t})_{3 \times 1} + \hat{\mu}_{3 \times 3} \text{sign}(\dot{\sigma}(\mathbf{t}))_{3 \times 1})\end{aligned}\quad (33)$$

To prove the stability of the proposed control law, the theorem and its proof is as given below

**Theorem** Consider the mobile robot system subjected to uncertainties and external disturbance defined by

the equation of motion as in (19). The trajectory tracking error  $\mathbf{e}(\mathbf{t})_{3 \times 1}$  converges asymptotically to zero if the motors of the mobile robot follow the voltage control law as given in (33), provided the control design parameters  $\mathbf{k}_{\mathbf{p}_{3 \times 3}}, \mathbf{k}_{\mathbf{i}_{3 \times 3}}, \mathbf{k}_{\mathbf{d}_{3 \times 3}}, \rho_i (i = 1, 2, 3)$ , and  $\gamma_i (i = 1, 2, 3)$  are selected appropriately.

$$\text{Proof Let } \tilde{\lambda}_{3 \times 3} = \begin{bmatrix} \tilde{\lambda}_1 & 0 & 0 \\ 0 & \tilde{\lambda}_2 & 0 \\ 0 & 0 & \tilde{\lambda}_3 \end{bmatrix} = \begin{bmatrix} \hat{\lambda}_1 & 0 & 0 \\ 0 & \hat{\lambda}_2 & 0 \\ 0 & 0 & \hat{\lambda}_3 \end{bmatrix} -$$

$$\begin{bmatrix} \bar{\lambda}_1 & 0 & 0 \\ 0 & \bar{\lambda}_2 & 0 \\ 0 & 0 & \bar{\lambda}_3 \end{bmatrix} = \hat{\lambda}_{3 \times 3} - \bar{\lambda}_{3 \times 3} \text{ is the estimated error}$$

$$\text{where } \bar{\lambda}_{3 \times 3} \text{ is the nominal value of } \hat{\lambda}_{3 \times 3}. \text{ Similarly, } \tilde{\mu}_{3 \times 3} = \begin{bmatrix} \tilde{\mu}_1 & 0 & 0 \\ 0 & \tilde{\mu}_2 & 0 \\ 0 & 0 & \tilde{\mu}_3 \end{bmatrix} = \begin{bmatrix} \hat{\mu}_1 & 0 & 0 \\ 0 & \hat{\mu}_2 & 0 \\ 0 & 0 & \hat{\mu}_3 \end{bmatrix} -$$

$$\begin{bmatrix} \bar{\mu}_1 & 0 & 0 \\ 0 & \bar{\mu}_2 & 0 \\ 0 & 0 & \bar{\mu}_3 \end{bmatrix} = \hat{\mu}_{3 \times 3} - \bar{\mu}_{3 \times 3} \text{ is the estimated error}$$

where  $\bar{\mu}_{3 \times 3}$  is the nominal value of  $\hat{\mu}_{3 \times 3}$ . Consider Lyapunov function  $V$  as

$$V(t) = \frac{1}{2} \sigma(\mathbf{t})_{3 \times 1}^T \sigma(\mathbf{t})_{3 \times 1} + \frac{1}{2} \dot{\sigma}(\mathbf{t})_{3 \times 1}^T \dot{\sigma}(\mathbf{t})_{3 \times 1} + \sum_{i=1}^3 \frac{k_{di}}{2\rho_i} \tilde{\lambda}_i^2 + \sum_{i=1}^3 \frac{k_{di}}{2\gamma_i} \tilde{\mu}_i^2 \quad (34)$$

with  $V(t) = 0$  and  $V(t) > 0$  for  $\sigma(\mathbf{t})_{3 \times 1} \neq 0$  and  $\dot{\sigma}(\mathbf{t})_{3 \times 1} \neq 0$ , where  $\sigma(\mathbf{t})_{3 \times 1} = [\sigma_1(t) \ \sigma_2(t) \ \sigma_3(t)]^T$ . Differentiating (34) with respect to time yields

$$\dot{V}(t) = \sigma(\mathbf{t})_{3 \times 1}^T \dot{\sigma}(\mathbf{t})_{3 \times 1} + \dot{\sigma}(\mathbf{t})_{3 \times 1}^T \ddot{\sigma}(\mathbf{t})_{3 \times 1} + \sum_{i=1}^3 \frac{k_{di}}{\rho_i} \tilde{\lambda}_i \dot{\lambda}_i + \sum_{i=1}^3 \frac{k_{di}}{\gamma_i} \tilde{\mu}_i \dot{\mu}_i \quad (35)$$

$$\begin{aligned} &= \sigma(\mathbf{t})_{3 \times 1}^T \dot{\sigma}(\mathbf{t})_{3 \times 1} + \dot{\sigma}(\mathbf{t})_{3 \times 1}^T \mathbf{T}(-\mathbf{k}_{\mathbf{d}_{3 \times 3}} \xi(\mathbf{t}, \mathbf{u}(\mathbf{t}))_{3 \times 1} - \mathbf{k}_{\mathbf{d}_{3 \times 3}} \hat{\lambda}_{3 \times 3} \sigma(\mathbf{t})_{3 \times 1} \\ &\quad - \mathbf{k}_{\mathbf{d}_{3 \times 3}} \hat{\mu}_{3 \times 3} \text{sign}(\dot{\sigma}(\mathbf{t}))_{3 \times 1}) + \sum_{i=1}^3 \frac{k_{di}}{\rho_i} \tilde{\lambda}_i \dot{\lambda}_i \\ &\quad + \sum_{i=1}^3 \frac{k_{di}}{\gamma_i} \tilde{\mu}_i \dot{\mu}_i \end{aligned}$$

$$\begin{aligned} &= \sigma(\mathbf{t})_{3 \times 1}^T \dot{\sigma}(\mathbf{t})_{3 \times 1} - \dot{\sigma}(\mathbf{t})_{3 \times 1}^T \mathbf{k}_{\mathbf{d}_{3 \times 3}} \xi(\mathbf{t}, \mathbf{u}(\mathbf{t}))_{3 \times 1} \\ &\quad - \dot{\sigma}(\mathbf{t})_{3 \times 1}^T \mathbf{k}_{\mathbf{d}_{3 \times 3}} \hat{\lambda}_{3 \times 3} \sigma(\mathbf{t})_{3 \times 1} \\ &\quad - \dot{\sigma}(\mathbf{t})_{3 \times 1}^T \mathbf{k}_{\mathbf{d}_{3 \times 3}} \hat{\mu}_{3 \times 3} \text{sign}(\dot{\sigma}(\mathbf{t}))_{3 \times 1} + \sum_{i=1}^3 \frac{k_{di}}{\rho_i} \tilde{\lambda}_i \dot{\lambda}_i \end{aligned}$$

$$\begin{aligned} &+ \sum_{i=1}^3 \frac{k_{di}}{\gamma_i} \tilde{\mu}_i \dot{\mu}_i \\ &= \sum_{i=1}^3 \sigma_i(t) \dot{\sigma}_i(t) - \sum_{i=1}^3 \dot{\sigma}_i(t) k_{di} \xi_i - \sum_{i=1}^3 \dot{\sigma}_i(t) k_{di} \hat{\lambda}_i \sigma_i(t) \\ &\quad - \sum_{i=1}^3 \dot{\sigma}_i(t) k_{di} \hat{\mu}_i \text{sign}(\dot{\sigma}_i(t)) \\ &\quad + \sum_{i=1}^3 \frac{k_{di}}{\rho_i} \tilde{\lambda}_i \dot{\lambda}_i + \sum_{i=1}^3 \frac{k_{di}}{\gamma_i} \tilde{\mu}_i \dot{\mu}_i \\ &= \sum_{i=1}^3 \sigma_i(t) \dot{\sigma}_i(t) - \sum_{i=1}^3 k_{di} \xi_i \dot{\sigma}_i(t) - \sum_{i=1}^3 k_{di} \hat{\lambda}_i \sigma_i(t) \dot{\sigma}_i(t) \\ &\quad - \sum_{i=1}^3 k_{di} \hat{\mu}_i \dot{\sigma}_i(t) \text{sign}(\dot{\sigma}_i(t)) + \sum_{i=1}^3 \frac{k_{di}}{\rho_i} \tilde{\lambda}_i \rho_i \sigma_i(t) \dot{\sigma}_i(t) \\ &\quad + \sum_{i=1}^3 \frac{k_{di}}{\gamma_i} \tilde{\mu}_i \gamma_i |\dot{\sigma}_i(t)| = \sum_{i=1}^3 \sigma_i(t) \dot{\sigma}_i(t) \\ &\quad - \sum_{i=1}^3 k_{di} \xi_i \dot{\sigma}_i(t) - \sum_{i=1}^3 k_{di} \hat{\lambda}_i \sigma_i(t) \dot{\sigma}_i(t) - \sum_{i=1}^3 k_{di} \hat{\mu}_i |\dot{\sigma}_i(t)| \\ &\quad + \sum_{i=1}^3 k_{di} \tilde{\lambda}_i \sigma_i(t) \dot{\sigma}_i(t) + \sum_{i=1}^3 k_{di} \tilde{\mu}_i \gamma_i |\dot{\sigma}_i(t)| \\ &= \sum_{i=1}^3 \sigma_i(t) \dot{\sigma}_i(t) - \sum_{i=1}^3 k_{di} \xi_i \dot{\sigma}_i(t) - \sum_{i=1}^3 k_{di} \hat{\lambda}_i \sigma_i(t) \dot{\sigma}_i(t) \\ &\quad - \sum_{i=1}^3 k_{di} \hat{\mu}_i |\dot{\sigma}_i(t)| + \sum_{i=1}^3 k_{di} \hat{\lambda}_i \sigma_i(t) \dot{\sigma}_i(t) \\ &\quad - \sum_{i=1}^3 k_{di} \tilde{\lambda}_i \sigma_i(t) \dot{\sigma}_i(t) \\ &\quad + \sum_{i=1}^3 k_{di} \hat{\mu}_i \gamma_i |\dot{\sigma}_i(t)| - \sum_{i=1}^3 k_{di} \tilde{\mu}_i \gamma_i |\dot{\sigma}_i(t)| \\ &= \sum_{i=1}^3 \sigma_i(t) \dot{\sigma}_i(t) - \sum_{i=1}^3 k_{di} \xi_i \dot{\sigma}_i(t) - \sum_{i=1}^3 k_{di} \tilde{\lambda}_i \sigma_i(t) \dot{\sigma}_i(t) \\ &\quad - \sum_{i=1}^3 k_{di} \tilde{\mu}_i \gamma_i |\dot{\sigma}_i(t)| \\ &\leq \sum_{i=1}^3 |\dot{\sigma}_i(t)| (\sigma_i(t) - k_{di} \xi_i - k_{di} \tilde{\lambda}_i \sigma_i(t) - k_{di} \tilde{\mu}_i \gamma_i) \\ &\leq \sum_{i=1}^3 |\dot{\sigma}_i(t)| (|\sigma_i(t)| - k_{di} \xi_i - k_{di} \tilde{\lambda}_i |\sigma_i(t)| - k_{di} \tilde{\mu}_i \gamma_i) \\ &\leq \sum_{i=1}^3 |\dot{\sigma}_i(t)| (|\sigma_i(t)| + k_{di} \xi_{\max_i} - k_{di} \tilde{\lambda}_i |\sigma_i(t)| - k_{di} \tilde{\mu}_i \gamma_i) \\ &\leq \sum_{i=1}^3 |\dot{\sigma}_i(t)| (|\sigma_i(t)| (1 - k_{di} \tilde{\lambda}_i) + k_{di} (\xi_{\max_i} - \tilde{\mu}_i \gamma_i)) \end{aligned}$$

$$\leq - \sum_{i=1}^3 |\dot{\sigma}_i(t)| (\|\sigma_i(t)\| (k_{di}\bar{\lambda}_i - 1) + k_{di} (\bar{\mu}_i\gamma_i - \xi_{\max_i})) \quad (36)$$

The adaptive law given by (31) and (32) ensures that  $\bar{\lambda}_i > \frac{1}{k_{di}}$  and  $\bar{\mu}_i > \frac{\xi_{\max_i}}{\gamma_i}$ . Thus from (36), asymptotic stability of the system under uncertainties is guaranteed as the derivative of the Lyapunov function is a negative definite [30].  $\square$

*Remark 6* The use of signum function in equation (33) increases the chattering effect if the switching gain increases. Therefore, the proposed control law is further modified using boundary layer approach [30] to reduce the chattering effect as

$$\begin{aligned} \mathbf{u}_{\text{total}4 \times 1} &= \mathbf{u}_{\mathbf{rp}4 \times 1} + \mathbf{u}_{\mathbf{sw}4 \times 1} \\ &= \mathbf{g}(\mathbf{x})_{4 \times 3}^{-1} \left( \begin{array}{l} \frac{\mathbf{k}_{p3 \times 3}}{\mathbf{k}_{d3 \times 3}} \dot{\mathbf{e}}(\mathbf{t})_{3 \times 1} + \frac{\mathbf{k}_{i3 \times 3}}{\mathbf{k}_{d3 \times 3}} \mathbf{e}(\mathbf{t})_{3 \times 1} + \ddot{\mathbf{x}}_{3 \times 1} - \mathbf{f}(\mathbf{x})_{3 \times 1} \\ - \frac{\mathbf{k}_{i3 \times 3}}{\mathbf{k}_{d3 \times 3}} \dot{\sigma}(\mathbf{t})_{3 \times 1} \\ + \mathbf{g}(\mathbf{x})_{4 \times 3}^{-1} (\bar{\lambda}_{3 \times 3} \sigma(\mathbf{t})_{3 \times 1} + \hat{\mu}_{3 \times 3} \text{sat}(\dot{\sigma}(\mathbf{t})_{3 \times 1})) \end{array} \right) \end{aligned} \quad (37)$$

where,  $\text{sat}(\dot{\sigma}(\mathbf{t}))_{3 \times 1} = \begin{cases} \text{sign}(\dot{\sigma}(\mathbf{t}))_{3 \times 1}, & |(\dot{\sigma}(\mathbf{t}))_{3 \times 1}| > \delta > 0 \\ \frac{(\dot{\sigma}(\mathbf{t}))_{3 \times 1}}{\delta}, & |(\dot{\sigma}(\mathbf{t}))_{3 \times 1}| \leq \delta \end{cases}$ , and  $\delta$  is a small positive constant.

## 6 Pattern search optimization algorithm

This section provides the pattern search optimization algorithm which is utilized for selecting the optimal values of  $\mathbf{k}_{p3 \times 3}$ ,  $\mathbf{k}_{i3 \times 3}$ ,  $\mathbf{k}_{d3 \times 3}$ ,  $\mathbf{k}_{3 \times 3}$ ,  $\rho_i (i = 1, 2, 3)$ ,  $\gamma_i (i = 1, 2, 3)$  and  $\delta$ . Let  $\mathbf{x}_i$  be the initial solution of  $f(\mathbf{x}_i)$ . Then, for  $f(\mathbf{x}^+) < f(\mathbf{x}_i)$ ,  $\mathbf{x}_i$  is replaced by a new solution  $\mathbf{x}^+$  and with each iteration set of solutions given by a mesh  $M_s$  gets updated. The subset solution of  $M_s$  which is in direct neighborhood of  $\mathbf{x}_i$  is defined as pattern vector  $P_s$ . The algorithm for pattern search optimization are as follows [31]:

- Define objective function.
- Start with an initial guess for  $\mathbf{x}_i$ .
- Search for a new solution  $\mathbf{x}^+$  in  $P_s$  such that  $f(\mathbf{x}^+) < f(\mathbf{x}_i)$ .
- If previous step is successful then start the next iteration with  $\mathbf{x}_{i+1} = \mathbf{x}^+$  or else start the next iteration by contracting the mesh size and setting  $\mathbf{x}_{i+1} = \mathbf{x}_i$ .
- Check the stopping condition.
- If stopping condition is satisfied stop the algorithm or else repeat from third step.

**Table 1** Parametric values

Parameter	Symbol	Unit	Value
Mass of mobile robot	$M$	kg	6
Moment of inertia of mobile robot	$I_q$	kg m <sup>2</sup>	0.0945
Radius of wheel	$R$	m	0.05
Length of platform	$2a$	m	0.22
Breadth of platform	$2b$	m	0.36
Coefficient of friction	$\beta_x = \beta_y = \beta_z$	—	0.02
Motor coefficient	$\alpha$	N/V	0.087
Motor coefficient	$\beta$	kg/s	11.4

In this work, tracking norm error is defined as the objective function and 1000 iterations were performed to obtain the optimal values of  $\mathbf{k}_{p3 \times 3}$ ,  $\mathbf{k}_{i3 \times 3}$ ,  $\mathbf{k}_{d3 \times 3}$ ,  $\mathbf{k}_{3 \times 3}$ ,  $\rho_i (i = 1, 2, 3)$ ,  $\gamma_i (i = 1, 2, 3)$  and  $\delta$ .

## 7 Simulation results and discussion

To demonstrate and compare the effectiveness of the proposed robust adaptive control law given by (37), computer simulations have been performed and the results obtained are compared with ARSMC and PID control. For this purpose, two different trajectories have been chosen in a challenging environment where the uncertainties and external forces are present. Trajectory I is an eight trajectory with the constraint that the mobile robot should track the desired path without changing the orientation, i.e  $\phi = 0$ . Trajectory II is a closed rectangular path with changing orientation at the corners. To carry out the simulations, parametric values of the mobile robot are presented in Table 1.

### 7.1 Trajectory I: eight trajectory

The equation of trajectory I is given as

$$\begin{cases} x = \frac{0.3 \cos(t)}{1 + \sin^2(t)} \\ y = \frac{0.4 \sin(t) \cos(t)}{1 + \sin^2(t)} \quad \forall t \geq 0 \\ \phi = 0 \end{cases} \quad (38)$$

where  $t$  is the simulation time in seconds. The initial posture of the mobile robot is

$$[x_q \ y_q \ \phi]^T = [0.3 \ 0 \ 0]^T.$$

As the proposed controller performance has to be tested in an uncertain environment, the external force  $F_{\text{ex}}$  and bounded uncertainties  $\xi(t, \mathbf{u}(t))_{3 \times 1}$  are taken as

$$F_{\text{ex}} = 1.5 \quad \forall 7 < t \leq 10.$$

$$\xi(t, \mathbf{u}(t))_{3 \times 1} = \begin{cases} 0.5 & \forall 20 < t \leq 26 \\ 0.5 \sin(2t) & \forall 40 \leq t \\ 0.5 \sin(2t) & \forall 60 \leq t \end{cases}$$

Proper selection of design parameters of the controller is the major challenge faced during simulations. Hence, extensive computational simulations are performed to decide a range of each design parameters for minimizing the error with minimum control efforts. Further, the pattern search optimization algorithm is utilized to generate the optimized value of each parameter. Following are the observations noted while selecting the parameters:

- Sliding surface parameters  $\mathbf{k}_{\mathbf{p}_{3 \times 3}}$ ,  $\mathbf{k}_{\mathbf{i}_{3 \times 3}}$  and  $\mathbf{k}_{\mathbf{d}_{3 \times 3}}$  plays a major role in satisfying the ideal second-order sliding mode condition ( $\sigma(t)_{3 \times 1} = \dot{\sigma}(t)_{3 \times 1} = \ddot{\sigma}(t)_{3 \times 1} = 0$ ). It should be noted that  $\mathbf{k}_{\mathbf{d}_{3 \times 3}}$  value should be limited to avoid amplification of high frequency noise signals. Hence, a range was selected for each parameter and the optimal values were calculated using optimization method. The val-

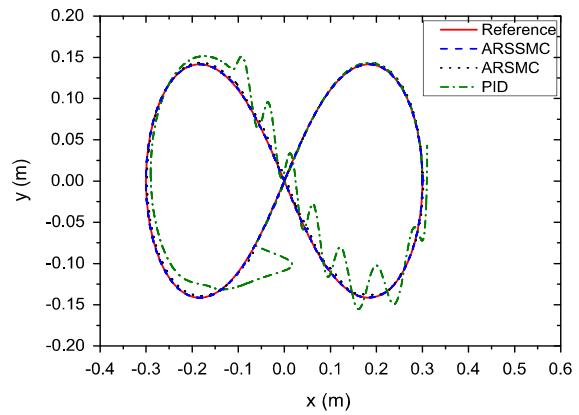
$$\text{ues are } \mathbf{k}_{\mathbf{p}_{3 \times 3}} = \begin{bmatrix} 121.4 & 0 & 0 \\ 0 & 134.7 & 0 \\ 0 & 0 & 56 \end{bmatrix}, \mathbf{k}_{\mathbf{i}_{3 \times 3}} = \begin{bmatrix} 41.05 & 0 & 0 \\ 0 & 36.7 & 0 \\ 0 & 0 & 12.1 \end{bmatrix}, \mathbf{k}_{\mathbf{d}_{3 \times 3}} = \begin{bmatrix} 23.4 & 0 & 0 \\ 0 & 18.6 & 0 \\ 0 & 0 & 6.74 \end{bmatrix}.$$

It should be noted that in the presence of bounded uncertainties these parameters are unable to reduce the tracking error.

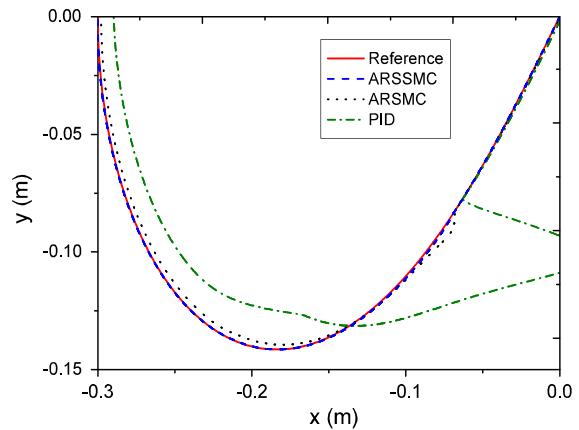
- The parameter  $\kappa_{3 \times 3}$  is selected such that it converges the sliding function to zero, but at the same time higher value should be avoided as to prevent saturation of control input. Accordingly, values are

$$\kappa_{3 \times 3} = \begin{bmatrix} 1.8 & 0 & 0 \\ 0 & 0.67 & 0 \\ 0 & 0 & .13 \end{bmatrix}.$$

- To improve the tracking performance of the controller in the presence of uncertainties, it is important to properly select the constants  $\rho_i (i = 1, 2, 3)$  and  $\gamma_i (i = 1, 2, 3)$ . Increasing these constants improves the adaptation speed and converges the sliding function to zero in the switching phase; however, increasing it beyond a limit can lead



**Fig. 10** Trajectory I in x-y plane

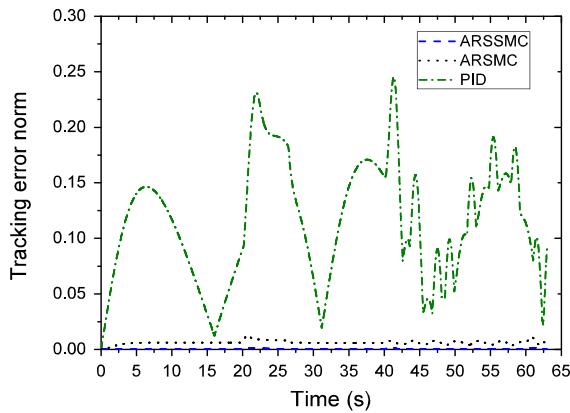


**Fig. 11** Trajectory I in x-y plane ( $20 < t \leq 26$ )

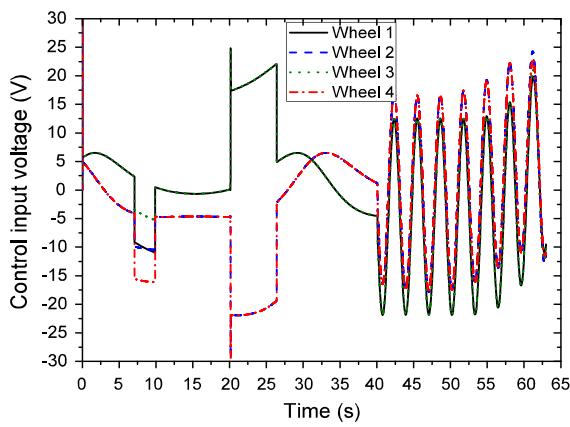
to undesirable control energy. The values of constants obtained are  $\rho_1 = 24.5$ ,  $\rho_2 = 22.7$ ,  $\rho_3 = 15.5$ ,  $\gamma_1 = 31.6$ ,  $\gamma_2 = 17.91$ , and  $\gamma_3 = 16.21$ .

- Increasing the value of  $\delta$  makes the control torque smooth, but very higher value overshoots the sliding surface. Hence, to satisfy the design requirements is selected as 0.1.

The tracking result for trajectory I with ARSSMC, ARSMC and PID controller is shown in Fig. 10. It is evident from the obtained results that in the absence of uncertainties, ARSSMC and ARSMC have similar tracking capability. However, as the uncertainties are fed after 20 s, PID controller is unable to track the trajectory, whereas, ARSMC performs much better (Fig. 11). It should be noted that even though the ARSMC converges the error to zero in minimum time caused by  $\xi_1$ , the mobile robot again deviates from its desired path as  $\xi_2$  and  $\xi_3$  are introduced at different time



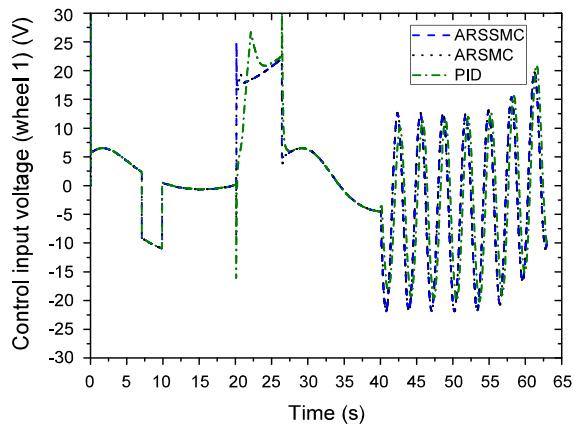
**Fig. 12** Tracking error norm versus time plot for trajectory I



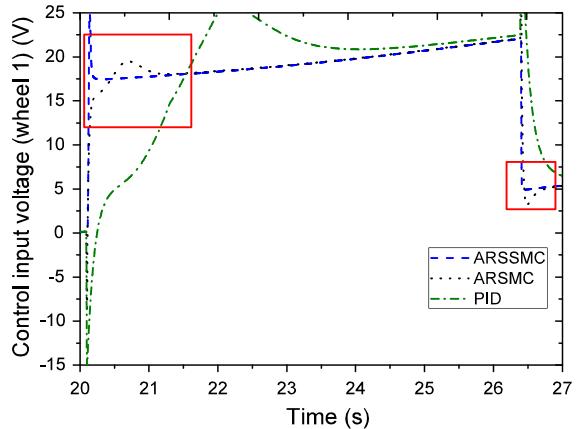
**Fig. 13** Control effort of ARSSMC for trajectory I

intervals. This change can be quantified from Fig. 12, which shows the change of tracking error norm with respect to time. It can be seen that PID controller tracking norm is unable to converge to zero, ARSMC tracking norm increases as the uncertainties are introduced, whereas proposed ARSSMC tracking error norm converges to zero in minimum time.

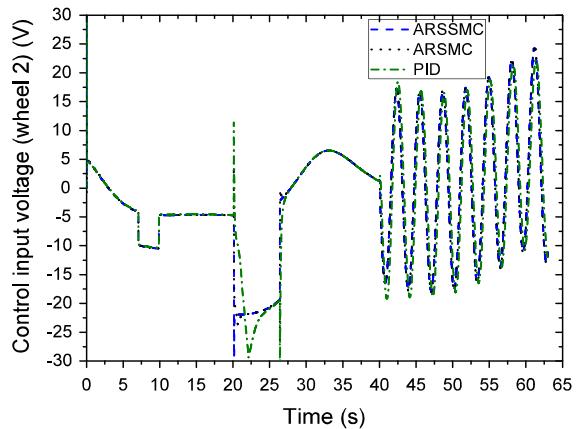
Figure 13 shows the control input voltage at each motor of the mobile robot to track the desired trajectory. The voltages of motor changes as per desired trajectory. The sudden changes in the control voltages for  $7 < t \leq 10$  and  $20 < t \leq 26$  is subjected to inclusion of  $F_{ex}$  and  $\xi_1$ . To give an insight into the control input voltage obtained by ARSSMC, ARSMC and PID controllers, comparison plots for wheel 1 and wheel 2 have been shown (Figs. 14, 15, 16, 17). As can be seen from Fig. 15, the control input voltage of wheel 1 at 20 s increases and settles down in a very less time.



**Fig. 14** Comparison of control effort of wheel 1

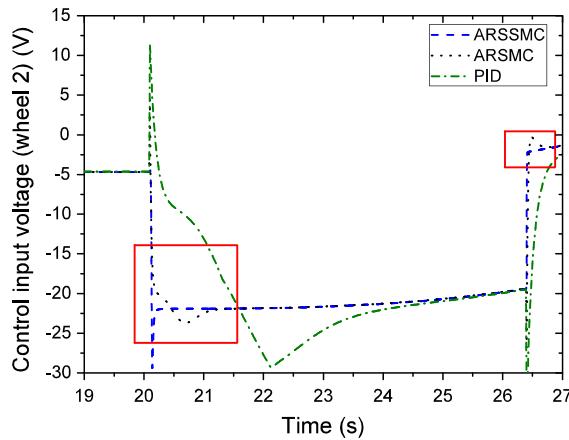


**Fig. 15** Comparison of control effort of wheel 1 ( $20 < t \leq 26$ )

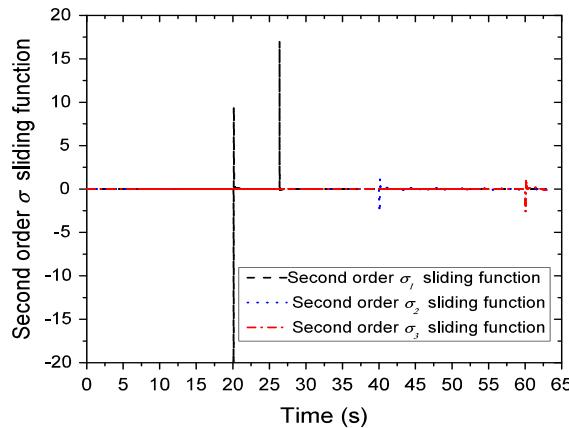


**Fig. 16** Comparison of control effort of wheel 2

Thus, responds very fast to the fed uncertainties. In comparison to this, ARSMC takes approximately 1 s



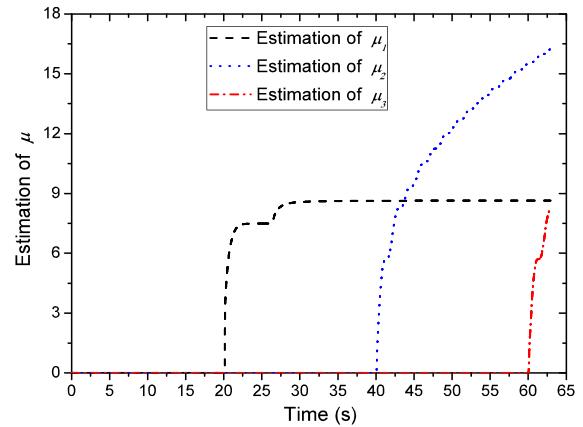
**Fig. 17** Comparison of control effort of wheel 2 ( $20 < t \leq 26$ )



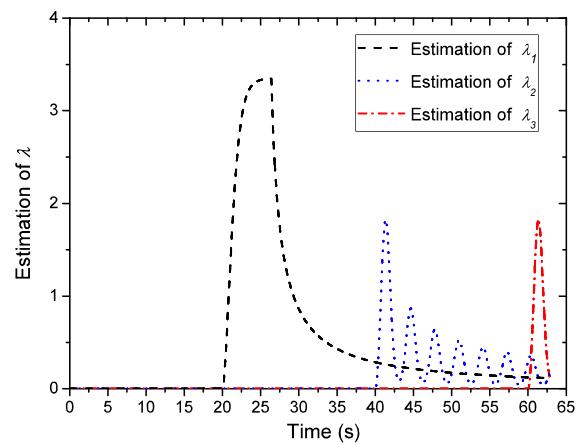
**Fig. 18** Second derivative of sliding function  $\sigma_1(t)$ ,  $\sigma_2(t)$  and  $\sigma_3(t)$

to settle down, which results the mobile robot to deviate from its desired trajectory. Moreover, when the  $\xi_1$  becomes zero at 26 s, the control voltages obtained by ARSSMC changes smoothly as compared to ARSMC and PID controller. It can be seen from Fig. 18 that at each instant when the uncertainties are fed, the proposed control law is able to drive the second-order sliding function  $\ddot{\sigma}(t)_{3 \times 1}$  to zero at a very fast rate of convergence. Thus, the closed-loop stability condition is satisfied. Figures 19 and 20 shows the online estimation of  $\mu_1$ ,  $\mu_2$ ,  $\mu_3$ , and  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$  respectively.

In order to evaluate the tracking performance of ARSSMC, ARSMC and PID control, integral square error (ISE), integral absolute error (IAE), and integral time-weighted absolute error (ITAE) have been calculated. The obtained values are for the full simulation



**Fig. 19** Estimation of  $\mu_1$ ,  $\mu_2$  and  $\mu_3$



**Fig. 20** Estimation of  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$

**Table 2** Tracking performance comparison (Trajectory I)

Controller	ISE	IAE	ITAE
ARSSMC	$10^{-6} \begin{bmatrix} 6.06 \\ 3.64 \\ 2.18 \end{bmatrix}$	$10^{-3} \begin{bmatrix} 9.7 \\ 7.11 \\ 2.11 \end{bmatrix}$	$\begin{bmatrix} 0.28 \\ 0.33 \\ 0.13 \end{bmatrix}$
ARSMC	$10^{-4} \begin{bmatrix} 3.8 \\ 2.8 \\ 2.9 \end{bmatrix}$	$\begin{bmatrix} 0.14 \\ 0.123 \\ 0.126 \end{bmatrix}$	$\begin{bmatrix} 4.375 \\ 4.1 \\ 4.24 \end{bmatrix}$
PID	$10^{-2} \begin{bmatrix} 3.9 \\ 1.4 \\ 2.7 \end{bmatrix}$	$\begin{bmatrix} 0.88 \\ 0.71 \\ 0.87 \end{bmatrix}$	$\begin{bmatrix} 27.8 \\ 30.01 \\ 37.96 \end{bmatrix}$

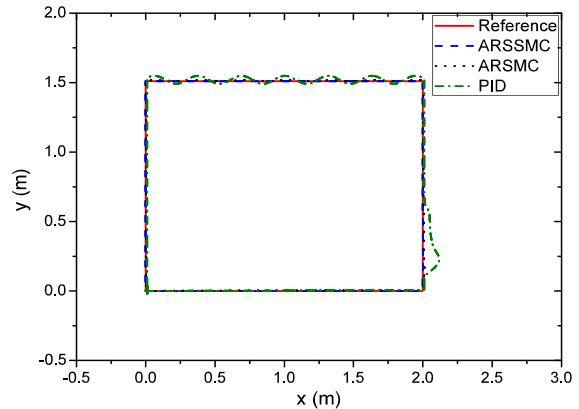
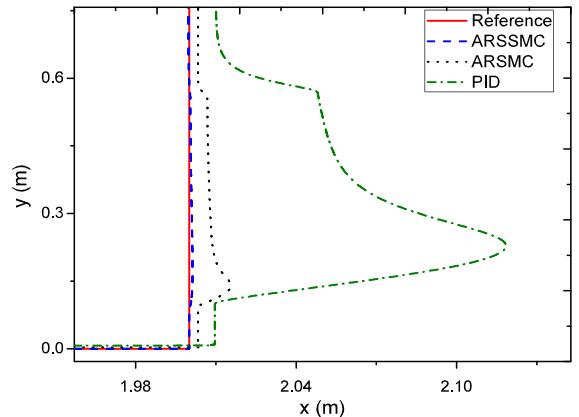
run time (63 s) in the presence of uncertainties and external forces. The tracking performance of each controller has been tabulated in Table 2. It can be seen from Table 2 that from ISE, IAE and ITAE point of view, ARSSMC provides a better tracking performance as compared to ARSMC and PID controller. Moreover,

**Table 3** Controller performance comparison (Trajectory I)

Controller	Control energy at each wheel ( $10^3$ ) (Nm)				Total variance of control input			
	Wheel 1	Wheel 2	Wheel 3	Wheel 4	Wheel 1	Wheel 2	Wheel 3	Wheel 4
ARSSMC	2.97	3.08	2.94	3.094	139.6	141.95	136.3	146.2
ARSMC	12	12.96	11.7	12.4	157.5	162.87	154.7	16.9

**Table 4** Computational time and memory consumption (Trajectory I)

Controller	Computational time (s)	Memory consumption (MB)
ARSSMC	9.4	234.5 ( $\pm 2.5$ )
ARSMC	5.62	188.1 ( $\pm 1.6$ )
PID	3.1	162.8 ( $\pm 0.7$ )

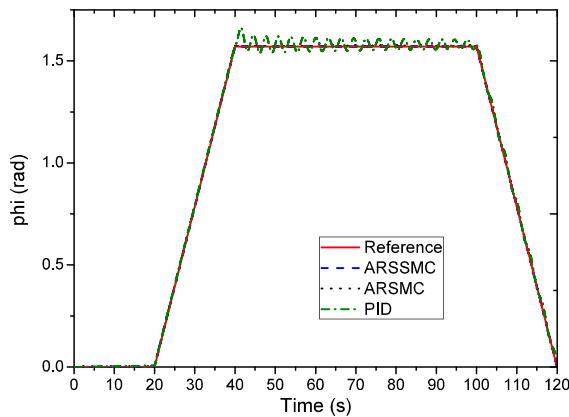
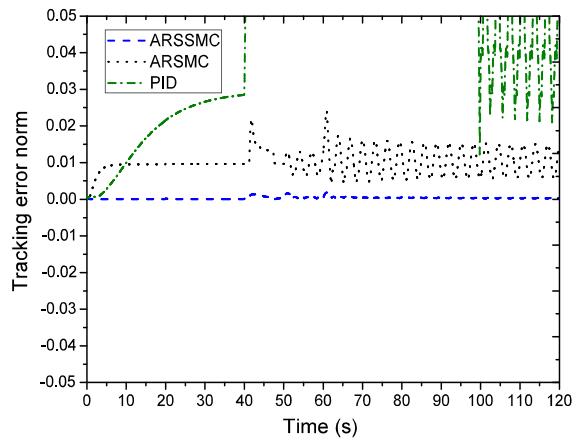
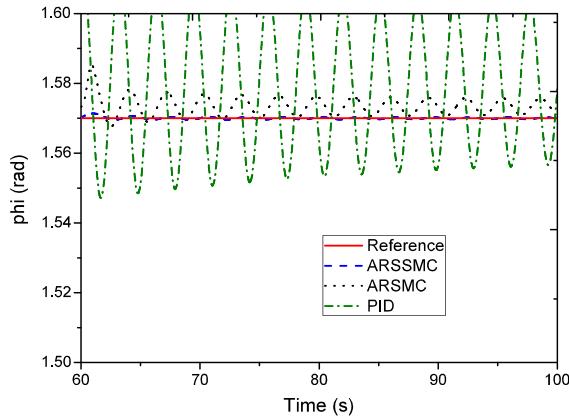
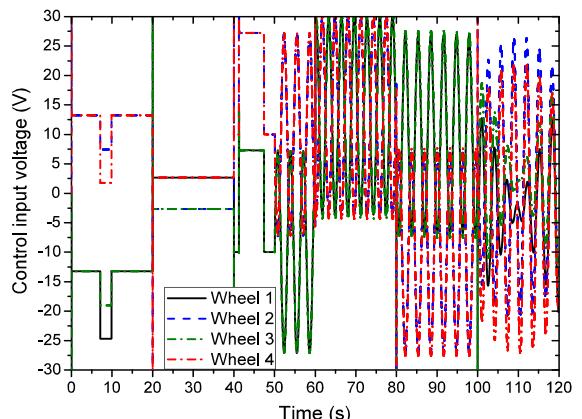
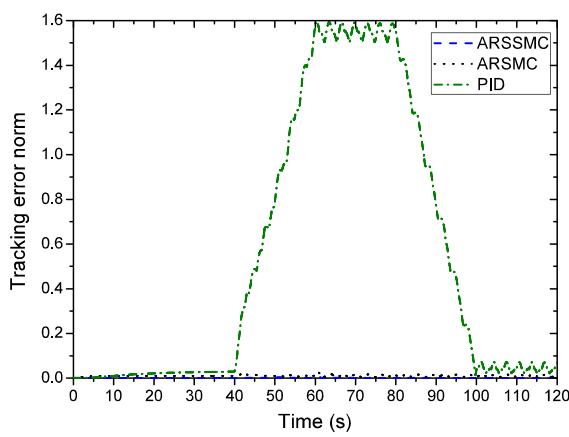
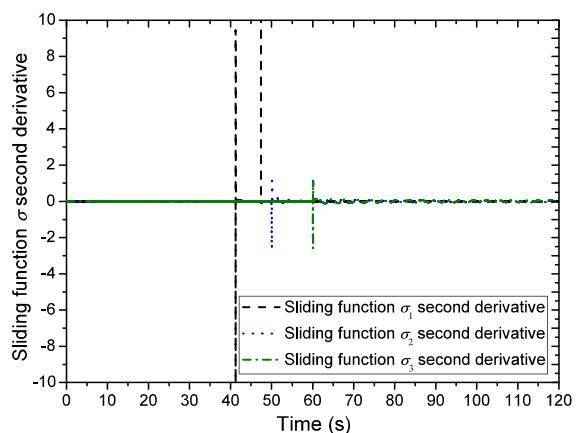
**Fig. 21** Trajectory II in x-y plane**Fig. 22** Trajectory II in x-y plane ( $41.2 < t \leq 47.4$ )

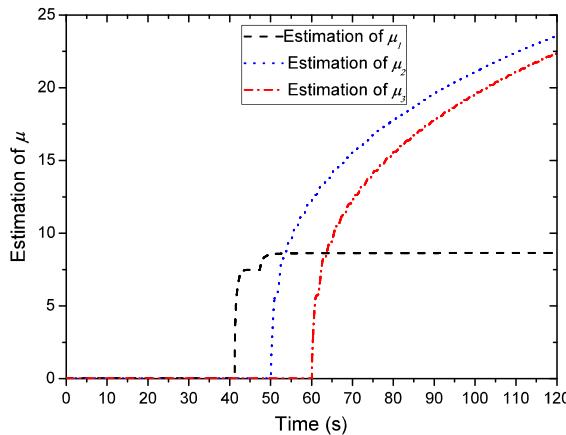
## 7.2 Trajectory II: close rectangular path

The equation of trajectory II is given as

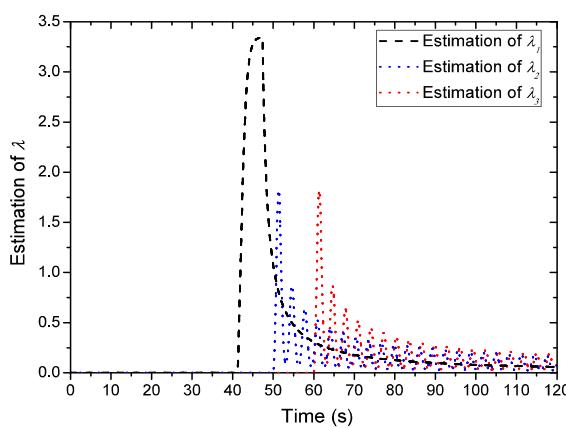
$$x = \begin{cases} 0.1t & \forall 0 < t \leq 20 \\ 2 & \forall 20 < t \leq 60 \\ -0.1t + 8 & \forall 60 < t \leq 80 \\ 0 & \forall 80 < t \leq 120 \end{cases}$$

$$y = \begin{cases} 0 & \forall 0 < t \leq 40 \\ 0.075t - 3 & \forall 40 < t \leq 60 \\ 1.5 & \forall 60 < t \leq 80 \\ 0.075t + 7.5 & \forall 80 < t \leq 120 \end{cases}$$

**Fig. 23** Orientation versus time plot**Fig. 26** Tracking error norm versus time plot for trajectory II (zoom in)**Fig. 24** Orientation versus time plot ( $60 < t \leq 100$ )**Fig. 27** Control effort of ARSSMC for trajectory II**Fig. 25** Tracking error norm versus time plot for trajectory II**Fig. 28** Second derivative of sliding function  $\sigma_1(t)$ ,  $\sigma_2(t)$  and  $\sigma_3(t)$



**Fig. 29** Estimation of  $\mu_1$ ,  $\mu_2$  and  $\mu_3$



**Fig. 30** Estimation of  $\hat{\lambda}_1$ ,  $\hat{\lambda}_2$  and  $\hat{\lambda}_3$

**Table 5** Tracking performance comparison (Trajectory II)

Controller	ISE	IAE	ITAE	
ARSSMC	$10^{-6}$	$\begin{bmatrix} 5.96 \\ 4.85 \\ 4.64 \end{bmatrix}$	$\begin{bmatrix} 0.01 \\ 0.013 \\ 0.012 \end{bmatrix}$	$\begin{bmatrix} 0.53 \\ 1.05 \\ 1.037 \end{bmatrix}$
ARSMC	$10^{-4}$	$\begin{bmatrix} 6 \\ 5.9 \\ 5.9 \end{bmatrix}$	$\begin{bmatrix} 0.45 \\ 0.42 \\ 0.41 \end{bmatrix}$	$\begin{bmatrix} 15.26 \\ 14.1 \\ 14.79 \end{bmatrix}$
PID	$10^{-2}$	$\begin{bmatrix} 4 \\ 3.9 \\ 4.6 \end{bmatrix}$	$\begin{bmatrix} 2.43 \\ 2.7 \\ 2.8 \end{bmatrix}$	$\begin{bmatrix} 87.45 \\ 126.8 \\ 128.4 \end{bmatrix}$

$$\phi = \begin{cases} 0 & \forall 0 < t \leq 20 \\ -0.0785t - 1.57 & \forall 20 < t \leq 40 \\ 1.57 & \forall 40 < t \leq 100 \\ -0.0785t + 9.42 & \forall 100 < t \leq 120 \end{cases} \quad (39)$$

where  $t$  is the simulation time in seconds. The initial posture of the mobile robot is

$$[x_q \ y_q \ \phi]^T = [0 \ 0 \ 0]^T.$$

the external force  $F_{ex}$  and bounded uncertainties  $\xi(t, u(t))_{3 \times 1}$  are taken as

$$F_{ex} = 1.5 \forall 7 < t \leq 10.$$

$$\xi(t, u(t))_{3 \times 1} = \begin{cases} 0.5 & \forall 41.2 < t \leq 47.4 \\ 0.5 \sin(2t) & \forall 50 \leq t \\ 0.5 \sin(2t) & \forall 60 \leq t \end{cases}$$

Design parameters for proposed ARSSMC is based on criterion mention in Sect. 7.1. The values calculated from pattern search optimization method are

$$k_{p_{3 \times 3}} = \begin{bmatrix} 142.6 & 0 & 0 \\ 0 & 167.8 & 0 \\ 0 & 0 & 136.7 \end{bmatrix},$$

$$k_{i_{3 \times 3}} = \begin{bmatrix} 51 & 0 & 0 \\ 0 & 41.9 & 0 \\ 0 & 0 & 32.1 \end{bmatrix},$$

$$k_{d_{3 \times 3}} = \begin{bmatrix} 23.8 & 0 & 0 \\ 0 & 19.2 & 0 \\ 0 & 0 & 16.4 \end{bmatrix},$$

$$k_{3 \times 3} = \begin{bmatrix} 1.05 & 0 & 0 \\ 0 & 2.67 & 0 \\ 0 & 0 & 3.13 \end{bmatrix},$$

$$\rho_1 = 29.5, \rho_2 = 22.1,$$

$$\rho_3 = 20.2, \gamma_1 = 32.6, \gamma_2 = 7.3, \text{ and } \gamma_3 = 26.1, \delta = 0.08.$$

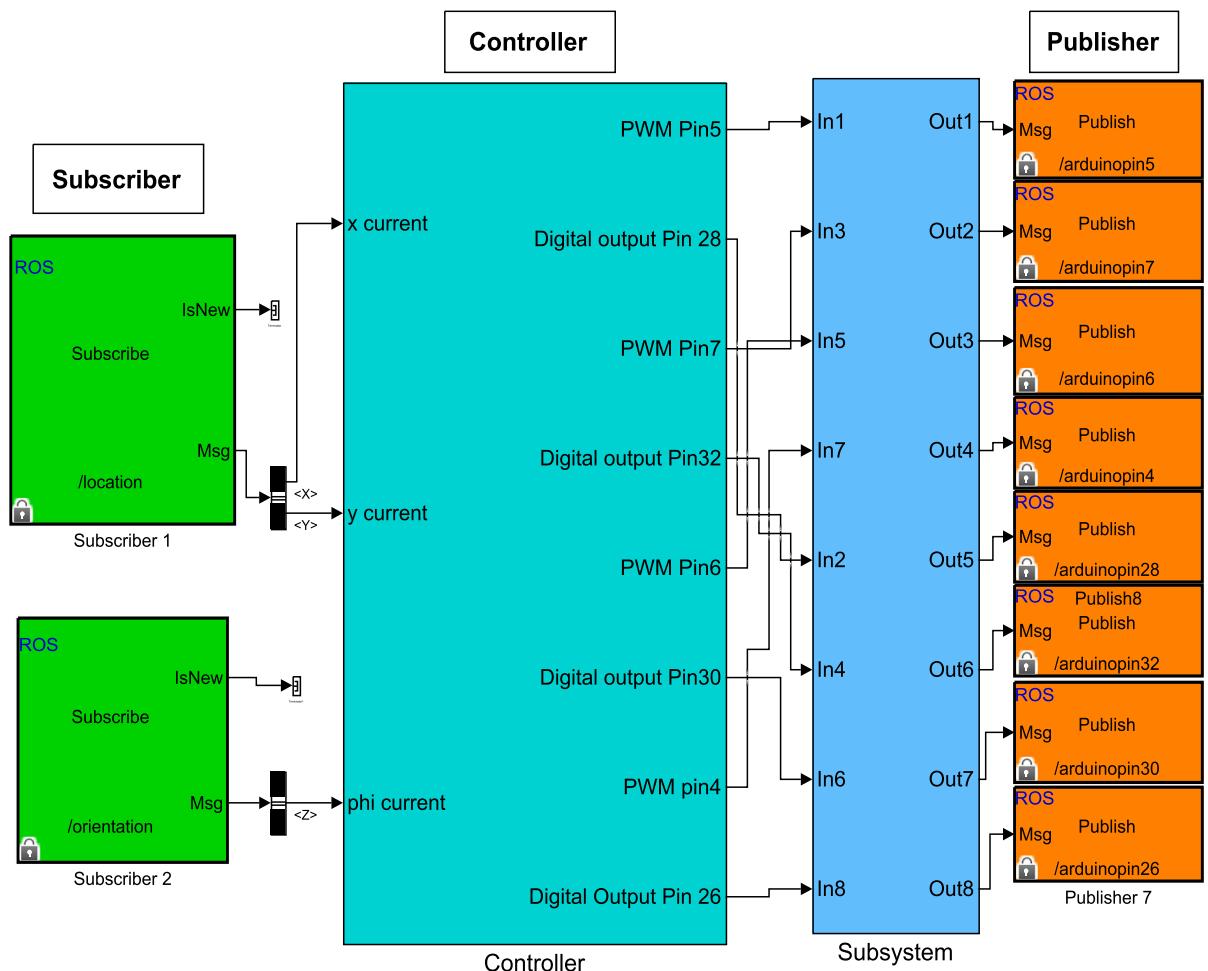
Figures 21 and 22 compares capability of ARSSMC, ARSMC and PID controller to track the closed rectangular path with changing orientation. As it can be seen that ARSSMC efficacy to track the trajectory is higher as compared to other two controllers. Angular trajectory tracking capability comparison is shown in Figs. 23 and 24, which proves that ARSSMC results closely matches with the reference trajectory. In order to quantify the tracking error, tracking error norm time history is presented in Figs. 25 and 26. It is evident that in the presence of uncertainties, ARSSMC tracking norm is reduced to a small value of the order  $10^{-5}$ , which is an acceptable value considering the movement of mobile robot in a real world scenario. The control effort at each wheel obtained by ARSSMC, is shown in Fig. 27. The sudden changes in the control input are related to the mobile robot effort during sharp turns and uncertainties. The second-order sliding function convergence with time is presented in Fig. 28. Figures 29

**Table 6** Controller performance comparison (Trajectory II)

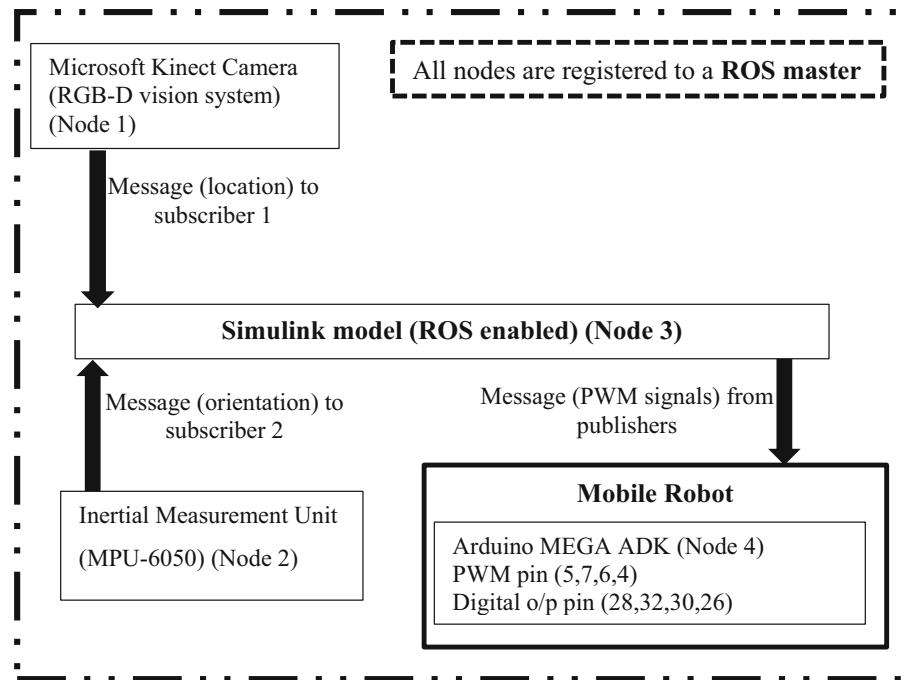
Controller	Control energy at each wheel ( $10^3$ ) (Nm)				Total variance of control input			
	Wheel1	Wheel 2	Wheel 3	Wheel 4	Wheel 1	Wheel 2	Wheel 3	Wheel 4
ARS-SMC	9.89	9.7	9.63	9.9	816.19	836.5	772.1	801.44
ARS-MC	18.3	18.24	17.2	20.1	936.7	972.8	903.7	921.8

**Table 7** Computational time and memory consumption (Trajectory II)

Controller	Computational time (s)	Memory consumption (MB)
ARSSMC	14.2	273.5 ( $\pm 3.2$ )
ARSMC	11.5	191.9 ( $\pm 1.9$ )
PID	7.8	165.8 ( $\pm 0.95$ )

**Fig. 31** Simulink model (ROS enabled)

**Fig. 32** Block diagram of ROS MATLAB experiment scheme

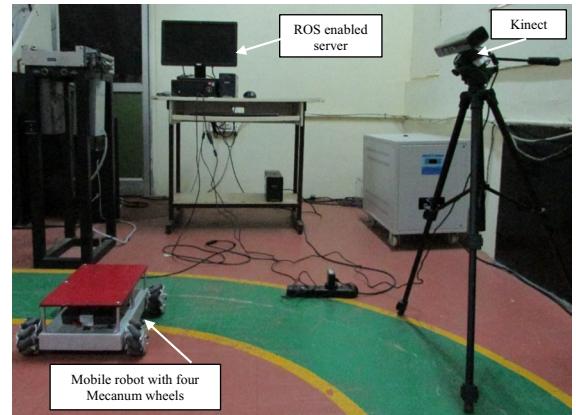


and 30 depicts the online estimation of  $\mu_1$ ,  $\mu_2$ ,  $\mu_3$ , and  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$  respectively.

Similar to trajectory I, to verify the tracking performance, ISE, IAE, and ITAE of ARSSMC, ARSMC and PID controllers are compared in Table 5. Further, to compare the control energy and variance of the proposed controller with other two controllers, Table 6 is presented. As can be seen from Table 4 and 5, ARSSMC tracks the rectangular closed path in the presence of uncertainties with smooth control voltage, less control energy and less tracking error. Finally, Table 7 presents a computational time and memory usage comparison of all the three controllers for Trajectory II.

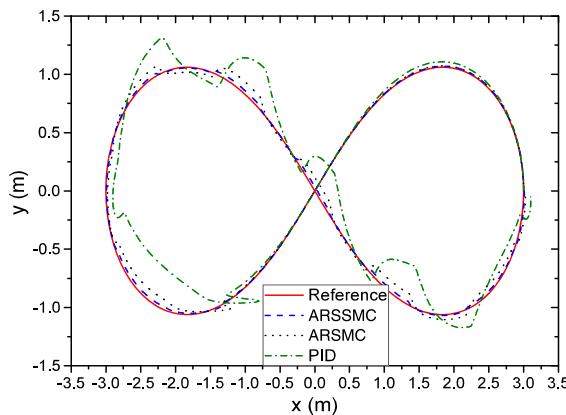
## 8 Experiments

To demonstrate the effectiveness and the supremacy of the proposed control law over ARSMC and PID controllers, a real-time control system has been implemented for the mobile robot with four Mecanum wheels. The robot consists of four Mecanum wheels and each wheel has nine rollers inclined at  $\pm 45^\circ$ . Each wheel is actuated by a brushless DC motor. The computer used for the implementation has the specification as, Intel Core-i5 CPU 3.20 GHz and 8 GB RAM. It should be noted that during the movement of the



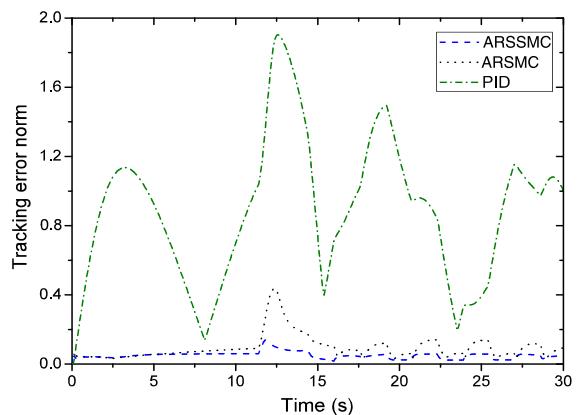
**Fig. 33** Experimental setup

mobile robot in a real environment, there is a tendency of slipping between driving wheels and the surface, which can result in position and orientation error. This is the major drawback if the current position and orientation of the mobile robot is measured using encoders. Hence, to nullify this loss of data during slippage, a Microsoft Kinect camera and an IMU sensor (MPU-6050) are used as position and orientation sensor. An Arduino MEGA ADK microcontroller is used for the implementation of control algorithm.



**Fig. 34** Trajectory tracking experiment result for trajectory I

The major challenge during the real-time implementation was to reduce the computational time while solving the nonlinear equations comprising the kinematics of the robot, the dynamics of the robot and the proposed control law. Moreover, use of wrong solver can lead to singularity error during the solution. Hence, ode2 (Heun) solver available in MATLAB/Simulink 2015b is used to solve the equation during real-time implementation. Next, it is also observed that delay in feedback data can lead to increase the convergence time of tracking error. Hence, robot operating system (ROS) packages for sensors is used to acquire real-time feedback data. Moreover, as ROS is Linux based, the processing time is less. Matlab Robot Operating System Toolbox is utilized to develop an interface between ROS and MATLAB/Simulink 2015b. When the run is started, ROS assumes all the parts of the real-time closed-loop system consisting of sensors, the computer on which equation is solved and microcontroller as nodes, enables these nodes to send and receive data in the form of messages. For example, Microsoft Kinect camera, IMU sensor, Simulink model (ROS enabled) (Fig. 31) and Arduino MEGA ADK are different nodes as shown in Fig. 32. The block diagram of the real-time control scheme is shown in Fig. 32. The Kinect sensor data and IMU sensor data publish topics in ROS. Then, the data in the form of messages is send to the subscriber of Simulink model, which sends the real time  $x(t) = [x_q \ y_q \ \phi]^T$  values to the controller. Based on the trajectory error, the control law gives the required control voltage in order to track the desired trajectory. The obtained control voltage at each motor is given in the form of pulse width modulation (PWM). In order



**Fig. 35** Trajectory error norm comparison for trajectory I



**Fig. 36** Overlay of several snapshots for trajectory I



**Fig. 37** Overlay of several snapshots for trajectory II

to send the controller o/p signals to PWM pins and digital o/p pins of the microcontroller, topics are published in Simulink, and Arduino subscriber running on ROS receives the signal in the form of messages which finally actuate the wheels.

Figure 33 shows the experimental setup with ROS enabled server, mobile robot and a Kinect camera. In

order to prove the supremacy of the proposed controller over ARSMC and PID, experiments are conducted to track the desired trajectory I. To test the robustness, an external disturbance of magnitude 3 is fed at 14 s. It is evident from Fig. 34 that in a real environment ARSSMC tracks the trajectory with minimum error. Compared to ARSSMC, ARSMC and PID controller are unable to track the reference trajectory. The efficacy of ARSSMC can also be proved from Fig. 35 where, which shows less tracking error for ARSSMC. The overlay of several snapshots of the mobile robot while performing trajectory I and II in real time is shown in Figs. 36 and 37. The satisfactory performance by ARSSMC verifies that the proposed method can track any type of complicated trajectory in presence of uncertainties.

## 9 Conclusion

In this paper, an adaptive robust trajectory tracking controller for a Mecanum-wheeled mobile robot has been proposed. Newton–Euler approach has been used to derive a generalized equation of motion of the robot in presence of external force disturbance, uncertainties and friction. To make the system robust, a higher-order sliding mode control law has been derived. Adaptive laws have been designed for the auto adjustment of switching gains in response to the uncertainties. Thus, it avoids the need for the prior knowledge of the bounds of the uncertainties. The asymptotic stability of the proposed control law is proved based on Lyapunov stability theory. The proposed controller efficacy has been tested for two different types of trajectories. Simulation and experimental results verify and prove the excellent tracking capability of ARSSMC with less control energy and smooth control input compared to ARSMC and PID controllers. The future work to design adaptive laws for sliding surface gain parameters is under investigation.

## References

- Pin, F.G., Killough, S.M.: A new family of omnidirectional and holonomic wheeled platforms for mobile robots. *IEEE Trans. Robot. Autom.* **10**(4), 480–489 (1994)
- Tzafestas, S.G.: *Introduction to Mobile Robot Control*. Elsevier, Waltham (2014)
- Muir, P.F., Neuman, C.P.: Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. In: Cox, I.J., Wilfong, G.T. (eds.) *Autonomous Robot Vehicles*, pp. 25–31. Springer, New York (1990)
- Conceicao, A.S., Moreira, A.P., Costa, P.J.: Practical approach of modeling and parameters estimation for omnidirectional mobile robots. *IEEE/ASME Trans. Mechatron.* **14**(3), 377–381 (2009)
- Tlale, N., De Villiers, M.: Kinematics and dynamics modelling of a mecanum wheeled mobile platform. In: 15th International Conference on Mechatronics and Machine Vision in Practice. pp. 657–662. IEEE (2008)
- De Villiers, M., Tlale, N.S.: Development of a control model for a four wheel mecanum vehicle. *J. Dyn. Syst. Meas. Control.* **134**(1), 011007 (2012)
- Yang, J.M., Kim, J.H.: Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Trans. Robot. Autom.* **15**(3), 578–587 (1999)
- Viet, T.D., Doan, P.T., Hung, N., Kim, H.K., Kim, S.B.: Tracking control of a three-wheeled omnidirectional mobile manipulator system with disturbance and friction. *J. Mech. Sci. Technol.* **26**(7), 2197–2211 (2012)
- Fierro, R., Lewis, F.L.: Control of a nonholonomic mobile robot using neural networks. *IEEE Trans. Neural Netw.* **9**(4), 589–600 (1998)
- Purwin, O., D Andrea, R.: Trajectory generation and control for four wheeled omnidirectional vehicles. *Robot. Auton. Syst.* **54**(1), 13–22 (2006)
- Xu, D., Zhao, D., Yi, J., Tan, X.: Trajectory tracking control of omnidirectional wheeled mobile manipulators: robust neural network-based sliding mode approach. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **39**(3), 788–799 (2009)
- Ryu, J.C., Agrawal, S.K.: Differential flatness-based robust control of mobile robots in the presence of slip. *Int. J. Robot. Res.* **30**(4), 463–475 (2011)
- Utkin, V.I.: *Sliding Modes in Control and Optimization*. Springer Science and Business Media, New York (2013)
- Utkin, V., Shi, J.: Integral sliding mode in systems operating under uncertainty conditions. In: Proceedings of the 35th IEEE Conference on Decision and Control, vol. 4, pp. 4591–4596. IEEE (1996)
- Utkin, V., Guldner, J., Shi, J.: *Sliding Mode Control in Electro-Mechanical Systems*. CRC Press, Boca Raton (2009)
- Fridman, L.M.: An averaging approach to chattering. *IEEE Trans. Autom. Control* **46**(8), 1260–1265 (2001)
- Fridman, L.M.: Chattering analysis in sliding mode systems with inertial sensors. *Int. J. Control.* **76**(9–10), 906–912 (2003)
- Boiko, I., Fridman, L., Iriarte, R.: Analysis of chattering in continuous sliding mode control. In: Proceedings of the 2005 American Control Conference, pp. 2439–2444. IEEE (2005)
- Bartolini, G., Pisano, A., Punta, E., Usai, E.: A survey of applications of second-order sliding mode control to mechanical systems. *Int. J. Control.* **76**(9–10), 875–892 (2003)
- Levant, A.: Principles of 2-sliding mode design. *Automatica* **43**(4), 576–586 (2007)
- Mondal, S., Mahanta, C.: Nonlinear sliding surface based second order sliding mode controller for uncertain linear systems. *Commun. Nonlinear Sci. Numer. Simul.* **16**(9), 3760–3769 (2011)

22. Salgado-Jimenez, T., Jouvencel, B.: Using a high order sliding modes for diving control a torpedo autonomous underwater vehicle. In: Proceedings of OCEANS 2003, vol. 2, pp. 934–939. IEEE (2003)
23. Mihoub, M., Nouri, A.S., Abdennour, R.B.: Real-time application of discrete second order sliding mode control to a chemical reactor. *Control Eng. Pract.* **17**(9), 1089–1095 (2009)
24. Chen, C.Y., Li, T.H., Yeh, Y.C., Chang, C.C.: Design and implementation of an adaptive sliding-mode dynamic controller for wheeled mobile robots. *Mechatronics* **19**(2), 156–166 (2009)
25. Chen, C.Y., Li, T.H., Yeh, Y.C.: EP-based kinematic control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile robots. *Inf. Sci.* **179**(1), 180–195 (2009)
26. Chen, N., Song, F., Li, G., Sun, X., Ai, C.: An adaptive sliding mode backstepping control for the mobile manipulator with nonholonomic constraints. *Commun. Nonlinear Sci. Numer. Simul.* **18**(10), 2885–2899 (2013)
27. Cui, M., Liu, W., Liu, H., Jiang, H., Wang, Z.: Extended state observer-based adaptive sliding mode control of differential-driving mobile robot with uncertainties. *Nonlinear Dyn.* **83**(1), 667–683 (2016)
28. Huang, J.T., Van Hung, T., Tseng, M.L.: Smooth switching robust adaptive control for omnidirectional mobile robots. *IEEE Trans. Control Syst. Technol.* **23**(5), 1986–1993 (2015)
29. Wang, H., Liu, L., He, P., Yu, M., Do, M.T., Kong, H., Man, Z.: Robust adaptive position control of automotive electronic throttle valve using PID-type sliding mode technique. *Nonlinear Dyn.* **85**(2), 1331–1344 (2016)
30. Slotine, J., Li, W.: *Applied Nonlinear Control*. Prentice-Hall, New Jersey (1991)
31. Lewis, R.M., Torczon, V., Trosset, M.W.: Direct search methods: then and now. *J. Comput. Appl. Math.* **124**(1), 191–207 (2000)

## Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”). Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

[onlineservice@springernature.com](mailto:onlineservice@springernature.com)