Encoder-less position estimation and error correction techniques for miniature mobile robots

Article *in* Turkish Journal of Electrical Engineering and Computer Sciences · January 2013

DOI: 10.3906/elk-1109-65

CITATIONS
21

READS
1,432

2 authors:

Farshad Arvin
The University of Manchester
84 PUBLICATIONS **900** CITATIONS
SEE PROFILE

Masoud Bekravi
Islamic Azad University, Ardabil Branch
8 PUBLICATIONS **115** CITATIONS
SEE PROFILE

Turkish Journal of Electrical Engineering & Computer Sciences

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

Research Article

Turk J Elec Eng & Comp Sci
(2013) 21: 1631 – 1645

doi:10.3906/elk-1109-65

Encoderless position estimation and error correction techniques for miniature mobile robots

Farshad ARVIN,[1,*] **Masoud BEKRAVI**[2]

[1]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[2]Department of Computer Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran

Received: 30.09.2011 • **Accepted:** 15.12.2011 • **Published Online:** 02.10.2013 • **Printed:** 28.10.2013

Abstract: This paper presents an encoderless position estimation technique for miniature-sized mobile robots. Odometry techniques, which are based on the hardware components, are commonly used for calculating the geometric location of mobile robots. Therefore, the robot must be equipped with an appropriate sensor to measure the motion. However, due to the hardware limitations of some robots, employing extra hardware is impossible. On the other hand, in swarm robotic research, which uses a large number of mobile robots, equipping the robots with motion sensors might be costly. In this study, the trajectory of the robot is divided into several small displacements over short spans of time. Therefore, the position of the robot is calculated within a short period, using the speed equations of the robot's wheel. In addition, an error correction function is proposed that estimates the errors of the motion using a current monitoring technique. The experiments illustrate the feasibility of the proposed position estimation and error correction techniques to be used in miniature-sized mobile robots without requiring an additional sensor.

Key words: Position estimation, mobile robot, error correction, odometry

1. Introduction

Applications for autonomous mobile robots for solving human problems have increased rapidly. Various mobile robots are being developed to assist humans in conducting their daily lives. A reliable autonomous mobile system requires the estimation of its accurate position [1]. Miniature-sized mobile robots are frequently used in the study of multirobotics. For the multirobotic application, it is important to estimate the actual position and orientation of the robots continuously [2,3]. In swarm robotics [4–6], which is a subset of ordinary multirobotic research, the position estimation is an important function for performing the collective behaviors.

Odometry is a common technique used to estimate the proportional location of mobile robots using the differential of wheel displacement in a span of time [1]. Aside from the wheel encoder that is widely utilized as the measuring component [7], there are several approaches for measuring the wheel displacement of the mobile robots. Multiple optical flow sensors instead of wheel encoders were used in [8]. To obtain accurate odometry, the proposed technique attempted to detect and eliminate the faulty sensors from the calculations. Experiments were performed in a defined circular path with different radii (80, 90, and 140 mm). A laser sensor was also used for an odometry system in [9]. Hence, the accuracy of the proposed technique was investigated in 3 different paths, including straight forward, L-shaped, and rectangular motions. Moreover, the odometry technique that was proposed in [10] used an optical mouse for measuring the travelled distance of the robot. In general, the

*Correspondence: farshadarvin@yahoo.com

estimated values of the odometry techniques include measuring and calculation errors. Thus, an error correction function assists in calculating the high accuracy coordinates of the mobile robot.

Due to the impressionability of odometry techniques, usually error detection and measuring techniques are combined to correct the estimated coordinates of odometry. There are 2 types of correction factors, systematic and nonsystematic. The first factor is estimable, such as the friction of the robot's wheels. The next factor is unexpected, such as the forces exerted by avoiding or blocking obstacles. Calibration of odometry reduces the systematic errors that occur during experiments. In [11], a calibration technique to improve the accuracy of the odometry was proposed. The effects of the wheel diameter and tread were investigated in the calibration of the systematic errors and the reduction of the nonsystematic errors. Several techniques to reduce the systematic errors were presented in [12]. Based on the proposed techniques, 2 parameters, the dissimilarity of the wheel diameters and the uncertainty of the wheelbase, were investigated. A neural network approach was used to implement the calibration of the odometry [9]. In addition, the calibration of the optical odometry was presented in [8,10]. The error correction of a mobile robot with a geographic information system was performed in [13]. A real-time localization of a low-cost mobile robot was proposed in [14], which used a limited ultrasonic data set to correct the odometry errors. Visual odometry and error correction could be an alternative technique in the mobile robots' localization; however, it requires the capturing of media and image processing using complex calculations [15,16]. Similarly, the speckle velocimetry technique was presented to be used in the Mars exploration rover in [17].

Generally, a wheel's encoder or other types of sensors are employed for calculating the wheel's locomotion. However, they require a mechanical configuration and are prone to wear and tear. In addition, placing extra sensors will not be a simple task in a large population. Therefore, sensorless techniques with minimal changes in the mechanical structure of the robot are efficient approaches if their reliability is confirmed [18]. The current of a motor is a suitable benchmark when the motor does not have a normal behavior. It relies on the applied voltage and the motor's manufacturing characteristics [19]. The use of the motor current was previously proposed in [20] to estimate the joint torque of a humanoid robot's arm. It utilized the motor current to detect and measure the applied force without using any hardware components. In [21], a current-based wheel slippage detection technique was proposed, where a current compensation module was developed to reduce the wheel's slippage.

In this paper, an encoderless positioning technique is proposed that uses a pulse-width modulation (PWM) duty cycle to estimate the actual position of mobile robot. It uses the displacement of the robot's wheels over short spans of time. The trajectory of the robot is divided into several short time intervals and the calculations are applied continuously. This type of positioning does not require any knowledge about the trajectory and target. The position estimation calculation usually includes the error in the estimation. Hence, an error correction function is required to recognize the calculation errors. In this work, a current-based error correction method is proposed that recognizes and estimates the errors of the positioning function using the motor's current and voltage. The autonomous miniature robot (AMiR) is an open-hardware miniature-sized mobile robot that was developed for swarm robotics [22]. In this study, the AMiR with 3 different configurations is employed as the platform robot for implementing the proposed technique.

2. Mobile robot architecture

2.1. Mechanical configuration and platform

To obtain a reliable and accurate positioning, a precise mathematical model considering the mechanical design of the wheels and their configuration is required. Mobile robots are developed with different classifications of the wheels, such as differential [23,24], car-type [11], omnidirectional [2], and synchronous drives [25]. The differential drive is the common wheel configuration in mobile robots due to its modeling simplicity and practicality. In this technique, 2 motors are employed to provide motion and 1 or 2 caster wheels are used for balancing the robot's structure.

The AMiR is an autonomous mobile robot that was designed for swarm applications [22], as shown in Figure 1a. The hardware diagram of the AMiR is shown in Figure 1b. The AMiR uses 2 DC motors with an internal gearbox that is controlled with the PWM technique [26]. The AMiR employs 6 infrared emitter proximity sensors for obstacle avoidance and communication [27]. It is a low-cost and open-hardware platform that is suitable for education and research in swarm robotics. The AMiR has been utilized in several swarm robotic research studies, such as in the honeybee aggregation presented in [6]. It has a small form factor and preliminary tests show the reliability of the mechanical and sensory system modules for swarm applications. An ATMEGA-168 microcontroller is deployed as the main processor, which allows for controlling of the motors using individual PWM channels. The battery level of the robot V_{bat} is measured continuously by an analog-to-digital converter (ADC) channel of the processor.

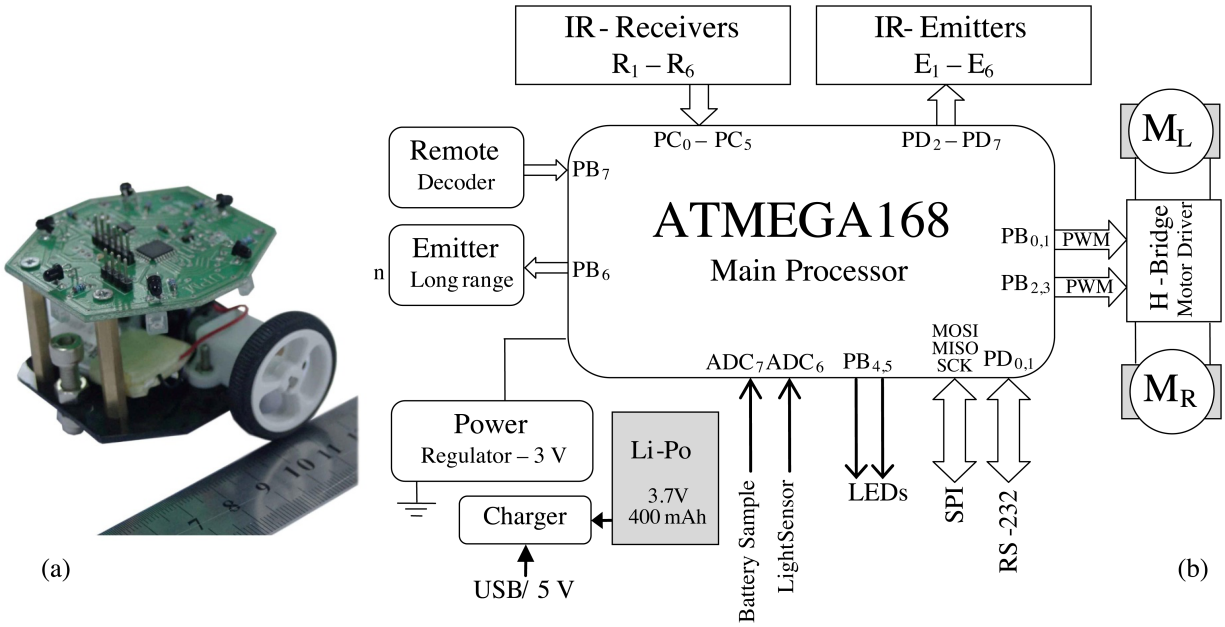

Figure 1. a) Autonomous miniature robot (AMiR) and b) hardware diagram of the AMiR.

In order to test the accuracy of the proposed positioning technique, the AMiR is configured with 3 different DC motors. The difference is due to the gearbox ratios, which are 100:1 for AMiR-1, 150:1 for AMiR-2, and 298:1 for AMiR-3. All of the robots use a similar processing unit with 3 different clock sources (1, 4, and 8 MHz). The different frequencies of the processing units evaluate the accuracy of the method in order to calculate the position estimation with the different processing speeds of the controller. Table 1 shows the characteristics of the AMiRs.

Table 1. The AMiRs' characteristics.

Robot platform	Processor clock source	Gear ratio	No-load speed	No-load current	Stall current	Working voltage	Armature resistance
AMiR-1	8 MHz	100:1	132 RPM	40 mA	360 mA	2–9 V	20 Ω
AMiR-2	4 MHz	150:1	85 RPM	40 mA	360 mA	2–9 V	20 Ω
AMiR-3	1 MHz	298:1	45 RPM	38 mA	360 mA	2–9 V	20 Ω

2.2. Motion control using PWM

The following relationship allows for the controlling of the DC motor speed as a function of the physical and electrical parameters [19]:

$$N = \frac{v_m - i_m R_m}{K\varphi}. \tag{1}$$

N is the speed of the motor, v_m is the applied voltage, i_m is the current of the motor, R_m is the armature resistance, K is the constant value depending on the construction of the motor, and φ is the total flux of the machine. The parameters R_m, K, and φ are obtained from the motor's manufacturer. The applied voltage is a variable that the robot uses for reaching different speeds. PWM is a popular technique for reducing voltage. It is a flexible, reliable, and fast solution, while at the same time, it reduces power consumption. The average voltage of the PWM technique is usually a fraction of the source voltage, depending on the duty cycle:

$$v_{avg} = \frac{1}{T}\int_0^T f(t)dt = \frac{1}{T}\left(\int_0^{p_w T} v_{\max} dt + \int_{p_w T}^{T} v_{\min} dt\right) = p_w v_{\max} + (1-p_w)v_{\min}, \tag{2}$$

where $f(t)$ is the pulse wave with a period of T, v_{min} and v_{max} are the minimum and maximum voltages of the wave, and p_w is the duty cycle $0 \leq p_w \leq 1$. v_{max} in the AMiR is the voltage of the battery and v_{min} is 0 V. If $v_{max} = V_{bat}$, then v_{avg} will not be a constant value in the constant duty cycle due to the battery discharge during a task. To produce a constant speed, the battery discharge could be compensated with the enlarging of p_w. Hence, the processor changes p_w to provide a similar speed at different battery levels [26]. The average voltage is calculated by the following formula:

$$v_{avg} = p_w . V_{bat}. \tag{3}$$

The maximum spinning velocity of the shaft can be estimated by the following proposed model of the robot's motor:

$$N_{\max} = \alpha V_{bat} + \beta, \tag{4}$$

where N is the spinning velocity of the wheels; V_{bat} is the voltage of the battery, which is measured by the microcontroller (ADC-7, Figure 1b); and α and β are 2 coefficients that are dependent on the motor's characteristics and mechanical design. Suitable values of α and β are extracted using the empirical experiments. The proposed model allows for fast estimation of the kinematic calculations and reduces the burden of the processor. According to Eq. (5), the speed of the robot is a function of the shaft spinning, N, and diameter of the wheels, d_w. Hence, the speed of the robot is calculated by the following formula:

$$v_{\max} = \pi d_w N_{\max}. \tag{5}$$

The required torque at the motor's shaft, τ_m, can be calculated by:

$$\tau_m = \tau_{noload} + \tau_{movement}, \quad \tau_{movement} = \frac{\tau_w}{n}, \tag{6}$$

where τ_w is the required torque at the robot's wheels and n is the gearbox ratio. As the gearboxes ratios are high, this results in a small torque for movement ($\tau_{movement} \approx 0$). As a result, the acceleration of the motor is similar to that of a no-load condition and the speed settlement takes place after a few milliseconds.

3. Position estimation and error correction

The position of the robot is defined with 3 variables, as represented in Eq. (7), where x and y are the coordinates of the robot and θ is the robot's orientation. These values are calculated continuously during a motion.

$$P_{robot}(x, y, \theta) \tag{7}$$

3.1. Encoderless positioning

This technique is a real-time approach in the position estimation that uses PWM set points for calculating the displacements. The positioning function is independent on the motion types and it is calculated over short intervals of time, without any knowledge of the next target. It is suitable for obtaining real-time position variables for complex trajectories and random motions. Figure 2 illustrates the displacement of a miniature mobile robot at a specified time interval (δt).

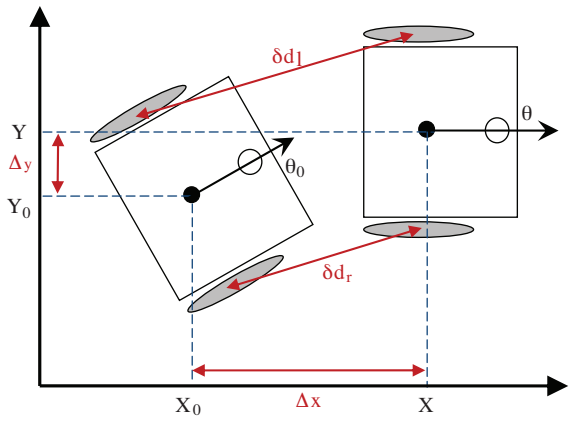

Figure 2. Movement of the mobile robot in a time span (δt).

Assume that in a short time interval, the mobile robot changes its position from (x_0, y_0, θ_0) to (x, y, θ). First, the new orientation of the robot, which is dependent on the differential of the distance travelled by each wheel, must be estimated. The displacements of the wheels are extracted based on the PWM channels' set point (duty cycle). There are 2 different registers that control the left and right wheels of the robot. The values between 0 and 511 are defined for forward motion and the values between 512 and 1023 are for reverse motion.

For estimating the displacement of wheels, the speed of wheels (φ) must be calculated, which is directly dependent on the PWM set point and maximum speed of the robot:

$$\phi_{l,r} = P_{l,r}.v_{\max}, \tag{8}$$

where v_{max} is the maximum speed of the motors with respect to the battery capacity and the motor characteristics, and P is the pulse-width of the left and right motors ($\varphi = P\pi d_w(\alpha V_{bat} + \beta)$). Hence, the displacement of the wheels (δd) for a span of time (δt) is calculated by the following formula:

$$\delta d_{l,r} = \phi_{l,r}.\delta t. \tag{9}$$

The time span has a key role in resolution of the calculation. To obtain an accurate position estimation, an appropriate value for δt must be assigned. The resolution decreases if longer values are selected and higher resolution might not be handled by the processor. After estimation of the wheel displacement, the new orientation of the robot can be estimated by:

$$\theta = \theta_0 + \frac{\delta d_r - \delta d_l}{w}, \tag{10}$$

where θ is the new orientation, θ_0 is the previous orientation, δd is the travelled distance of the wheels (left and right), and w is the wheelbase of the robot. The displacement of the robot for an interval is the average of the traveled distances by both of the wheels. Therefore, the proportional position of the mobile robot is calculated by:

$$x = x_0 + \frac{\delta d_r + \delta d_l}{2}\cos\theta, \quad y = y_0 + \frac{\delta d_r + \delta d_l}{2}\sin\theta. \tag{11}$$

The kinematic modeling of the given robot is defined as the following model [28]:

$$\zeta = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = f(\phi_l, \phi_r, \delta t). \tag{12}$$

According to Eq. (4), the motor's speed (φ_l, φ_r) is a function of the battery level, which must be captured for calculating the maximum speed of the robot (v_{max}). In real-world experiments, several types of unavoidable deterrents occur, such as friction and collision. These deterrents must be recognized and applied to the kinematic model.

3.2. Error modeling of motion

An encoderless positioning technique, without requiring any changes in the hardware configuration, has been proposed. The error correction in this technique requires some feedback from the motors to detect the errors in the calculations. The current of the motor is a useful parameter for recognizing unsystematic errors. It has a relationship with the motor's voltage and the motor's internal resistor. The DC motor follows 2 linear relationships between its voltage and current for the motion and stop states. The stop state means that the applied voltage is not enough to actuate the motor.

$$i_m = \begin{cases} a_M v_m + b_M & v_m \geq Thr \quad \text{(Motion)} \\ a_M v_m + b_S & v_m < Thr \quad \text{(Stop)} \end{cases} \tag{13}$$

Here, i_m is the motor current, v_m is the motor voltage, a_M and b_M are the coefficients of the motion state, a_S and b_S are the coefficients of the stop state, and Thr is the threshold voltage value for the starting of the motion that is calculated by $-\beta/\alpha$ from Eq. (4). These parameters will be extracted using the empirical experiments.

Normal motion must follow Eq. (13). After measuring the current and voltage of the motor, the speed of the wheels is multiplied by a correction coefficient:

$$\phi = f_c.\phi_0, \tag{14}$$

where f_c is the correction value for the positioning function that covers the systematic and nonsystematic errors. The output of the function is between 0 and 1, where the 0 value shows that the robot is completely stalled and 1 shows that the robot has its normal motion without any problems. As shown in Eq. (15), f_c is a function of the current, voltage, applied duty cycle, and battery level.

$$f_c(i_m, v_m, p_m, V_{bat}) \tag{15}$$

In addition, f_c is applied to estimate the required time for rotation, t_r, straight motion, t_s, and turning, t_t, of the robot. The estimated times for the motion types before the force is t_0:

$$t_{r,s,t} = \frac{t_{0r,s,t}}{f_c}. \tag{16}$$

This function is used twice, for the left, f_{cl}, and right, f_{cr}, wheels, using the measured values by the individual ADC channels of the main controller. Hence, the velocities of the motors are calculated by:

$$\phi_l = f_{cl}\phi_{0l}, \quad \phi_r = f_r\phi_{0r}. \tag{17}$$

In order to extract the actual speed of the motor, the induced voltage, v_i, must be extracted [19]:

$$v_i = v_m - i_m R_m, \tag{18}$$

where v_m is the motor voltage, i_m is the motor current, and R_m is the armature resistance. The actual speed of the motor has a linear relationship with v_i. Therefore, a correction on the calculations is applied in Eq. (9), with the new wheel speeds. The robot's kinematic model is improved after extracting the f_c value of Eq. (19) to the new model, as shown in Eq. (20).

$$f_c = \frac{v_i}{p_w V_{bat}} \Rightarrow f_c = \frac{v_m - i_m R_m}{p_w V_{bat}} \tag{19}$$

$$\zeta = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = f(\phi_l, \phi_r, \delta t, f_{cl}, f_{cr}) \tag{20}$$

Figure 3 illustrates the block diagram of the proposed system, including the position estimation using the displacement functions and the error correction using feedback from the motors.

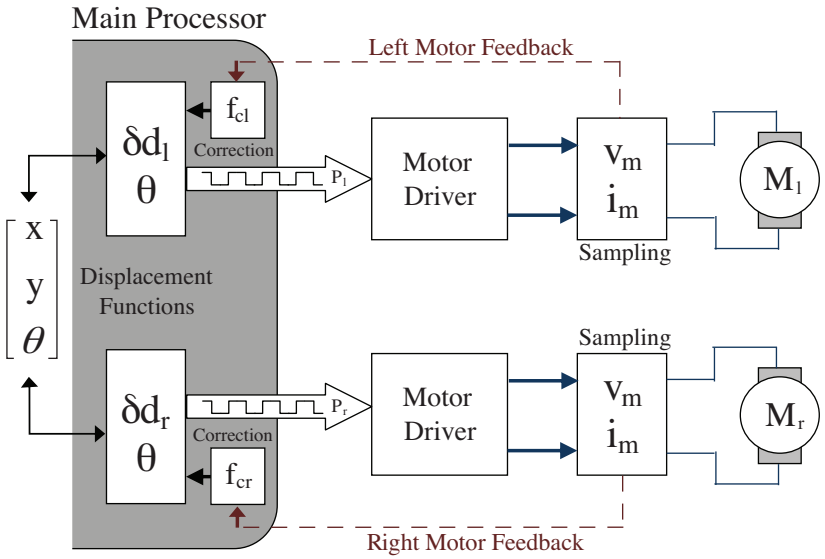

Figure 3. Block diagram of the position estimation and error correction techniques.

4. Preliminary experiments

A preliminary experiment is used to extract the model variables that are described by Eq. (4). In this regard, the robot's velocity as a function of the applied voltage is captured (Figure 4). The speed of the DC motor has a linear relationship with the applied voltage [19]; therefore, a coefficient value that influences the fundamental kinematic calculation of the robot can be extracted.

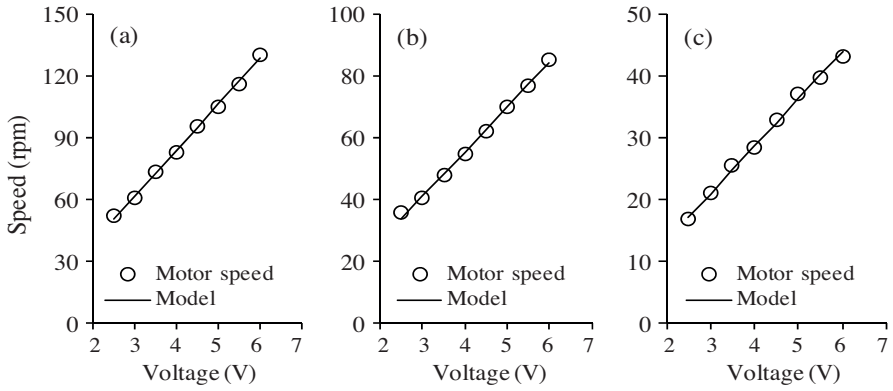

Figure 4. Linear relationship between the voltage of the battery and the speed of the motor in different platforms: a) AMiR-1, b) AMiR-2, and c) AMiR-3.

Table 2 illustrates the extracted model parameters for the employed platforms. The value of $-\beta/\alpha$ shows that the DC motors start at around 0.2 V.

5. Results and discussion

In this section, results of the several performed experiments in order to demonstrate the feasibility of the proposed technique are discussed. For tracking the robot's motion during the experiments, SwisTrack-4.0,

which is an open-source tracking software, was utilized [29]. It has been designed for tracking robots, animals, and objects of a given video.

Table 2. Extracted model parameters for the platforms.

Robot platform	α(RPM/V)	β (RPM)	R^2	$-\beta/\alpha$(V)
AMiR-1	22.32	–5.53	0.9979	0.247
AMiR-2	14.68	–3.51	0.9973	0.239
AMiR-3	7.67	–2.19	0.9997	0.285

5.1. Encoderless positioning

From Eqs. (8) and (9), 2 effective components in this technique are v_{max} and δt, without respect to the motion and configuration of the environment. From $v_{max} = V_{bat}$ and the extracted parameters for Eq. (4), the change in the battery level does not influence the robot's speed; hence, the robot has a constant speed at different battery levels. Regarding the study on effects of δt, an experiment was performed to find the best time intervals for different speed activities of the robots.

Figure 5 shows the resolution of positioning at different time intervals. The robots change their orientations to avoid random obstacles during their motion. The accuracy was measured with various speeds of the robots. The results show that $\delta t = 100$ ms is the suitable time interval in both quiet and complex arenas. In the short time spans, the calculation error increased due to the limitation of the decimal-digit's computation by the microcontroller. In the long time intervals, the robots cannot detect the rotations caused by obstacle avoidance. An average accuracy of about 98%, regardless of the platform, was achieved in the quiet arena. Although the complex arena causes successive rotations to avoid collisions, an accuracy of 96% can be achieved by this technique.

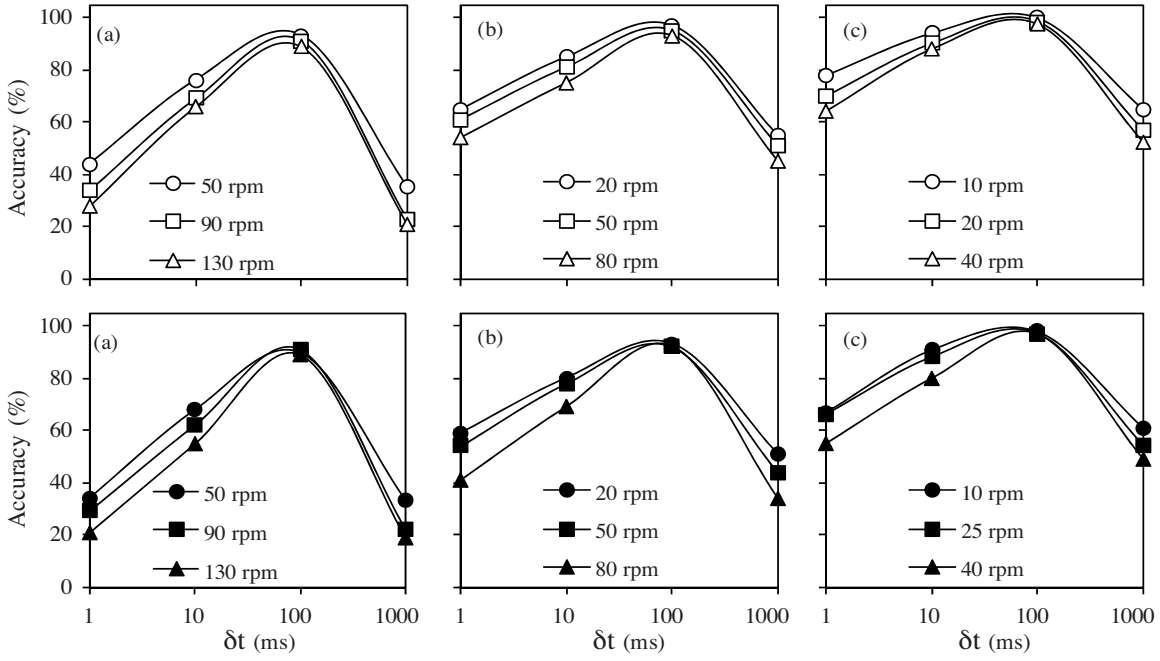

Figure 5. Accuracy of the positioning as a function of the time interval (δt) for a) AMiR-1, b) AMiR-2, and c) AMiR-3 robot platforms (empty symbols indicate a quiet arena and filled symbols indicate a complex arena).

A dead-reckoning experiment was performed in which the robots should continuously move in a specified rectangular path (80 × 50 cm). The robots transmit the estimated values of their locations using a serial RS-232 cable. They move at their maximum speeds and complete the laps 20 times. Figure 6 illustrates differences between the real and estimated positions for the robots for the long-term motion. $X_e = X_i - x_i$ and $Y_e = Y_i - y_i$ are the instant errors of system for the X and Y axes, respectively. X and Y are the predicted values, x and y are the real positions in the arena, and i is the sample's number. According to the curves, the system shows the controlled behavior with an independent fluctuation due to the processor and hardware limitations. The results demonstrate that the system does not have a significant error and that the robots followed the defined path with slight slippages.

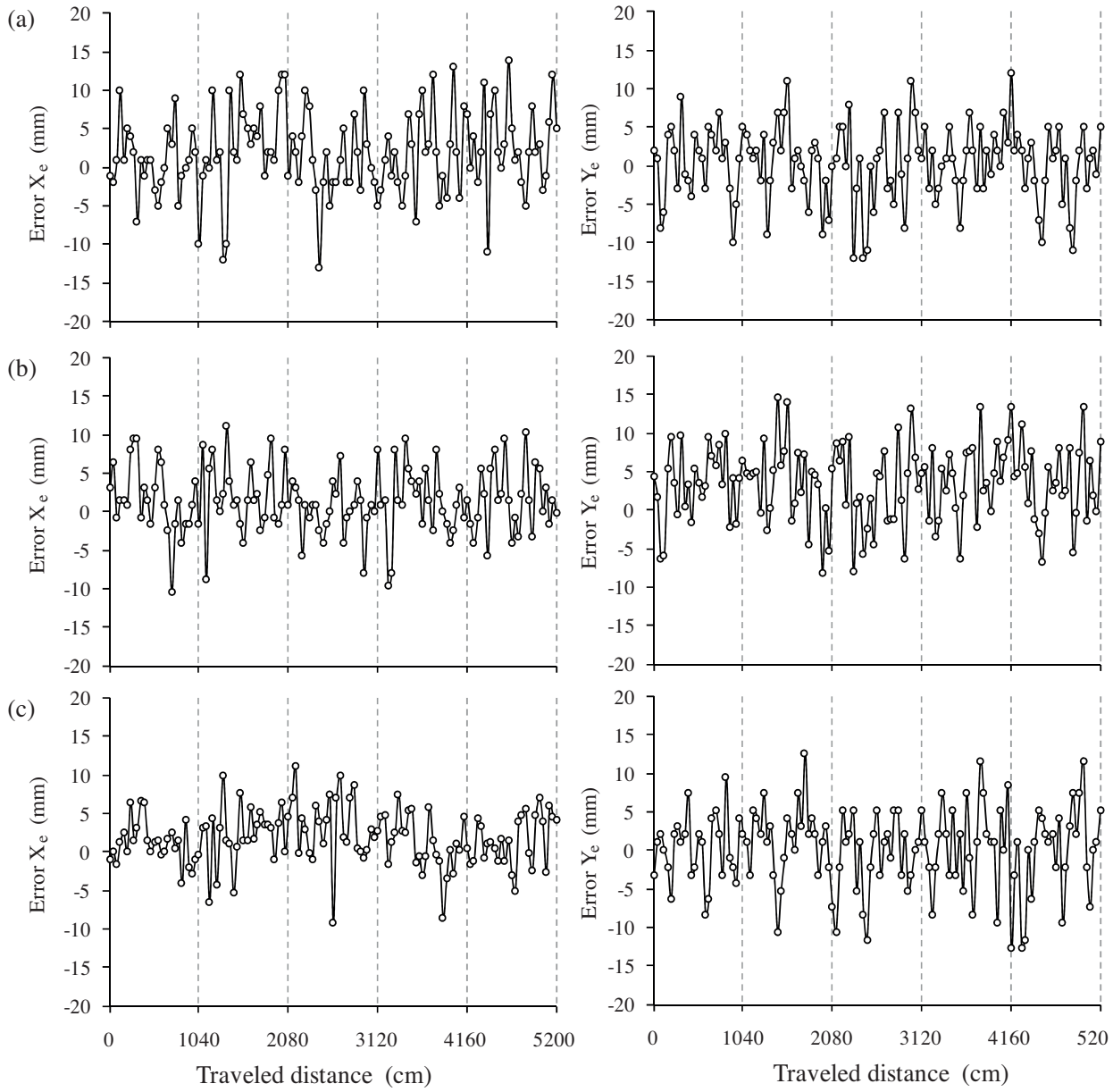

Figure 6. Instant difference between the estimated and real positions for the motion of the robot in the specified rectangle (80 × 50 cm) for a) AMiR-1, b) AMiR-2, and c) AMiR-3.

Table 3 illustrates the average error of the proposed technique in a rectangular trajectory experiment that was calculated by $\frac{1}{n}\sum\sqrt{(X_i - x_i)^2}$ in the case of the X axis, where n is the number of samples. Similarly, the average error for the Y axis was calculated using the previous formula. Each experiment was repeated 10 times and the mean values were extracted. The average error of approximately 0.5 mm demonstrates that the proposed technique is amenable for use as an encoderless positioning technique, without respect to the platform robots.

Table 3. Average error for the trajectory in the specified rectangle (80 × 50 cm) for 20 laps.

Robot platform	Maximum speed (cm/s)	Average error Y-axis (mm)	Average error Y-axis (mm)
AMiR-1	21.8	0.563 ± 0.06	0.528 ± 0.05
AMiR-2	13.4	0.396 ± 0.05	0.385 ± 0.04
AMiR-3	6.7	0.312 ± 0.03	0.379 ± 0.03

To analyze the efficiency of the proposed technique, another experiment was performed, where the robots moved aimlessly in quiet (only arena walls) and complex (more obstacles) arenas. The motions of the robots were captured for a duration of 5 min. The average errors of the random motions in all of the platforms showed the insignificant errors. The average error increased in the complex arena; however, the average error rate of 2 mm is an appropriate value for the mobile robot's positioning approach. Table 4 reveals the average error of the system for different platforms.

Table 4. Average error for the random trajectory of the robots.

Robot platform	Quiet arena		Complex arena	
	Average error X-axis (mm)	Average error Y-axis (mm)	Average error X-axis (mm)	Average error Y-axis (mm)
AMiR-1	0.487 ± 0.13	0.557 ± 0.11	2.675 ± 0.86	2.439 ± 0.87
AMiR-2	0.331 ± 0.10	0.396 ± 0.08	1.863 ± 0.58	2.129 ± 0.65
AMiR-3	0.322 ± 0.11	0.392 ± 0.10	1.282 ± 0.54	1.728 ± 0.67

Although the proposed technique did not use additional measuring hardware, its accuracy was close to the results obtained by the encoder-based odometry reported in [1,23,25]. The odometry using an optical mouse [10] that moved slowly (1.5 mm/s) illustrated high accuracy; however, its accuracy was related to the travelled distance, surface, and motion types. In a similar encoderless odometry technique that was implemented by the Jasmin-III microrobot [24], the reported accuracy was about 6% and 11% for the displacement and rotation, respectively. In addition, the differences shown in Figure 6 were not gained during the experiment and the robot moved with a slight fluctuation; hence, the proposed encoderless positioning is a controlled distance-independent function.

5.2. Current-based error correction

The utilized motors use a similar armature with different gearbox ratios. Hence, the behavior of the motors in different situations almost follows a similar relationship. Figure 7 shows the behavior of the motors with different PWM values.

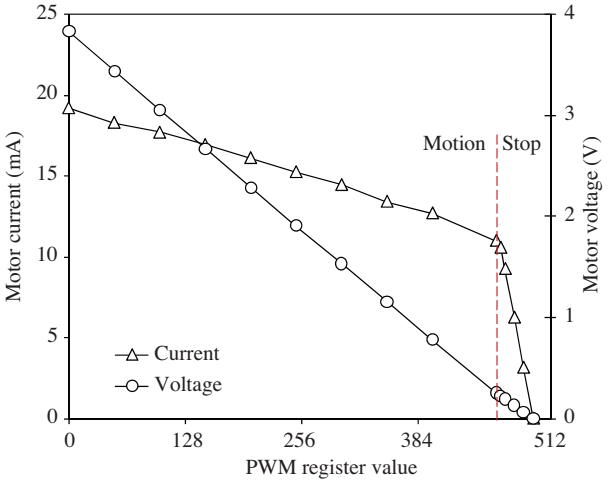

Figure 7. Current and voltage of the motor as the function of PWM values.

Therefore, the motor's behavior is modeled with 2 different equations for motion and stop. Under working conditions, the voltage and current of the motors are functions of the PWM value, P_w:

$$v_m = -76 \times 10^{-4} P_w + 3.82, \quad i_m = -17 \times 10^{-3} P_w + 19.35. \tag{21}$$

When the motor is stopped, its current has a different relationship than when under working conditions, as shown in Eq. (22).

$$i_m = -0.3 P_w + 151.54 \tag{22}$$

The relationship between the voltage and current of the motor is shown in Figure 8. It shows a linear relationship between these 2 values in motion and stop modes. Hence, the effective parameters of the motor (a_M, b_M for motion and a_S, b_S for stop) are extracted from Eq. (13). The motion of the robot is without any external forces when the v_m and i_m values are fitted into Eq. (23).

$$i_m = \begin{cases} 2.23 v_m + 10.84 & v_m \geq 0.2 \quad \text{(Motion)} \\ 48.v_m & v_m < 0.2 \quad \text{(Stop)} \end{cases} \tag{23}$$

Figure 9 shows the relationship between the voltage drop (depends on the load) and the motor current in the specified PWM values (10, 40, 100, and 200).

Therefore, when the voltage and current of the motor do not follow Eq. (23), the error correction function must be called on to estimate the actual speed of the motor. The extraction of the induced voltage, as shown in Eq. (18), is the first step for estimating the real speed of the robot. Afterward, the actual velocity of the motor is estimated by Eq. (14).

Next, an experiment was performed to show the feasibility of f_c as an error correction function. In this experiment, the motion of the robot in a rectangular path (80 $\times$ 50 cm) with an additional uncontrolled load was captured. In the first step, f_c was not applied, and the average error of the robot and the estimated positions are shown in Figure 10a. The instant error for the ith sample was calculated based on $E_{ix} = \sqrt{(X_i - x_i)^2}$ and $E_{iy} = \sqrt{(Y_i - y_i)^2}$, and the average error was estimated using $\overline{E_i} = \frac{E_{ix} + E_{iy}}{2}$, where X_i and Y_i are the expected positions of the robot in the rectangular path and x_i and y_i indicate the calculated and real positions of the

robot. As shown in Figure 10a, before applying f_c there were big differences (about 25 cm) between the estimated values and the trajectory of the robot.

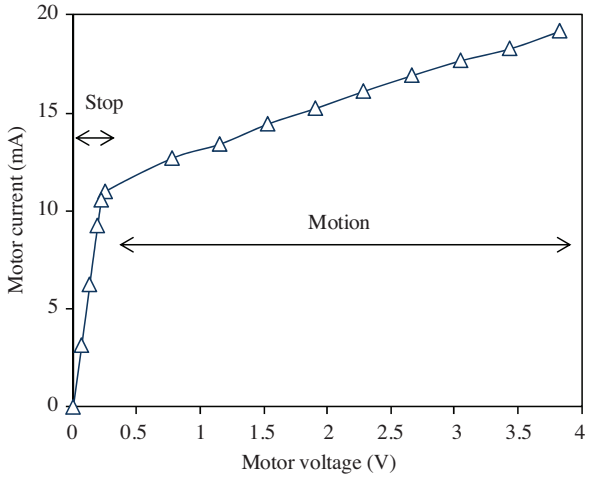

Figure 8. Relationship between voltage of the motor and the current.

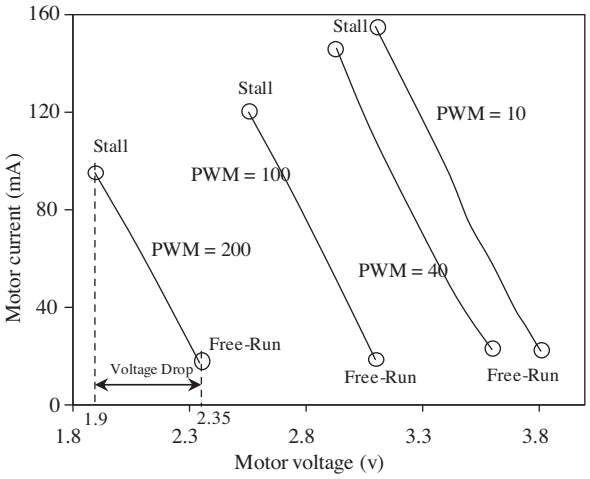

Figure 9. Current and voltage of the motor with an extra load.

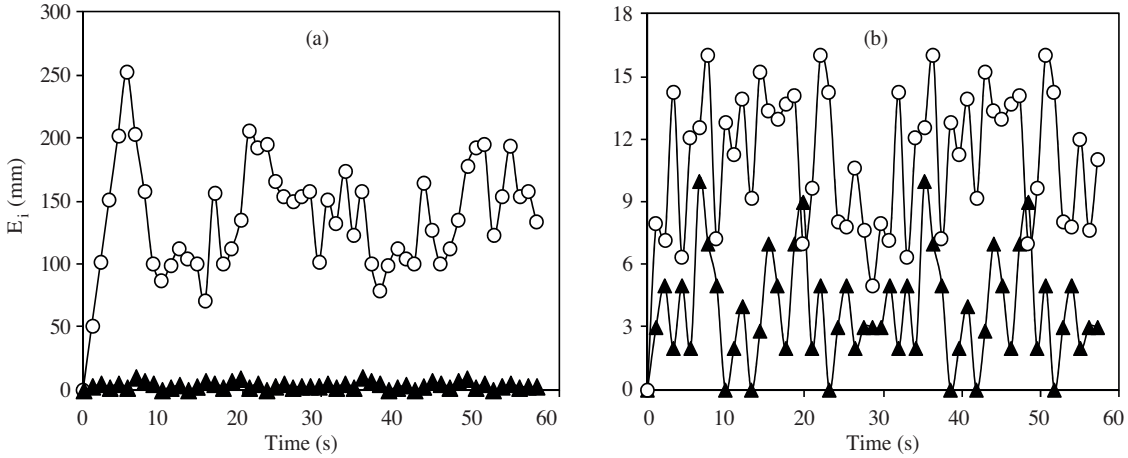

Figure 10. Instant error of the robot trajectory in a rectangular path. Triangle indicates estimated position and circle indicates real position of the robot a) before and b) after applying the f_c correction function.

Next, an experiment was performed using f_c. The average error of the estimated position and the robot motion with respect to the rectangular path are shown in Figure 10b. In this process, a proper f_c was calculated by Eq. (19), and it was applied as the error correction of the positioning in Eq. (14) and the correction of the required time of the different motions in Eq. (16). The results show the improved error correction for the robot under an extra uncontrolled load with an average accuracy of approximately 15 mm.

Finally, the random motion experiments of the robot in 2 arenas (quiet and complex) with extra load were performed. The trajectories of the robots were captured for a duration of 5 min. Average accuracies of approximately 13 mm and 16 mm for the quiet and complex arenas were obtained, respectively. Table 5 illustrates the average error for different types of motion in the X and Y axes. The results showed that the

accuracies for different robots are almost similar, but with slight differences. The overall accuracies of the error correction for the odometry techniques that utilized different types of sensors are shown in Table 6.

Table 5. Average error for the random trajectory of the robots with an extra load.

Robot platform	Quiet arena		Complex arena	
	Average error X-axis (mm)	Average error Y-axis (mm)	Average error X-axis (mm)	Average error Y-axis (mm)
AMiR-1	14.23 ± 2.54	14.93 ± 2.32	18.66 ± 2.91	17.29 ± 2.56
AMiR-2	12.88 ± 2.14	13.22 ± 2.08	16.30 ± 2.87	15.33 ± 2.48
AMiR-3	10.73 ± 1.67	11.83 ± 1.81	15.21 ± 2.39	15.64 ± 2.25

Table 6. Comparison of the overall accuracy from the literature.

Odometry method	Type of robot	Overall accuracy
Wheel's encoder [23]	Two-wheel differential drive	3 mm
Ultrasonic sensor [14]	Two-wheel differential drive	4%
Optical sensor [10]	Two- and 4-wheel differential drives	0.8 mm (single sensor) 0.2 mm (average over 10 sensors)
Laser sensor [9]	Two-wheel differential drive	29 mm

Despite the fact that the proposed technique does not use any encoder or hardware components, it showed reliable positioning with suitable accuracy, close to the odometry techniques that were implemented using various sensors.

6. Conclusion

In this paper, encoderless positioning and error correction techniques were proposed. The positioning technique estimated the proportional location of the robot with displacement of wheels in time spans, without any respect to the motion types and environment. The effect of the time span on the resolution of the positioning was tested and a suitable value of 100 ms was selected. A suitable accuracy with an average error of 0.5 mm for the quiet and 2 mm for the complex arenas was obtained. Additional force causes estimation errors; hence, an error correction function was developed for estimating the errors in the positioning calculations. This function uses the current and voltage of the motor to detect any force on the wheels. The final experiment that was set up with an extra unbalanced load demonstrated the feasibility of the correction function in nonsystematic errors. An appropriate accuracy with average error of 13 mm was obtained after applying the correction function.

References

[1] A. Martinelli, N. Tomatis, R. Siegwart, "Simultaneous localization and odometry self-calibration for mobile robot", Autonomous Robots, Vol. 22, pp. 75–85, 2007.

[2] J.A. Batlle, A. Barjau, "Holonomy in mobile robots", Robotics and Autonomous Systems, Vol. 57, pp. 433–440, 2009.

[3] L.M. Ortega, A.J. Rueda, F.R. Feito, "A solution to the path planning problem using angle preprocessing", Robotics and Autonomous Systems, Vol. 58, pp. 27–36, 2010.

[4] E. Şahin, "Swarm robotics: from sources of inspiration to domains of application", Lecture Notes in Computer Science, Vol. 3342, pp. 10–20, 2005.

[5] O. Soysal, E. Bahçeci, E. Şahin, "Aggregation in swarm robotic systems: evolution and probabilistic control", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 15, pp. 199–225, 2007.

[6] F. Arvin, K. Samsudin, A.R. Ramli, M. Bekravi, "Imitation of honeybee aggregation with collective behavior of swarm robots", International Journal of Computational Intelligence Systems, Vol. 4, pp. 739–748, 2011.

[7] G. Antonelli, S. Chiaverini, G. Fusco, "A calibration method for odometry of mobile robots based on the least-squares technique: theory and experimental validation", IEEE Transactions on Robotics, Vol. 21, pp. 994–1004, 2005.

[8] J.S. Hu, Y.J. Chang, Y.L. Hsu, "Calibration and on-line data selection of multiple optical flow sensors for odometry applications", Sensors and Actuators A, Vol. 149, pp. 74–80, 2009.

[9] H. Xu, J. Collins, "Estimating the odometry error of a mobile robot by neural networks", International Conference on Machine Learning and Applications, pp. 378–385, 2009.

[10] J. Palacin, I. Valganon, R. Pernia, "The optical mouse for indoor mobile robot odometry measurement", Sensors and Actuators A, Vol. 126, pp. 141–147, 2006.

[11] K. Lee, W. Chung, K. Yoo, "Kinematic parameter calibration of a car-like mobile robot to improve odometry accuracy", Mechatronics, Vol. 20, pp. 582–595, 2010.

[12] S. Bergbreiter, K.S.J. Pister, "CotsBots: an off-the-shelf platform for distributed robotics", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1632–1637, 2003.

[13] J.M. Mirats Tur, C. Zinggerling, A. Corominas Murtra, "Geographical information systems for map based navigation in urban environments", Robotics and Autonomous Systems, Vol. 57, pp. 922–930, 2009.

[14] P. Hoppenot, E. Colle, "Real-time localisation of a low-cost mobile robot with poor ultrasonic data", Control Engineering Practice, Vol. 6, pp. 925–934, 1998.

[15] A.I. Comport, E. Malis, P. Rives, "Real-time quadrifocal visual odometry", International Journal of Robotics Research, Vol. 29, pp. 245–266, 2010.

[16] C. Sagues, J.J. Guerrero, "Visual correction for mobile robot homing", Robotics and Autonomous Systems, Vol. 50, pp. 41–49, 2005.

[17] T.O.H. Charrett, L. Waugh, R.P. Tatam, "Speckle velocimetry for high accuracy odometry for a Mars exploration rover", Measurement Science and Technology, Vol. 21, pp. 1–12, 2010.

[18] I. Kahalil, E.D. Kunt, A. Şabanoviç, "A novel algorithm for sensorless motion control, parameter identification and position estimation", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 18, pp. 799–810, 2010.

[19] S.J. Chapman, Electric Machinery Fundamentals, New York, McGraw-Hill Higher Education, 2005.

[20] W. Lee, Y. Bang, K. Lee, B. Shin, J.K. Paik, I. Kim, "Motion teaching method for complex robot links using motor current", International Journal of Control, Automation and Systems, Vol. 8, pp. 1072–1081, 2010.

[21] L. Ojeda, D. Cruz, G. Reina, J. Borenstein, "Current-based slippage detection and odometry correction for mobile robots and planetary rovers", IEEE Transactions on Robotics, Vol. 22, pp. 366–378, 2006.

[22] F. Arvin, K. Samsudin, A.R. Ramli, "Development of a miniature robot for swarm robotic application", International Journal of Computer and Electrical Engineering, Vol. 1, pp. 452–459, 2009.

[23] J. Borenstein, L. Feng, "Measurement and correction of systematic odometry errors in mobile robots", IEEE Transactions on Robotics and Automation, Vol. 12, pp. 869–880, 1996.

[24] S. Kernbach, R. Thenius, O. Kernbach, T. Schmickl, "Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system", Adaptive Behavior: Animals, Animats, Software Agents, Robots, Adaptive Systems, Vol. 17, pp. 237–259, 2009.

[25] A. Martinelli, "The odometry error of a mobile robot with a synchronous drive system", IEEE Transactions on Robotics and Automation, Vol. 18, pp. 399–405, 2002.

[26] F. Arvin, K. Samsudin, M.A. Nasseri, "Design of a differential-drive wheeled robot controller with pulse-width modulation", Innovative Technologies in Intelligent Systems and Industrial Applications, pp. 143–147, 2009.

[27] F. Arvin, K. Samsudin, A.R. Ramli, "Development of IR-based short-range communication techniques for swarm robot applications", Advances in Electrical and Computer Engineering, Vol. 10, pp. 61–68, 2010.

[28] R. Siegwart, I.R. Nourbakhsh, Introduction to Autonomous Mobile Robots, Cambridge, MIT Press, pp. 47–82, 2004.

[29] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, A. Martinoli, "SwisTrack-a flexible open source tracking software for multi-agent systems", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4004–4010, 2008.