Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 704 – 716

doi:10.3906/elk-2003-12

A multiple sensor fusion based drift compensation algorithm for mecanum wheeled mobile robots

Abdulrahman ALHALABI[1], **Mert EZIM**[2], **Kansu Oguz CANBEK**[1], **Eray A. BARAN**[1,*]
[1]Mechatronics Engineering, Faculty of Engineering and Natural Sciences, Istanbul Bilgi University, Istanbul, Turkey
[2]Automation and Control Engineering, School of Industrial and Information Engineering Politecnico di Milano, Milan, Italy

Received: 02.03.2020 • **Accepted/Published Online:** 10.08.2020 • **Final Version:** 30.03.2021

Abstract: This paper investigates a multiple sensor fusion based drift compensation technique for a mecanum wheeled mobile robot platform. The mobile robot is equipped with high-precision encoders integrated to the wheels and four accelerometers placed on its chassis. The proposed algorithm combines the information from the encoders and the acceleration sensors to estimate the total drift in the acceleration dimension. The inner loop controller is designed utilizing a disturbance-observer-based acceleration control structure which is blind against the slipping motion of the wheels. The estimated drift acceleration from the sensor fusion is then mapped back to the joint space of the robot and used as additional compensation over the existing controllers. The proposed algorithm is tested on a series of experiments. The results of the experiments are also compared with those of a recent study in order to provide a benchmark evaluation. The enhanced tracking performance yielding towards smaller error magnitudes in the experiments illustrate the efficacy and success of the proposed control architecture in attenuating the positioning drift of mecanum wheeled robots.

Key words: Holonomic mobile robot, mecanum wheel, drift compensation, sensor fusion, acceleration control

1. Introduction

Precise positioning of mobile robots is among the most significant and difficult aspects of localization studies, since the robot wheels drift on slippery ground [1]. Furthermore, the task of localizing mobile robots becomes less feasible for holonomic mobile robots due to the specialized mecanum-type wheels [2]. The demand for mobile robots with mecanum wheels increases in many sectors because of their easy maneuvering ability. Today, these mecanum wheeled robots are used in factories and warehouses for tasks that require accurate and fast conveying of objects [3] and for semiautonomous inspection and maintenance duties in hazardous environments where humans cannot operate safely [4]. Besides navigation, manipulators attached to mobile robots with mecanum wheels enable a wider range of possible configurations to execute more sophisticated tasks such as haptic feedback generation [5] or intelligent manufacturing [6].

High-precision operation of mobile robots, on the other hand, is still among the most challenging problems requiring sophisticated solutions. Traditional control methods are restricted to kinematic models and are proven to be minimally efficient because they cannot model the drifting effects of the wheels [7]. Particularly in mobile robots with mecanum wheels, discontinuities between the rollers lead to larger amounts of disturbances [8]. Slipping can significantly change the orientation of the robot, making it harder to control and perform a desired task [9]. Addressing the disturbance problem due to wheel slip, several odometry-based control methods have

*Correspondence: eray.baran@bilgi.edu.tr

been studied in the literature. Among those methods, control methods using multiple ultrasonic distance sensors [10] or stereo CCD cameras [11] can be counted. On the other hand, the limited range sensing ability of ultrasonic sensors and the real-time computational requirements of visual odometry put certain limitations on the practical implementation of these solutions. The methods with comparatively low computational requirements are mostly based on the sensor sets like encoders, accelerometers, gyros, or inertial measurement units. On the other hand, the employment of different sensor types comes with the problem of data reliability. As a solution to this problem, sensor fusion techniques have been proposed for mobile robots in several studies [12, 13]. A variety of inertial sensors were fused with either odometry and/or vision systems to estimate the robot position better. However, inertial navigation sensors cannot be used alone due to their inherent random noise which results in potential integration drifts. To compensate for the effect of the induced noise, estimation methods relying on Kalman filter [14] or recursive online map generation [15] have been investigated.

Using multiple sensors of the same type was also proven to be an efficient approach in addition to serving as an obstacle avoidance method as shown in [16] and [17]. Particularly, the study given in [17] shows the mathematical derivations of a cost-effective solution where three ultrasonic sensors along with two receivers were used to estimate the position and orientation of a robot accurately. In [18], realization of a sensor fusion algorithm is shown for a mecanum wheeled mobile robot. This algorithm simply fuses two different inertial sensor data in a Kalman filter to contribute to the prediction of the angle error and to correct it for a secondary Kalman filter that estimates the robot position. Besides the sensor fusion, solutions based on mechanical improvements and more sophisticated control algorithms also contribute to the positioning precision of mobile robots [19]. However, precise positioning of mecanum wheeled mobile robots without expensive absolute measurement techniques like differential GPS seems to be a serious challenge for the practical implementations.

Addressing the positioning drift of mecanum wheeled mobile robots, in this paper, a sensor fusion algorithm is presented making use of four analogue accelerometers and measurements of the wheel rotations taken from the motor encoders. The algorithm relies on taking the differences of the robot center acceleration vectors obtained from these two sensor sets. The center acceleration vectors are calculated by applying the forward kinematics of the robot on double differentiated encoder measurements and by fusing the four accelerometer measurements located at the corners of the platform, respectively. The difference in acceleration vectors is then used to enforce the compensation torque of each wheel in order to minimize the drifting motion. The inner loop reference position tracking of the wheels is enforced by a disturbance-observer-based robust acceleration control scheme. The proposed algorithm is verified through a set of experiments making use of different reference position trajectories. Meanwhile, the performance of the algorithm is verified with a commercial mouse sensor attached to the bottom of the experimental platform to track the robot's displacement in X and Y axes.

The organization of the paper is as follows: In Section 2, a general overview of the system under consideration is given where the kinematics of the system is derived and the details of the experimental platform is explained. In Section 3, the methods used for the sensor fusion and the estimation of drifting motion are explained. Section 4 is reserved for the derivation of the drift compensation algorithm where the robust acceleration control method is also summarized. In Section 5, the illustrations and comparative numerical evaluations of the experiment results are given. Finally in Section 6, concluding remarks are presented.

2. System description

2.1. Design

The mobile robot used in this study is composed of four mecanum wheels to provide omnidirectional motion with holonomic constraints. In order to provide a deeper insight about this structure, a schematic drawing is provided in Figure 1. In this figure, symmetrically positioned accelerometers measuring the X-axis and Y-axis accelerations are shown. Here, $\mathbf{r}_n = [r_{n,x}, r_{n,y}]^T \in \mathbb{R}^{2\times 1}$ represents the vector from the center of the robot chassis to the n^{th} accelerometer center for $n = \{1, 2, 3, 4\}$. Furthermore, the variables A_x, A_y, α, and ω respectively stand for the linear accelerations of the robot chassis center along the x and y axes, the angular acceleration of the robot and the angular velocity of the robot, all being computed from the accelerometer measurements. These variables will be used throughout the derivation shown in the following sections.

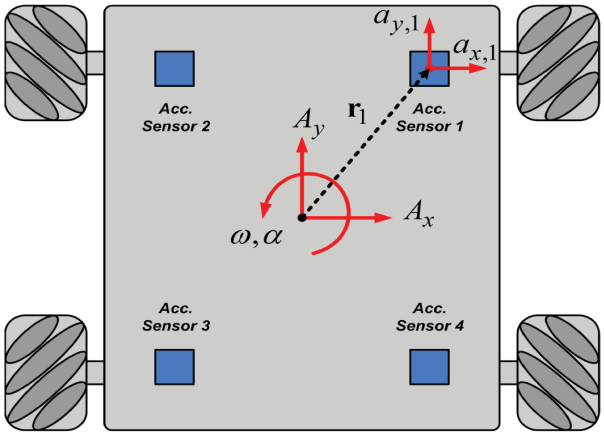
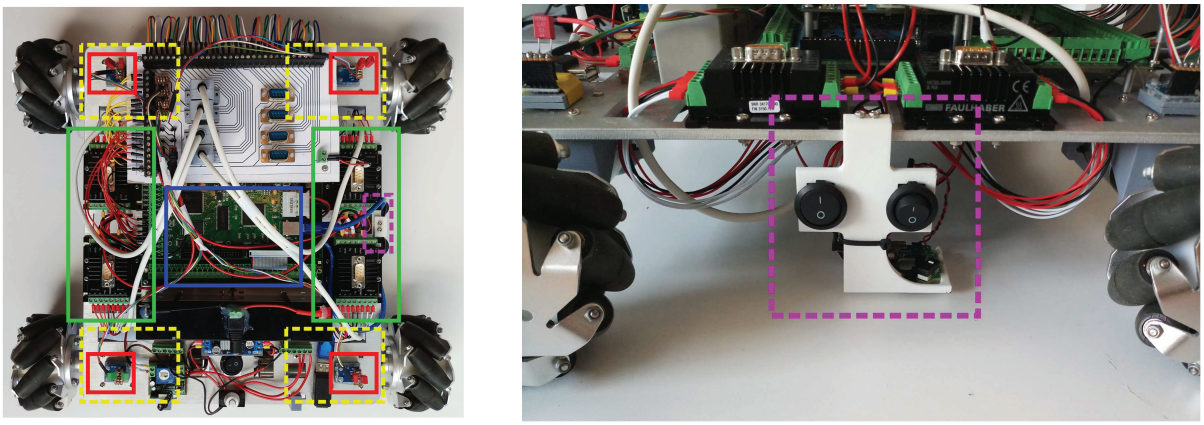


Figure 1. Schematic representation of the mobile robot

2.2. Setup

The mobile robot platform used for the validation of the proposed algorithm is shown in Figure 2. The setup contains four actuators from Faulhaber 3242-G-012-BX series brushless DC motor systems. Each motor is equipped with an incremental encoder having 1000 lines/rev. resolution. The motors are connected to the wheels with planetary gearboxes having a 1:19 reduction ratio. Driving of the motors are made in torque control mode using Faulhaber $MCBL$-3006-S driver systems. The accelerometers are selected from $ADXL$-335 series which provide an analogue output with a measurement range of ± 3 g. All four accelerometers are positioned on the chassis with distances $r_{n,x} = 147$ mm and $r_{n,y} = 194$ mm. The mecanum wheels have 0.05-m radius containing a total of 9 rollers per wheel. The control algorithm is executed on a real-time control board from Dynomotion KFlop+KAnalog series with implementation being done by coding in C language. The algorithm is finally verified with an optical mouse sensor that was detached from a Logitech M185 series commercial wireless mouse. The optoelectronic sensor of the mouse is mounted on the chassis of the mecanum wheeled robot and has a resolution of 800 dots per inch (dpi) which is scaled to meters by an open-source program called MouseTester . The optoelectronic sensor can provide measurements with a sampling frequency of 125 Hz. During the tests prior to the implementation of the proposed algorithm, the error margin of the optoelectronic sensor is measured to be $\pm 0.5\%$ on flat ground. The 2.4 GHz wireless communication interface of the optoelectronic sensor leads to a measurement delay of approximately 100 ms. However, as this measurement

is only used for validation purpose and is not used in the feedback loop, the measurement delay does not create any problem during the implementation. The overall weight of the robotic platform is measured to be 8.2 kg including the batteries while the dimensions of the system are measured as 415.8 mm × 436.4 mm × 158.3 mm for the length, width, and height, respectively. In order to better illustrate the implementation details of the real-time algorithm, a block diagram depicting the signal flow between the components of the system is shown in Figure 3.



(A) (B)

Figure 2. Mobile robot setup: accelerometers (red boxes), motors (yellos boxes), motor drivers (green boxes), control board (blue box), optoelectronic position sensor (purple box). Top view (a) and side view (b).

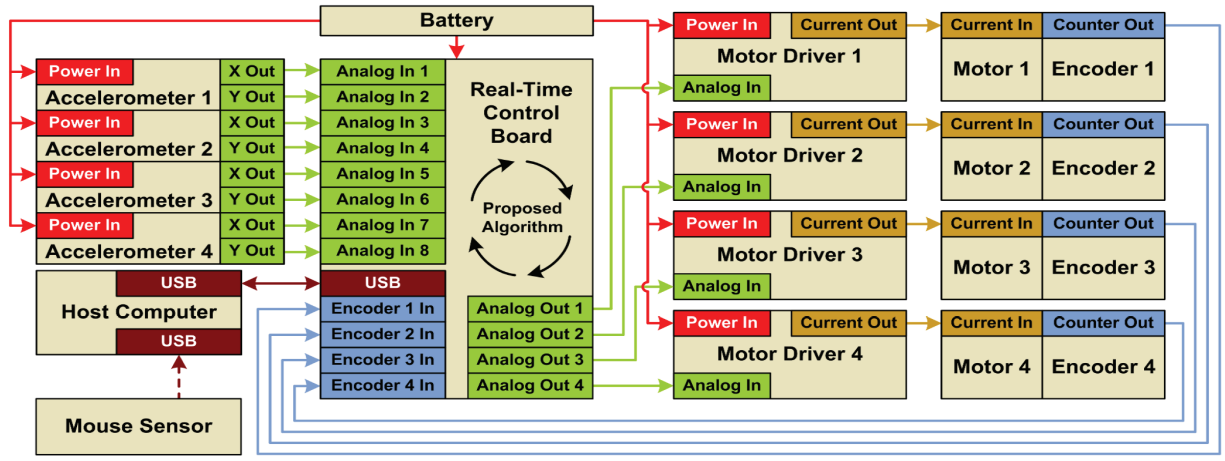


Figure 3. Signal flow between the components of the robot.

2.3. Kinematics

The kinematic analysis of mobile robots are usually different than that of robot manipulators. Since the mobile platform does not have any solid linkage with the earth, the notion of absolute motion does not make much sense. Hence, in mobile robots, usually the velocity level kinematics is taken into consideration. The kinematic relationship of the holonomic mobile robot under consideration has been well studied in many papers [20, 21]. Here, in order to give a complete framework, we summarize these results referring to the variables shown in

Figure 1. The relationship between the wheel velocities (i.e. joint space) and the robot chassis velocity (i.e. task space) can be given with the following formula:

$$\dot{\mathbf{\Phi}} = \mathbf{\Psi}\dot{\mathbf{\Lambda}} \tag{1}$$

where $\dot{\mathbf{\Phi}} = \begin{bmatrix} \dot{\phi}_1 & \dot{\phi}_2 & \dot{\phi}_3 & \dot{\phi}_4 \end{bmatrix}^T \in \mathbb{R}^{4\times 1}$ represents the vector of joint space velocities with $\dot{\phi}_i$ being the angular speed of the i^{th} wheel and $\dot{\mathbf{\Lambda}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T \in \mathbb{R}^{3\times 1}$ stands for the vector of task space velocities (i.e. the derivatives of the actual states of the robot). In (2), the term $\mathbf{\Psi} \in \mathbb{R}^{4\times 3}$ has the following content:

$$\mathbf{\Psi} = \frac{1}{R}\begin{bmatrix} -1 & 1 & (b+a) \\ -1 & -1 & (b+a) \\ 1 & -1 & (b+a) \\ 1 & 1 & (b+a) \end{bmatrix} \tag{2}$$

Here, variables a and b represent the distance from the robot chassis center to the center of each wheel in the vertical and horizontal directions, respectively. Further, the variable R represents the radius of the mecanum wheels. Using (1) and (2), the kinematic Jacobian ($\mathbf{J} \in \mathbb{R}^{3\times 4}$) for the velocity mapping of the system under consideration can then be obtained using the following pseudoinverse operation:

$$\mathbf{J} = \left(\mathbf{\Psi}^T\mathbf{\Psi}\right)^{-1}\mathbf{\Psi}^T \tag{3}$$

which yields:

$$\mathbf{J} = \frac{R}{4}\begin{bmatrix} -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{(b+a)} & \frac{1}{(b+a)} & \frac{1}{(b+a)} & \frac{1}{(b+a)} \end{bmatrix} \tag{4}$$

Having formulated the kinematic Jacobian, one can finally obtain the mappings between the configuration space and the operational space respectively for the velocity and the acceleration levels as:

$$\dot{\mathbf{\Lambda}} = \mathbf{J}\dot{\mathbf{\Phi}} \tag{5}$$

$$\ddot{\mathbf{\Lambda}} = \mathbf{J}\ddot{\mathbf{\Phi}} \tag{6}$$

It is important to recall here that the term $\dot{\mathbf{J}}\dot{\mathbf{\Phi}}$ that should naturally exist in equation (6) after differentiation is ignored since the kinematic Jacobian has constant parameters resulting in $\dot{\mathbf{J}} = \mathbf{0}$.

3. Sensor fusion and drift estimation

3.1. Accelerometer data fusion

As also mentioned in the preceding sections, the platform under consideration is equipped with 4 accelerometers each of which gives 3-axis measurements. Here, a summary of accelerometer data fusion algorithm is provided from which the accelerations and the angular speed of the mobile robot platform is extracted in the control loop. The derivations are made using the dimensions and descriptions given in Figure 1. Originating from the rigid body dynamics, one can write the generalized acceleration vector $\mathbf{a} \in \mathbb{R}^{3\times 1}$ at any specific point as:

$$\mathbf{a} = \mathbf{A}_0 + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) + \boldsymbol{\alpha} \times \mathbf{r} \tag{7}$$

Here, $\mathbf{A}_0 = [A_x, A_y, 0]^T \in \mathbb{R}^{3\times1}$, $\boldsymbol{\omega} = [0, 0, \omega]^T \in \mathbb{R}^{3\times1}$, $\boldsymbol{\alpha} = [0, 0, \alpha]^T \in \mathbb{R}^{3\times1}$, and $\mathbf{r} = [r_x, r_y, 0]^T \in \mathbb{R}^{3\times1}$ stand for the linear acceleration of the center of the robot chassis, the angular velocity of the robot, the angular acceleration of the robot, and the vector from the center of the robot to the selected point, respectively. The operator $\times$ represents the cross product in (7). Expanding equation (7) for the n^{th} accelerometer ($n \in \{1, 2, 3, 4\}$), one can obtain:

$$\begin{aligned} a_{x,n}\boldsymbol{i} + a_{y,n}\boldsymbol{j} = A_x\boldsymbol{i} + A_y\boldsymbol{j} + \omega\boldsymbol{k} \times (\omega\boldsymbol{k} \times (r_{x,n}\boldsymbol{i} + r_{y,n}\boldsymbol{j})) \\ +\alpha\boldsymbol{k}\,(r_{x,n}\boldsymbol{i} + r_{y,n}\boldsymbol{j}) \end{aligned} \tag{8}$$

with $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ standing for the unit vectors in x, y, and z axes, respectively. Decoupling (8), the identities relating the center acceleration of the robot chassis to the accelerometer measurements can be calculated as:

$$a_{n,x} = A_x - \omega^2 r_{n,x} - \alpha r_{n,y} \tag{9}$$

$$a_{n,y} = A_y - \omega^2 r_{n,y} + \alpha r_{n,x} \tag{10}$$

Using the symmetry of the system given in Figure 1, it is possible to write down $r_{n,x} = \{-r_x, r_x, r_x, -r_x\}$ and $r_{n,y} = \{-r_y, -r_y, r_y, r_y\}$ for $n = \{1, 2, 3, 4\}$, respectively. Combining the equations (9) and (10) for all four accelerometers, it is possible to formulate the following identity:

$$\begin{bmatrix} 1 & 0 & -r_x & -r_y \\ 0 & 1 & -r_y & r_x \\ 1 & 0 & r_x & -r_y \\ 0 & 1 & -r_y & -r_x \\ 1 & 0 & r_x & r_y \\ 0 & 1 & r_y & -r_x \\ 1 & 0 & -r_x & r_y \\ 0 & 1 & r_y & r_x \end{bmatrix} \begin{bmatrix} A_x \\ A_y \\ \omega^2 \\ \alpha \end{bmatrix} = \begin{bmatrix} a_{1,x} \\ a_{1,y} \\ a_{2,x} \\ a_{2,y} \\ a_{3,x} \\ a_{3,y} \\ a_{4,x} \\ a_{4,y} \end{bmatrix} \tag{11}$$

Solution of this linear system yields the acquisition of the accelerations of the robot chassis along with its angular velocity. The identity given in (11) provides a natural sensor fusion structure with certain benefits. Since (11) has an overdetermined structure, its solution attenuates the effect of noise in the measurement and provides a more reliable information compared to individual sensor measurements.

3.2. Estimation of the drifting motion

Equations (6) and (11) provide two different ways to estimate the acceleration of the mobile robot. The first one is obtained from the joint space position measurements via the encoder data. The encoders provide an exact measurement of the angular positions of each wheel. Calculation of the task space acceleration is then made by proper differentiation of these wheel positions and kinematics transformation (6) applied on the wheel angular accelerations. On the other hand, the task space accelerations are also gathered using the fusion of accelerometer data (11) as explained in the preceding section. Among those two acceleration measurements, the one obtained from the encoders is blind against the wheel drift while the later one is affected by the slipping motion of the robot as it is based on direct accelerometer data. The drift of the robot in the acceleration dimension can be acquired from the difference of these two measurements as:

$$\delta\ddot{\mathbf{\Lambda}} = \mathbf{A}_s - \ddot{\mathbf{\Lambda}} \tag{12}$$

where $\delta\ddot{\mathbf{\Lambda}} = \begin{bmatrix} \delta\ddot{x} & \delta\ddot{y} & \delta\ddot{\theta} \end{bmatrix}^T \in \mathbb{R}^{3\times1}$, $\ddot{\mathbf{\Lambda}} = \begin{bmatrix} \ddot{x} & \ddot{y} & \ddot{\theta} \end{bmatrix}^T \in \mathbb{R}^{3\times1}$, and $\mathbf{A}_s = \begin{bmatrix} A_x & A_y & \alpha \end{bmatrix}^T \in \mathbb{R}^{3\times1}$ stand for the vectors showing the drift in acceleration dimension, the accelerations computed from the encoders, and the accelerations obtained from the accelerometer measurements, respectively. This difference is induced from slippage where the encoders may sense a change in position even though no change in its physical location occurs. In the following subsection, the drift compensation algorithm will be derived using the acceleration measurement error given in (12). In order to provide further insight about this acceleration difference, measurements taken from the mobile robot used in this study are illustrated in Figure 4a. In this figure, the acceleration measurements are taken while the robot was tracing a trajectory in the shape of infinity sign. Furthermore, the results given in Figures 4b and 4c respectively stand for the velocity and position responses obtained from direct integration of the accelerations shown in Figure 4a. In those plots, the differences between the measurements obtained from the encoders and from the accelerometers clearly show the extent of the drifting motion. Especially in position control, the disturbance created by the drifting motion is even bigger than the reference motion, which is an unacceptable situation for precise localization.

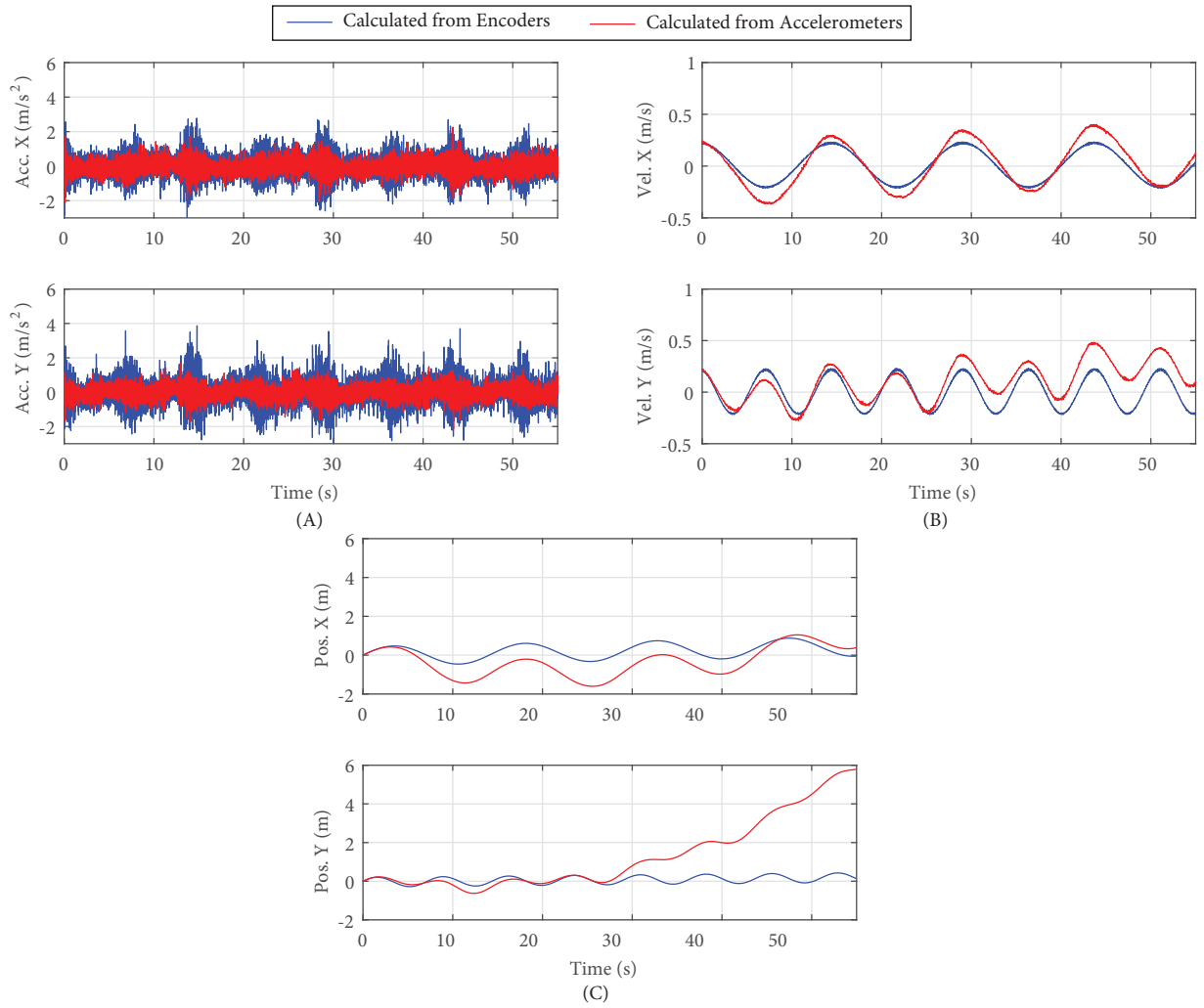


Figure 4. Responses obtained from the encoders and accelerometers: accelerations (a), velocities (b), positions (c).

4. Proposed drift compensation method

The proposed drift compensation algorithm relies on the generation of acceleration references for the wheels. On the other hand, tracking of these acceleration references requires proper estimation and compensation of the disturbances acting on the wheels. In order to present the whole picture, firstly, a summary of the inner loop disturbance compensation algorithm is given which is followed by the proposed drift control method.

4.1. Disturbance-observer-based robust acceleration control

Compensation of the disturbances acting on a motion control system is a very well-investigated concept in many different studies [22–24]. Here, a short derivation of the robust acceleration tracking control algorithm will be made for a single degree of freedom (DOF) system. The plant dynamics for a single DOF motion control system (i.e. for each wheel) can be given by:

$$J(\phi)\ddot{\phi}(t) + B(\phi,\dot{\phi}) + G(\phi) = \tau_c(t) - \tau_e(t) \tag{13}$$

where the scalars $J(\phi)$, $B(\phi,\dot{\phi})$, $G(\phi)$, $\tau_c(t)$, and $\tau_e(t)$ represent the inertia of the plant, the viscous friction torque, the load torque induced by the gravity, the control torque, and the external torque acting on the system, respectively. The dynamics given in (13) contains all of the torques due to the mechanical constraints of the system under consideration. The realization of the robust motion control depends on the estimation and compensation of the nonlinear terms of (13). Furthermore, keeping in mind that the control torque can be written as a scalar multiple of the control current (i.e. $\tau_c(t) = K_t i_c$), the previous equation can be rearranged as follows:

$$J_n\ddot{\phi}(t) = K_n i_c(t) - \tau_d(t) \tag{14}$$

where J_n and K_n stand for the nominal inertia value (i.e. $J(\phi) = J_n + \Delta J$) and the nominal torque constant of the system under consideration (i.e. $K_t = K_n + \Delta K$). In (14), the term $\tau_d(t)$ represent the disturbance torque acting on the system which has the following content:

$$\tau_d(t) = \Delta J\ddot{\phi}(t) + B(\phi,\dot{\phi}) + G(\phi) + \tau_e(t) - \Delta K i_c(t) \tag{15}$$

Here, $\Delta J\ddot{\phi}(t)$ and $\Delta K i_c(t)$ respectively represent the acceleration-induced disturbance torque and the current-induced disturbance torque acting on the system. Using the nominal parameters K_n and J_n of the system and a low-pass filter, the estimation of the disturbance torque ($\hat{\tau}_d(t)$) can be made via a disturbance observer (DOB) which is illustrated in Figure 5. Further information on the details of the DOB structure can be found in [25] while a stability and robustness analysis of the DOB-based motion control systems is given in [26]. Assuming a high-accuracy estimation of the disturbance term (i.e. $\tau_d(t) - \hat{\tau}_d(t) \approx 0$), one can write down the following identity:

$$\frac{J_n}{K_n}\ddot{\phi}(t) = i_c(t) \tag{16}$$

Equation (16) can be interpreted in several ways. The most important implication of this identity is that a motion control system with disturbance compensation can trace the desired acceleration references by properly scaled control currents. Hence, a plant with the inner loop DOB can be referred as a robust motion control system which can follow the enforced acceleration references. Accordingly, the drift compensation method proposed in the following subsection will be formulated to yield the desired acceleration references for system under consideration.

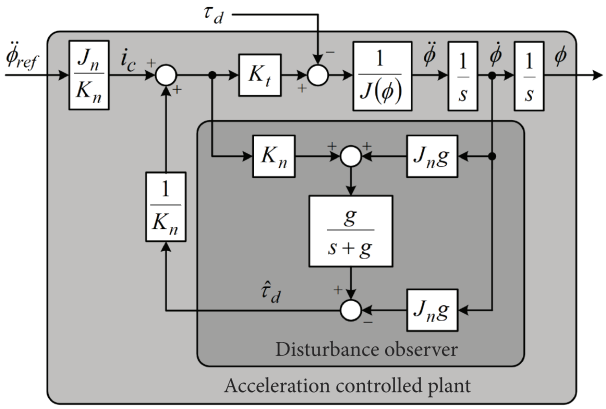


Figure 5. Block diagram of the robust acceleration control algorithm based on disturbance observer.

4.2. Drift estimation and compensation

After the acquisition of the acceleration drift vector $\delta\ddot{\mathbf{\Lambda}} = \begin{bmatrix} \delta\ddot{x} & \delta\ddot{y} & \delta\ddot{\theta} \end{bmatrix}^T$, one can multiply it with $\mathbf{\Psi}$ to convert the acceleration drift back to the wheels' angular acceleration drift. This relation is given as follows:

$$\mathbf{\Psi}\delta\ddot{\mathbf{\Lambda}} = \delta\ddot{\mathbf{\Phi}} \tag{17}$$

where $\delta\ddot{\mathbf{\Phi}} = \begin{bmatrix} \delta\ddot{\phi}_1 & \delta\ddot{\phi}_2 & \delta\ddot{\phi}_3 & \delta\ddot{\phi}_4 \end{bmatrix}^T \in \mathbb{R}^{4\times 1}$ stands for the vector of wheel drifts in the acceleration dimension. Basically, $\delta\ddot{\mathbf{\Phi}}$ contains the amount of difference in acceleration for each wheel that is needed to be controlled to have a less drifting wheel and better follow the reference path. In order to create additional compensation torque from (17), one has to multiply the drift accelerations with the nominal inertia of the motors and wheel set J_n, which can be given as:

$$\delta\boldsymbol{\tau} = \begin{bmatrix} \delta\tau_1 & \delta\tau_2 & \delta\tau_3 & \delta\tau_4 \end{bmatrix}^T \tag{18}$$

Here, $\delta\boldsymbol{\tau}_n \in \mathbb{R}^{4\times 1}$ with $n \in \{1,2,3,4\}$ represents the control torque for the compensation of the drift in the n^{th} wheel. The inner loop compensation, on the other hand, is made by calculating the reference accelerations using a PD control algorithm and enforcing them via the disturbance observer (DOB)-based acceleration control method as detailed in the previous section. The power of the proposed control scheme relies on the generation of additional feedback directly in the acceleration dimension which can easily be fused with the inner loop controller. In order to provide further insight about the proposed control structure, a schematic drawing of the controller block diagram is shown in Figure 6.

5. Experiments

The algorithm was tested to prove its efficiency in compensating drift caused mainly by slippage and uneven floor. The slippage would be more observable with complex references where the wheels are required to change rotation speed and direction rapidly during the desired motion.

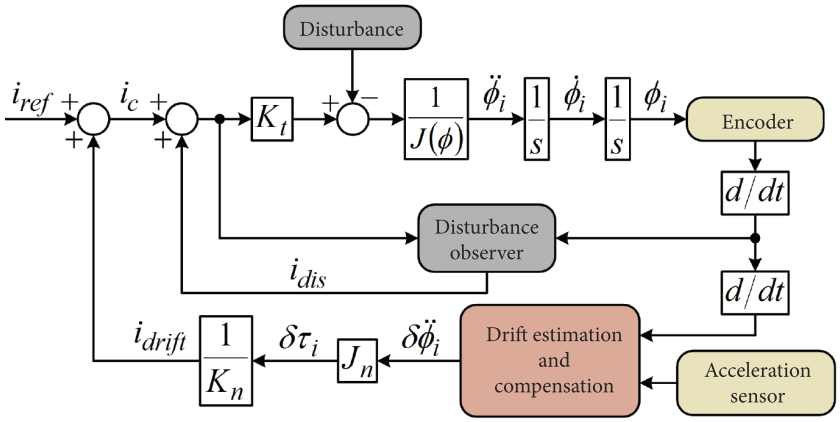


Figure 6. Block diagram of the proposed drift compensation algorithm.

5.1. Setting of the experiments

In order to highlight the performance of the proposed compensation algorithm, three different sets of experiments are made. In each experiment set, two different tests are made for the cases without any compensation and with the proposed compensation method making use of the same reference trajectory. The references in the experiment sets are selected as circular trajectory, infinity sign trajectory, and sinusoidal trajectory on X-Y plane. The alternation between these trajectory profiles had the specific purpose of testing the proposed algorithm under different acceleration responses. The circular and infinity trajectories aim at testing the responses under uniform and periodic accelerations with the same frequencies and with different frequencies in two axes, respectively. On the other hand, sinusoidal trajectory is selected to observe the performance with a periodic reference on one axis and a linear reference on the other axis. The duration of the tests are selected such that the uncompensated response in each experiment set leads to at least 5% tracking error.

5.2. Experiment results

The results obtained from these experiments are illustrated in Figure 7. For each experiment set, the planar trajectories are illustrated in (a)-type plots while the time-dependent X and Y trajectories (i.e. $X(t)$ and $Y(t)$) are shown in (b)- and (c)-type plots, respectively. Inspecting the results shown in the response plots, the superior performance of the proposed control architecture can be clearly observed. In all of the experiments, the compensated responses show better tracking than the uncompensated response.

The graphical responses are further supported by the numerical results obtained from the experiments. The table shows the root mean squared (RMS) tracking errors of the reference trajectories along with the improvements in errors with the proposed control scheme. In the table, the superscripts $*^{nc}$ and $*^{wc}$ represent the responses taken from experiments without additional compensation and with the proposed compensation method, respectively. The numerical results in the improvements are also compared with another recent study focusing on a state observer-based sliding mode control approach for the trajectory tracking of mecanum wheeled mobile robots [27]. Using [27] as a benchmark study, the improvement in the RMS value of tracking error with respect to the uncompensated case for the circular reference trajectory is increased from 27.8% to 47.3% in the x-axis and from 35.1% to 52.6% in the y-axis, respectively. Similarly, the improvement of the RMS value of tracking error for the infinite trajectory is increased from 52.8% to 58.3% in the x-axis and from 52.3% to

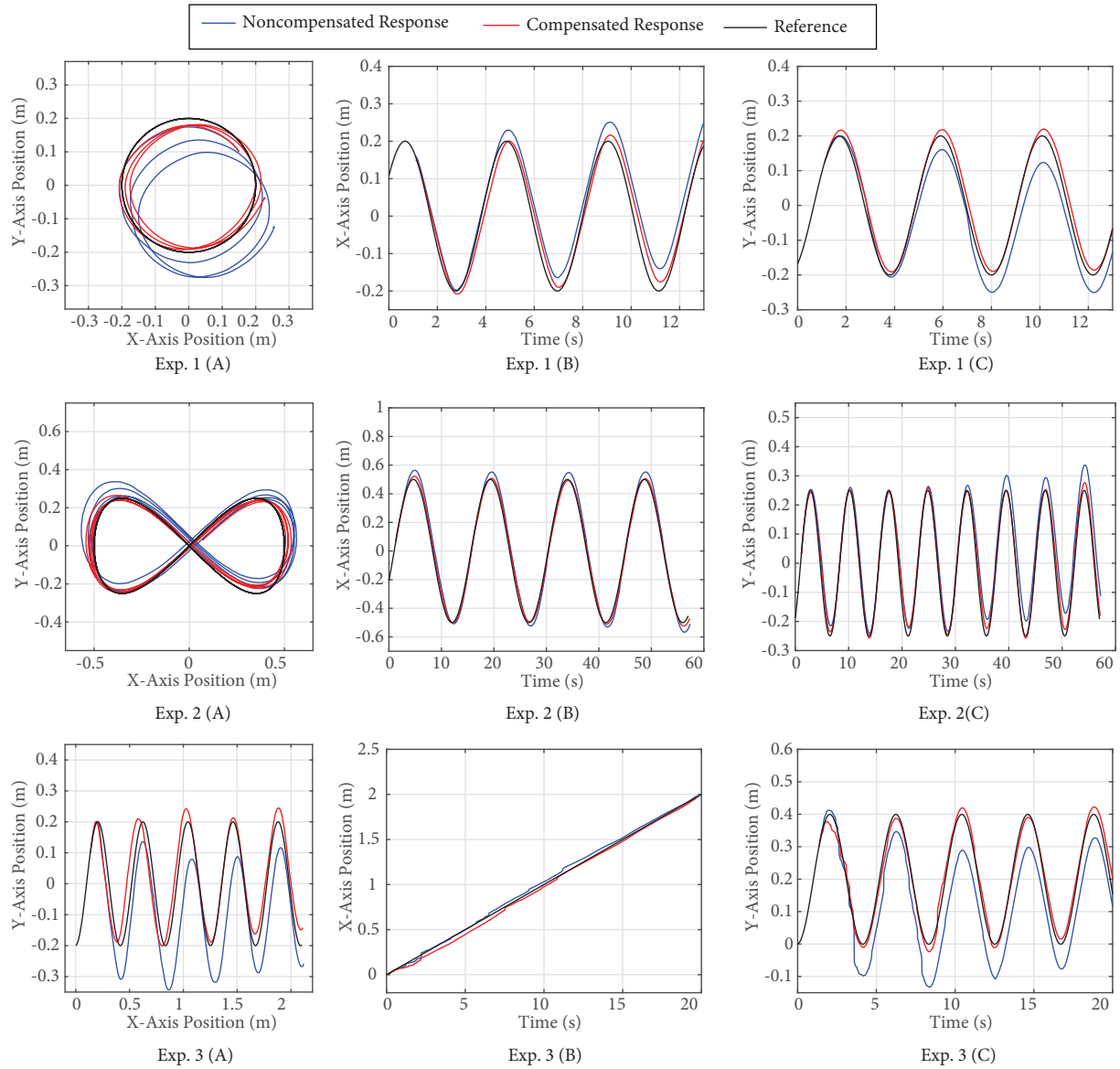


Figure 7. Experiment results. left column: responses obtained on XY plane. Middle column: X-axis reference trajectories and responses. Right column: Y-axis reference trajectories and responses.

61.4% in the y-axis, respectively. These RMS error values also prove the superior performance of the proposed drift compensation method.

Table . Root mean squared tracking errors.

	$\varepsilon^{nc}_{rms,x}$	$\varepsilon^{wc}_{rms,x}$	Improvement (%)	$\varepsilon^{nc}_{rms,y}$	$\varepsilon^{wc}_{rms,y}$	Improvement (%)
Experiment Set-1	1.4686	0.7728	47.3	1.4572	0.6913	52.6
Experiment Set-2	1.7462	0.7280	58.3	1.6725	0.6460	61.4
Experiment Set-3	1.7822	1.5712	11.8	1.5207	0.4308	71.7

6. Conclusion

In this study, a novel drift compensation algorithm is proposed for precise indoor navigation of mecanum wheeled robots. The proposed algorithm relies on the estimation of the acceleration drift, which is used for the generation of compensation torques for the wheels against drifting motion. The estimation algorithm makes use of multiple accelerometer data and the position measurements of the wheels from the encoders. The estimated drifting torque is used as additional compensation over an inner loop control architecture based on robust acceleration control with disturbance observer. The proposed drift compensation algorithm is validated over a series of experiments. The results obtained from these experiments illustrate the improvement made by the proposed controller in positioning performance of mecanum wheeled mobile robots.

Acknowledgment

This study is partially supported by the Internal Research Grant "2017.01.004" of İstanbul Bilgi University.

References

[1] Cho BS, Moon WS, Seo WJ, Baek KR. A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding. Journal of Mechanical Science and Technology 2011; 25 (11): 2907-2917. doi: 10.1007/s12206-011-0805-1

[2] Blyth WA, Barr RWB, Ferdinando RYB. A reduced actuation mecanum wheel platform for pipe inspection. In: IEEE International Conference on Advanced Intelligent Mechatronics (AIM); Banff, Canada; 2016. pp. 419-424.

[3] Kang JW, Kim BS, Chung MJ. Development of omni-directional mobile robots with mecanum wheels assisting the disabled in a factory environment. In: IEEE International Conference on Control, Automation and Systems (ICCAS); Seoul, South Korea; 2008. pp. 2070-2075

[4] Adăscăliţei F, Doroftei I. Practical applications for mobile robots based on mecanum wheels-a systematic survey. The Romanian Review Precision Mechanics, Optics and Mechatronics 2011; 40: 21-29

[5] Han KL, Choi OK, Hwang I, Lee JS, Choi S. Design and control of omni-directional mobile robot for mobile haptic interface. In: IEEE International Conference on Control, Automation and Systems (ICCAS); Seoul, South Korea; 2008. pp. 1290-1295

[6] Qian J, Zi B, Wang D, Ma Y, Zhang, D. The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system. Sensors 2017; 17 (9); 2073-2087. doi: 10.3390/s17092073

[7] Galicki M. Optimal cascaded control of mobile manipulators. Nonlinear Dynamics 2019; 96 (2): 1367-1389. doi: 10.1007/s11071-019-04860-7

[8] Bae JJ, Kang N. Design optimization of a mecanum wheel to reduce vertical vibrations by the consideration of equivalent stiffness. Shock and Vibration 2016; 2016: 1-8. doi: 10.1155/2016/5892784

[9] Alakshendra V, Chiddarwar SS. A robust adaptive control of mecanum wheel mobile robot: simulation and experimental validation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); Daejeon, Korea; 2016. pp. 5606-5611

[10] Chu B. Position compensation algorithm for omnidirectional mobile robots and its experimental evaluation. International Journal of Precision Engineering and Manufacturing 2017; 18 (12); 1755-1762. doi: 10.1007/s12541-017-0204-3

[11] Killpack M, Deyle T, Anderson C, Kemp CC. Visual odometry and control for an omnidirectional mobile robot with a downward-facing camera. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); Taipei, Taiwan; 2010. pp. 139-146

[12] Julliere M, Place H, Bazin E, Radenac JF. A relative localisation system of a mobile robot. Journal of Intelligent and Robotic Systems 1988; 1 (3): 243-257. doi: 10.1007/BF00238768

[13] Hong S. Park S. Minimal-drift heading measurement using a MEMS gyro for indoor mobile robots. Sensors 2008; 8 (11): 7287-7299. doi: 10.3390/s8117287

[14] Coito F. Eleutério A. Valtchev S. Coito F. Tracking a mobile robot position using vision and inertial sensor. In: Doctoral Conference on Computing, Electrical and Industrial Systems; Costa de Caparica, Portugal; 2014. pp. 201-208

[15] Matía F. Jiménez A. Multisensor fusion: an autonomous mobile robot. Journal of Intelligent and Robotic Systems 1998; 22 (2): 129-141. doi: 10.1023/A:1007928600410

[16] Srinivasan K, Gu J. Tracking a mobile robot position using vision and inertial sensor. In: Canadian Conference on Electrical and Computer Engineering (CCECE); Vancouver, Canada; 2007. pp. 1207-1210

[17] Wu C, Tsai C. Localization of an autonomous mobile robot based on ultrasonic sensory information. Journal of Intelligent and Robotic Systems 2001; 30 (3): 267-277. doi: 10.1023/A:1008154910876

[18] Kim J, Woo S, Kim J, Do J, Kim S et al. Inertial navigation system for an automatic guided vehicle with mecanum wheels. International Journal of Precision Engineering and Manufacturing 2012; 13 (3): 379-386. doi: 10.1007/s12541-012-0048-9

[19] Siravuru A, Shah SV, Krishna KM. An optimal wheel-torque control on a compliant modular robot for wheel-slip minimization. Robotica 2017; 35 (2): 463-482. doi: 10.1017/S0263574715000685

[20] Lee IS, Kim JY, Lee JH, Kim JM, Kim SS. Kalman filter-based sensor fusion for improving localization of AGV. Advanced Materials Research 2012; 488: 1818-1822. doi: 10.4028/www.scientific.net/AMR.488-489.1818

[21] Sahoo SR, Chiddarwar SS, Alakshendra V. Intuitive dynamic modeling and flatness-based nonlinear control of a mobile robot. Simulation 2018; 94 (9): 797-820. doi: 10.1177/0037549717741192

[22] Baran EA, Uzunovic T, Sabanovic A. Performance improvement of bilateral control systems using derivative of force. Robotica 20118; 36 (11): 1627-1640. doi: 10.1017/S0263574718000607

[23] Zhang J, Liu X, Xia Y, Zuo Z, Wang Y. Disturbance observer-based integral sliding-mode control for systems with mismatched disturbances. IEEE Transactions on Industrial Electronics 2016; 63 (11): 7040-7048. doi: 10.1109/TIE.2016.2583999

[24] Xu B, Shi Z, Yang C. Composite fuzzy control of a class of uncertain nonlinear systems with disturbance observer. Nonlinear Dynamics 2015; 80 (1-2): 341-351. doi: 10.1007/s11071-014-1872-5

[25] Ohnishi K, Shibata M, Murakami T. Motion control for advanced mechatronics. IEEE/ASME Transactions on Mechatronics 1996; 1 (1): 56-67. doi: 10.1109/3516.491410

[26] Sariyildiz E, Ohnishi, K. Stability and robustness of disturbance-observer-based motion control systems. IEEE Transactions on Industrial Electronics 2015; 62 (1): 414-422. doi: 10.1109/TIE.2014.2327009

[27] Yuan Z, Tian Y, Yin Y, Wang S, Liu J et al. Trajectory tracking control of a four mecanum wheeled mobile platform: an extended state observer-based sliding mode approach. IET Control Theory & Applications 2020; 14 (3): 415-426. doi: 10.1049/iet-cta.2018.6127