Adaptive Fuzzy-PD Tracking Controller for Optimal Visual-Servoing of Wheeled Mobile Robots

Omer Saleem*, Humble Hassan*, Awais Khan*, Usman Javaid**

** Electrical Engineering Department, National University of Computer and Emerging Sciences, Lahore, Pakistan (e-mail: omer.saleem, humble.hassan, awais.khan@nu.edu.pk)*
*** Computer Science Department, National University of Computer and Emerging Sciences, Lahore, Pakistan (e-mail: l145412@lhr.nu.edu.pk)*

Abstract: This paper presents a framework for the development of a motion controller to optimize the tracking control of position and orientation of a differentially-driven wheeled mobile robot (WMR) in indoor environments using the visual feedback provided by an overhead camera. The visual information is processed to generate a collision-free trajectory for the WMR towards its destination. While traversing the planned trajectory, the extent of WMRs deviation in actual and desired posture is continuously checked via the visual feedback and motor-encoders. Two-layered control architecture is proposed for optimal visual-servoing of the WMR. The Proportional-Integral (PI) controllers are used as low-level motor speed controllers. An Adaptive-Fuzzy-Tuned-Proportional-Derivative (AFT-PD) control scheme is implemented in the high-level controller to remove the tracking errors. The Fuzzy-Logic (FL) controller serves to auto-tune the PD coefficients. The center(s) of the output membership functions of these FL controllers are adaptively updated via the least-squares method. The tracking performance of the proposed AFT-PD controller is compared with the PD controllers tuned by Particle-Swarm-Optimization (PSO) algorithm. The experimental results of the AFT-PD and PSO-PD control schemes are presented to validate the efficiency and robustness of the proposed scheme.

Keywords: Wheeled mobile robot, visual-servoing, self-tuned proportional-integral-derivative controller, adaptive fuzzy-logic controller, least-square method, particle swarm optimization.

1. INTRODUCTION

The increased utilization of wheeled mobile robots (WMRs) in various fields requires them to be equipped with robust closed-loop motion controllers to perform complex tasks with precision in real-time. These controllers depend on the feedback of a reliable sensing technique to learn the position and orientation of the WMR in the x-y plane. Usually, optical shaft encoders and digital compass (magnetometer) are mounted appropriately on the WMR for this purpose. Unfortunately, the error accumulations in these sensor readings tend to corrupt the results. An alternative approach to enhance the position measurement and tracking control of a WMR is to employ the visual-servoing method. This technique integrates mainly a vision sensor, in addition to other sensors (if needed), with the WMR. By mounting a downward-facing camera on the ceiling, the posture of a WMR can be directly obtained. The feedback from the overhead camera is processed in real-time to plan and generate a collision-free path. The attained visual information is fed to the controller which moves the WMR to track the reference trajectory with minimum deviation. The parametric uncertainty in an un-calibrated camera degrades the position information of the WMR. However, the effects of inadequate camera calibration can be completely compensated through a properly fabricated tracking controller.

Vision based indoor mobile robot navigation is a vast area. The utilization of downward facing (overhead) cameras has been very effective in indoor mobile robot navigation applications. The use of local model predictive controller in improving the trajectory tracking accuracy and time performance of a differentially driven mobile robot via robot vision has been discussed in (Pacheco et al., 2008). Recently, the fuzzy tracking controllers for vision guided WMRs have gained a lot of momentum (Keighobadi and Menhaj, 2012; Yu et al., 2014). Researchers have proposed and rigorously studied various dynamic and novel path planning strategies for WMRs (Yu et al., 2011; Yazici et al., 2014; El-sheikh et al., 2016). The robustness in visual-servoing tasks is improved by developing a control law that depends on second-order error-dynamics of the system (Hammouda et al., 2014). Lyapunov-based controllers can also effectively stabilize the error in position and orientation, of a WMR, which are deduced by two different cameras (Wang and Wang, 2012). The artificial neural network has also proven very effective in the establishment of robust tracking control schemes for WMRs, as demonstrated in (Rossomando et al., 2012; Ye, 2015). A sliding-mode trajectory tracking controller for WMRs is proposed in (Rossomando et al., 2014) that uses adaptive neural-network to implement an equivalent control in the vicinity of the sliding manifold. Different tracking control techniques yielding good results in visual-servoing of non-holonomic mobile robots have been presented in (Nitulescu, 2007; Killpack et al., 2010; Yang et al., 2012; Wang et al., 2014).

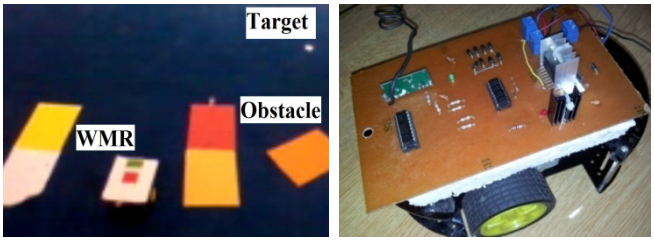
This paper outlines the implementation of a model-free and adaptive trajectory tracking controller in a differentially driven WMR. The controller is equipped with the proposed adaptive fuzzy-tuned (AFT) mechanism as well as the widely used PSO technique to optimally tune the PD coefficients. The visual information from the overhead camera is used to generate a poly-line trajectory of the shortest collision-free path towards the destination. The desired velocity profiles, extracted from the reference trajectory, are wirelessly communicated to the WMR. In order to minimize the tracking deviations, the current posture of the WMR is compared with its desired counter-part. The tracking errors are fed to the adaptive PD controllers. The performances of the two controllers (AFT-PD and PSO-PD) are comparatively analyzed by studying the path deviations exhibited by the WMR when it moves along a sequence of way-points defining the path, under the influence of each controller.

2. EXPERIMENTAL SETUP

For reliable indoor navigation and tracking control of WMR, the overhead camera captures and feeds the images of the navigation area to the Open-CV library (Gonzalez and Taha, 2008). It extracts the information related to the orientation and position of the WMR based upon the color of its orientation- and centroid-marker. The obstacles and the background are distinguished and identified based upon their distinct colors as well. The foreground is extracted. The threshold setting removes noise from the external light sources. The obstacles are grown. The visibility graph helps to compute all possible paths from the WMRs current location to the destination. The A* algorithm is used to find the shortest, safest collision free path using the adjacency matrix. The motors are rotated accordingly to traverse on the path. The camera keeps on capturing images at regular intervals. The successive frames are compared to check and see if the WMR has deviated from the actual path. The proposed controller keeps the WMR aligned with the planned trajectory with minimum error. The physical experimental setup and the WMR chassis is shown in Fig. 1(a) and (b) respectively. The image of environment is shown in Fig. 2.

2.1 Hardware setup

A web-cam is used to provide visual feedback. It is mounted at a height of 12 feet and is directly tethered to the computer.



(a) (b)

Fig. 1. (a) Experimental setup, (b) WMR chassis.

Fig. 2. Navigational environment for the experiment.

This camera covers an approximate area of 7.2 m^2 (3.0 m × 2.4 m). Once an optimal trajectory is planned, the velocity profiles and correctional commands are serially transmitted over a wireless link to the low-level speed controller running in an 8-bit embedded microcontroller at the WMR. The speed controller generates pulse-width-modulated (PWM) motor control instructions to actuate the direct-current (DC) geared motors via a dual H-Bridge motor driver circuit. The optical encoders installed at the motor shafts monitor the actual direction and speed of motor rotation. A two-wheeled differentially driven robot is used as shown in Fig. 1(b). It contains two independently driven motorized wheels and an unpowered castor-wheel.

2.2 Software setup

The differentiation between background and foreground is done by eliminating the background from the image via thresholding method. The color of the foreground (WMR and obstacles) has a higher contrast from the background (floor carpet). After suppressing the background, resulting image contains only the information of obstacles and WMR The obstacle detection is done by counting the total number of obstacles in the configuration space. The contours of the obstacles are approximated as a polygon and the coordinates of their vertices are extracted. A red-colored marker is placed at the center of the top plate of WMR, to identify its position in the area of navigation. It is differentiated from all other red regions in an image on the basis of its area. Similarly, the green-colored marker on the WMR tells us about the orientation (heading direction) of the WMR (Hovarth and Engedy, 2010). The obstacles are grown considering the maximum dimensions of the robot. The obstacles close to each other are merged into a single obstacle. All the path planning is done on the grown configuration space. The visibility graph (VG) provides all possible paths from the robot's current position to the destination position along the edges of the intervening obstacles, as shown in Fig. 3. The information regarding inter-node connection along the path is stored in an adjacency matrix.

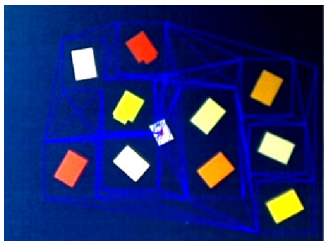


Fig. 3. Possible paths given by Visibility Graphs Algorithm.

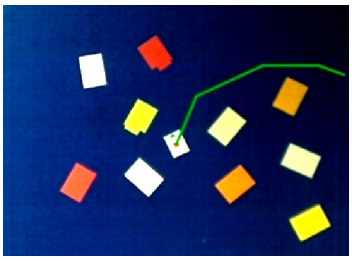


Fig. 4. Shortest path given by A* algorithm.

The A* algorithm is used to search and identify the VG for a shortest path, as shown in Fig. 4. The computed path consists of several sub-paths, making it a poly-line trajectory (Persson and Sherf, 2014; Shojaeipour et al., 2010).

3. CONTROL SYSTEM DESIGN

The classical proportional-integral-derivative (PID) controllers are the most efficient feedback controllers used in the process control industry today. However, they lack robustness when used independently. The online auto-tuning of PID controllers via soft-computing techniques yields faster rates to converge the tracking errors (Tandan and Swarnker, 2015; Bhatti et al., 2015). Therefore, in this paper, a model-free and intelligently optimized classical (PD) tracking controller is presented. The main objective of a tracking controller is to enable the WMR to optimally follow the desired path, under time constraint and in the presence of bounded disturbances. The desired trajectory formulated by the image-processing tool is composed of a sequence of straight line (or curved) segments. The corresponding spatiotemporal information associated with the path is transformed into a sequence of fixed way-points.

The block diagram of the proposed control architecture is shown in Fig. 5. It constitutes of a two layered control structure; namely, a high-level and a low-level motion controller. If the WMR gets displaced from the desired path, the controller brings it back on the path immediately. The high-level controller observes the entire image space, using successive frames, to evaluate the tracking (or deviation) errors. Based on these errors, it generates a correctional command to minimize the errors and passes it on to the low-level controller. The low-level acquires feedback from the motor encoders. It is responsible to align the WMR with the path tangent. It achieves this target by updating the speeds of the motorized wheels according to the desired and correctional profile(s) provided by the trajectory planner and the high-level controller respectively. Furthermore, it makes sure the wheels continue to rotate at the revised speeds.

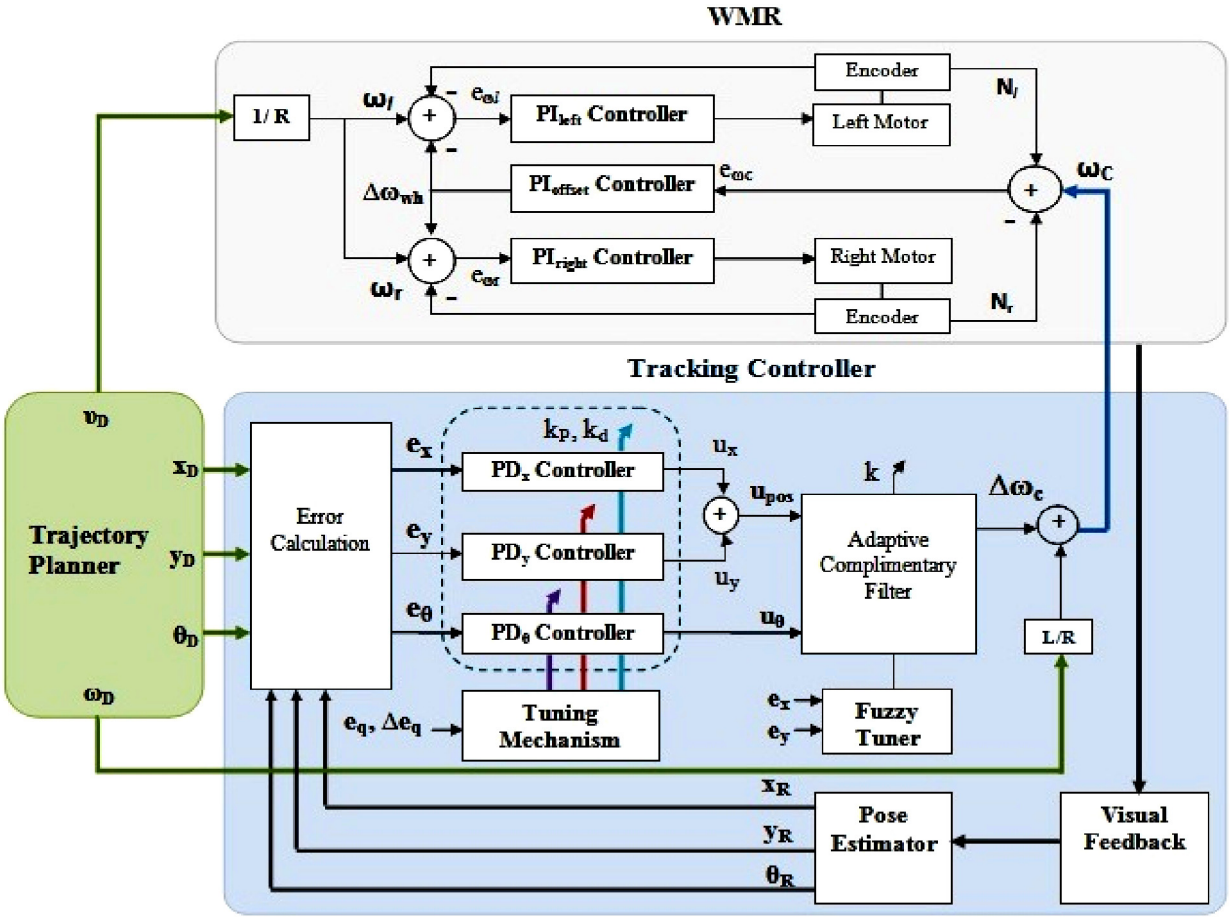


Fig. 5. Proposed trajectory-tracking control scheme.

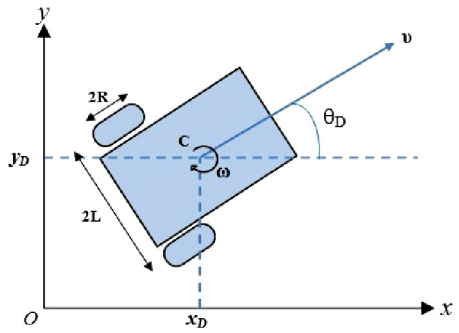


Fig. 6. Posture of WMR

The way-points consist of desired position and orientations, also known as the pose vector (posture), of the WMR. The posture of WMR is used to calculate the velocity profile of its wheels at each way-point of the trajectory. The control law uses the tracking errors, between the actual and desired posture, to decide whether the WMR should move or turn. The spatiotemporal information associated with the desired trajectory as well as the actual posture of the WMR, as shown in Fig. 6, is represented by the pose vectors given in (1) and (2) respectively.

$$q_d = [x_d, \quad y_d, \quad \theta_d]^T \tag{1}$$

$$q_r = [x_r, \quad y_r, \quad \theta_r]^T \tag{2}$$

At any given instant, *'x'* is the lateral position, *'y'* is the longitudinal position and *'θ'* is the orientation of WMR. In the proposed research, the wheel-radius (R) of the differentially driven robot is 0.0325 meters and the distance between the two wheels (2L) is 0.12 meters. The angular velocities of the actuated wheels on left and right (ω_l and ω_r) of the WMR are chosen as the input of the kinematic model as shown in (3). The WMR motors are assumed to have no slippage.

$$\omega_{wh} = [\omega_l \quad \omega_r]^T \tag{3}$$

The relationship of the linear velocity (υ) and the angular velocity (ω) of the WMR with the angular velocities of the actuated wheels is given by (4) (Solea et al., 2009).

$$\begin{bmatrix}\omega_l \\ \omega_r\end{bmatrix} = \begin{bmatrix}\frac{1}{R} & \frac{-L}{R} \\ \frac{1}{R} & \frac{L}{R}\end{bmatrix} \begin{bmatrix}v \\ \omega\end{bmatrix} \tag{4}$$

High-level controller

The main task of the high level motion controller (mentioned as 'tracking controller' and shaded blue in Fig. 5) is to nullify the tracking errors by introducing a correctional turning-offset in the angular velocity of the wheels. The high level controller (HLC), along with the trajectory planner, is implemented in the personal computer. The sampling interval, T_S, used for HLC is 1.0 milli-second (ms).

The visual feedback is received and the corresponding motion control commands are sent serially, over a wireless link, by the high-level controller. It records the deviation in the desired and the actual posture (pose vector). The deviation error consists of error in x-position (lateral), y-position (longitudinal) and orientation. The *posture-error (or tracking-error) vector* is given by (5) and is clearly illustrated in Fig. 7. These three errors are concurrently fed to three distinct PD controllers.

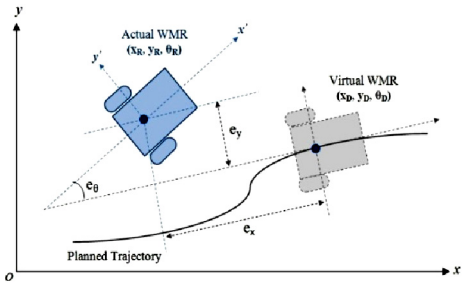


Fig. 7. Trajectory tracking errors.

$$e_q = \begin{bmatrix}e_x \\ e_y \\ e_\theta\end{bmatrix} = \begin{bmatrix}\cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1\end{bmatrix} \begin{bmatrix}x_r - x_d \\ y_r - y_d \\ \theta_r - \theta_d\end{bmatrix} \tag{5}$$

where, e_x is the lateral error, e_y is the longitudinal error and e_θ is the orientation error.

In order to optimize the available computational resources, the 'I' controller is avoided. Three PD controllers are used to individually take care of e_x, e_y, and e_θ; denoted as PD_x, PD_y, and PD_θ. The control law governing this strategy is given in (6).

$$u_q(n) = \boldsymbol{k_P} e_q(n) + \boldsymbol{k_D}\left[\frac{e_q(n) - e_q(n-1)}{T_S}\right] \tag{6}$$

where, k_P and k_D are the adaptively tuned PD coefficients. The initial coefficients, tuned via trial-and-error method, are provided in Table 1.

A. Stochastically optimized controller

The Particle-Swarm-Optimization (PSO) algorithm is a population-based stochastic optimization technique. The stochastic optimization of the PD coefficients surpasses the classical tuning techniques such as Ziegler-Nichols and Cohen-Coon method (Girirajkumar et al., 2010). This meta-heuristic algorithm begins with a random initial population of candidate solutions, called 'particles', and searches through the space to converge to the best-fit solution.

Table 1. PD coefficients.

PD Controller	k_P	k_D
Lateral-position (PD_x)	1.347	7.881×10^{-3}
Longitudinal-position (PD_y)	1.361	8.134×10^{-3}
Orientation (PD_θ)	1.736	10.826×10^{-3}

The initial search space of PD coefficients for PSO is taken as [0, 2]. Each particle has a position and velocity associated to it. The equations to update the velocity (V_i) and position (X_i) of a particle *'i'* are given in (7) and (8) respectively.

$$V_i = w_i V_i + b_1 r_1 (P_i - X_i) + b_2 r_2 (P_g - X_i) \tag{7}$$

$$X_i = X_i + V_i \tag{8}$$

where,

b_1, b_2 are the cognitive coefficients
r_1, r_2 are random real numbers in the interval [0, 1]
w is the inertia-weight

The optimizer serves to evaluate, store, and compare the fitness value of each particle with the best fitness value, also known as "local-best" (P_i), available so far. The function used to evaluate the fitness of the particles based on the current error, $e_q(n)$, is given in (9).

$$Fitness = \frac{1}{1+{e_q}^2(n)} \tag{9}$$

If a better fitness value is achieved then it is set as the new P_i. The particle with best fitness value among all the particles in the population is chosen as the "global-best" (P_g) (Gharghory and Kamal, 2012). The inertia-weight (w) of the optimizer decreases from 1 to 0 in each iteration, and converges the search space to best-fit solution. It is given by (10).

$$w_i = w_{max} - \left(\frac{w_{max} - w_{min}}{d_{max}}\right) d \tag{10}$$

where,

d_{max} is the maximum number of iterations
d is the current number of iteration

The flowchart of the PSO algorithm is shown in Fig. 8. The PSO algorithm optimally tunes the k_P and k_D coefficients of the three PD controllers, at each sampling interval, as shown in Fig. 9 (Djoewahir et al., 2013).

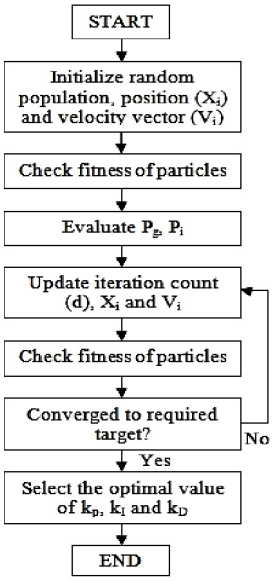


Fig. 8. Flowchart of PSO algorithm.

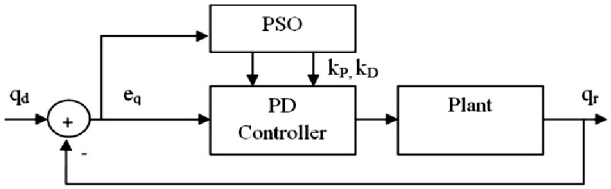


Fig. 9. PSO-PD controller.

B. Intelligently optimized adaptive controller

The fixed PD coefficients lead to slow convergence of tracking error and decrease the tolerance of the system to uncertain disturbances. Therefore, an adaptive-fuzzy mechanism is employed for the proper online tuning of the PD coefficients. The heuristic construction of the membership functions and rule bases for a Fuzzy-Logic Controller (FLC) proves to be insufficient to cater the unprecedented dynamic variations in practical control systems. Hence, the least-squares method proposed by (Elshazly et al., 2014) is used to adaptively tune the centers (c_i) of the output fuzzy membership functions. This technique removes uncertainties associated with the linguistic variables. The control architecture of the *Adaptive Fuzzy Tuned - PD (*AFT-PD) controller is shown in Fig. 10. The adaptive inference mechanism uses the current fuzzy controller output, tracking error and its rate-of-change as the inputs to update the centers of the output membership functions (MF). This way, the online PD-coefficient tuning based on the rule bases defined by the expert's experience can be avoided. The fuzzy system is defined by (11) or (12).

$$u_{COA} = \frac{\sum_{i=1}^{n} \mu_i(c) c_i}{\sum \mu_i(c)} \tag{11}$$

$$u_{COA} = \frac{\mu_1(c)c_1}{\sum \mu_1(c)} + \frac{\mu_2(c)c_2}{\sum \mu_2(c)} + \frac{\mu_3(c)c_3}{\sum \mu_3(c)} + \cdots + \frac{\mu_n(c)c_n}{\sum \mu_n(c)} \tag{12}$$

where,

u_{COA} is the control value based on centre-of-area
$\mu(c)$ is the grade value of MF
c_i is the value of the center of output MF

Using the substitution given in (13), the expression of u_{COA} can be simplified as given in (14).

$$\delta_i(c) = \frac{\mu_i(c)}{\sum \mu_i(c)} \tag{13}$$

$$u_{COA} = \delta_1(c)c_1 + \delta_2(c)c_2 + \cdots + \delta_n(c)c_n \tag{14}$$

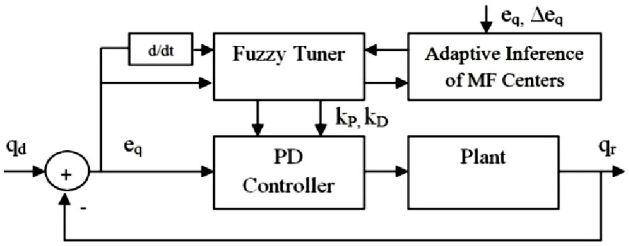


Fig. 10. Proposed AFT-PD controller.

The u_{COA} can be further reduced to the form given in (15).

$$u_{COA} = \varphi\beta(c) \tag{15}$$

where,

$\varphi(c) = \delta^T(c) = [\delta_1, \delta_2, .., \delta_n]^T$
$\beta(c) = [c_1, c_2, .., c_n]^T$

The corresponding error function can be defined by the (16). Using the error vector in (17), the expression for the error function in (16) can be re-written as (18).

$$e = u_{COA} - \varphi\beta(c) \tag{16}$$

$$E(n) = [e_1, e_2, .., e_n]^T \tag{17}$$

$$E(n) = u_{COA} - \varphi\beta \tag{18}$$

The squared error is defined by (19).

$$E^T E = {u_{COA}}^T u_{COA} - 2\beta^T \varphi^T u_{COA} + \beta^T \varphi^T \varphi\beta \tag{19}$$

The squared error can be minimized by partially differentiating it with respect to β, and then putting the

resultant expression equal to zero. Thus, the least square estimate of β can be evaluated according to (20).

$$\widehat{\beta_{LS}} = (\varphi^T \varphi)^{-1} \varphi^T u_{COA} \quad (20)$$

The FLC updates the MF centre based on the changes in error dynamics. Mamdani model is applied with some modification to attain the optimal parameters. The rule-base of k_P and k_D is shown in Table 2 and 3 respectively. The linguistic variables of the input MF are defined as; Negative-Big (NB), Negative-Small (NS), Zero (ZE), Positive-Small (PS) and Positive-Big (PB). The linguistic variables of the output MF are; Small (S), Medium (M) and Big (B). The input and output MFs are shown in Fig. 11 and 12 respectively.

Table 2. Fuzzy rule base for k_P.

$e_q \mid \Delta e_q$	NB	NS	ZE	PS	PB
NB	B	M	S	M	B
NS	M	M	S	M	M
ZE	S	S	S	S	S
PS	M	M	S	M	M
PB	B	M	S	M	B

Table 3. Fuzzy rule base for k_D.

$e_q \mid \Delta e_q$	NB	NS	ZE	PS	PB
NB	M	S	S	S	M
NS	B	M	S	M	B
ZE	B	B	M	B	B
PS	B	M	S	M	B
PB	M	S	S	S	M

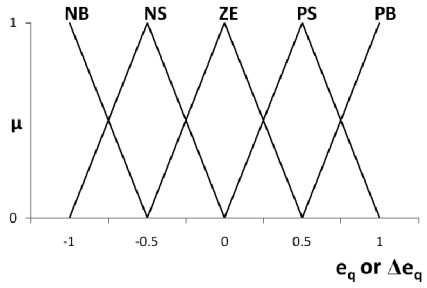


Fig. 11. Input membership function.

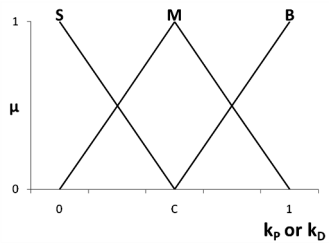


Fig. 12. Output membership function,

C. Fusion of controller outputs

The sum of control outputs from the PD_x and PD_y controller (u_x and u_y) provides the position control command, u_{pos}, as given by (21). The output of the PD_θ controller is u_θ.

$$u_{pos}(n) = u_x(n) + u_y(n) \quad (21)$$

These control outputs contribute in deciding the extent of turning required by the WMR to stay on track. The e_θ might increase abruptly while correcting the e_x and e_y. The contradiction in concurrent attempts to correct the position and orientation slows down the convergence and tracking process. As a remedy, the control outputs (u_{pos} and u_θ) are fused via an adaptive complimentary filter (ACF), as shown in (22).

$$\Delta\omega_c(n) = (k) \times u_{pos}(n) + (1-k) \times u_\theta(n) \quad (22)$$

The value of the '*k*', a real number belonging to [0, 1], is dynamically varied online via a dedicated FLC tuner. The e_x and e_y serve as the input of the FLC tuner. This summing technique adaptively prioritizes the tasks of position correction and orientation correction during path-traversal. When WMR is off the track, the priority is given to position correction. Once it reaches the desired position, then the priority is given to orientation correction. The fuzzy rule base of the tuner is given in Table 4.

Table 4. Fuzzy rule base for *k*.

$e_x \mid e_y$	NB	NS	ZE	PS	PB
NB	B	B	M	B	B
NS	B	M	S	M	B
ZE	M	S	S	S	M
PS	B	M	S	M	B
PB	B	B	M	B	B

The input and output MFs of the tuner are similar to those shown in Fig. 11 and 12 respectively. However, the value of '*c*' in the output MF is fixed at 0.5 in this case. The sum of the desired angular velocity (ω_D) and $\Delta\omega_c$, given in (23), generates the correctional angular speed (ω_c) to appropriately turn the WMR.

$$\omega_c(n) = \omega_D(n) \pm \Delta\omega_c(n) \quad (23)$$

3.1 Low-level controller

The low-level controller (LLC) contains PI speed controllers for the actuation of the two DC motors of WMR. The associated control law is given in (24).

$$u_q(n) = \boldsymbol{k_P} e_q(n) + \boldsymbol{k_I}\left[T_S \sum e_q(n)\right] \quad (24)$$

where, k_P and k_I are the adaptively tuned PI coefficients.

This controller is implemented in an 8-bit microcontroller. The sampling interval (T_S) is 1.0 ms. The LLC ensures precise velocity tracking of the motorized wheels. It is shown in the grey-shaded area of Fig. 5 (Braunl, 2008). The desired linear velocity (υ_D) of robot is provided by the trajectory planner. The linear velocity command is transformed into angular velocity (ω_l and ω_r) of the wheels and then fed to the motors. The shaft encoders monitor the actual speed and direction of motor rotation (N_l and N_r). An additional PI_{offset} controller is deployed in the low-level controller. The input to this PI controller, given in (25), is the turning command issued to WMR with respect to the current wheel rotations.

$$e_{\omega c} = \omega_c + (N_l - N_r) \quad (25)$$

The PI_{offset} controller generates a compound offset ($\Delta\omega_{wh}$) control command which is added (or subtracted) to the velocity command of the two wheels, as shown in (26).

$$e_{\omega l|\omega r} = (\omega_{l|r} - N_{l|r}) \pm \Delta\omega_{wh} \quad (26)$$

Based on the values of $e_{\omega l}$ and $e_{\omega r}$, the PI_{left} and PI_{right} controllers generate appropriate PWM commands to drive the motors. The coefficients of PI controllers, shown in Table 5, are tuned via the Ziegler-Nicholas method (Bequette, 2003).

4. TESTS AND RESULTS

Before formally testing the proposed scheme, the correctional contributions rendered by the constituent modules of the control architecture in optimizing the overall trajectory tracking performance of the WMR are to be validated. Therefore in this section, first of all, the LLC and ACF designs are individually analyzed.

Table 5. PI coefficients.

PI Controller	k_P	k_I
Left motor	21.283	1.483×10^4
Right motor	22.655	1.434×10^4
Correction offset	6.668	4.153×10^4

The PI_{offset} controller in the LLC serves to correct the yaw rotation of WMR along the path. In the absence of any turning command, it synchronizes the velocities of the two motors. This enables the WMR to move on a straight-line path with minimum deviation. Both motors are rotated at 1.8 rad/s. The response acquired from the motors without the PI_{offset} controller is shown in Fig. 13. A tuned PI_{offset} is then introduced and the experiment is repeated. The resulting response is shown in Fig. 14. The difference between the motor velocities without and with the PI_{offset} is represented in Fig. 15(a) and 15(b) respectively. The PI_{offset} has drastically reduced the difference and has synchronized the velocities. The absolute-maximum-error (AME) and root-mean-square-error (RMSE) between the velocities, inferred from Fig. 15, is provided in Table 6.

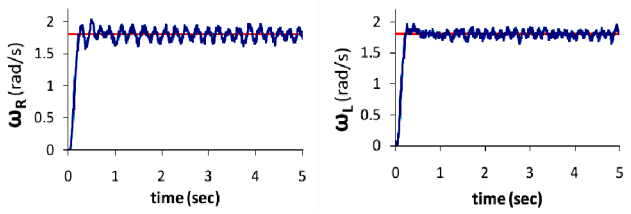


Fig. 13. Right and left motor velocity without PI_{offset}.

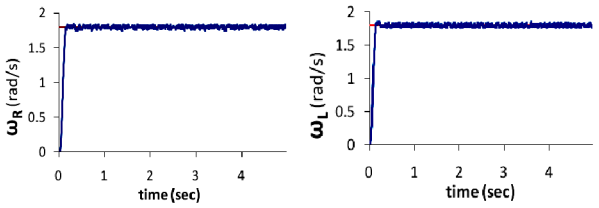


Fig. 14. Right and left motor velocity with PI_{offset}.

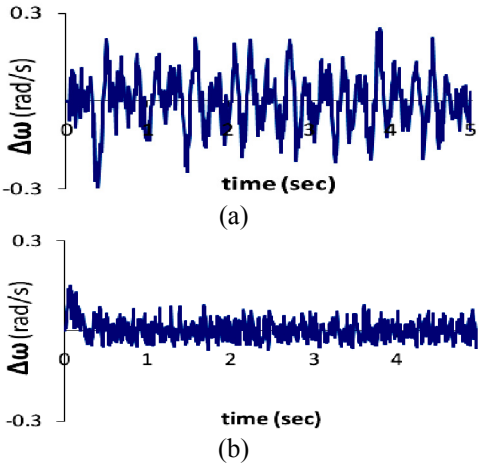


Fig. 15. Difference in motor velocity (a) without PI_{offset}, (b) with PI_{offset}.

Table 6. Summary of motor velocity difference

Control structure	AME (rad/s)	RMSE
LLC without PI_{offset}	0.300	0.262
LLC with PI_{offset}	0.138	0.048

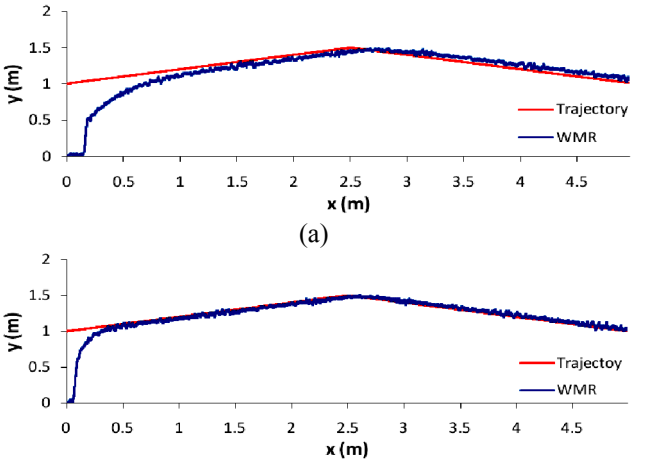


Fig. 16. Trajectory tracking (a) without adaptive fusion, (b) with adaptive fusion.

The adaptive fusion of the control outputs (u_{pos} and u_θ) also improves the trajectory tracking performance. The WMR is made to track the reference trajectory shown in Fig. 16. It is intentionally displaced from the actual starting point. In the first trial, the value of k (in ACF) is kept constant at 0.5. As shown in Fig. 16(a), without the adaptive fusion, the WMR converges very slowly towards the path. It maintains a constant Euclidean distance (error) of 0.04 m from the path. On the contrary, the WMR quickly returns to the nearest desired way-point on the path and tracks it accurately when k is dynamically tuned, as shown in Fig. 16(b).

The response of the PSO-PD and AFT-PD controllers is initially tested with a unit-step input. A step input of 1.0 rad. is applied to the orientation controller (PD_θ) in HLC with the PSO based tuning mechanism. The response is shown in Fig.

17. In the second trial, the same input is applied to the orientation controller that is equipped with the proposed AFT mechanism. The resulting response is shown in Fig. 18. The summary in Table 7 shows that the transient and steady state response of the AFT-PD controller is better than that of PSO-PD controller.

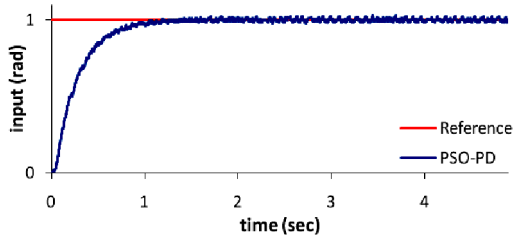


Fig. 17. PSO-PD controller response.

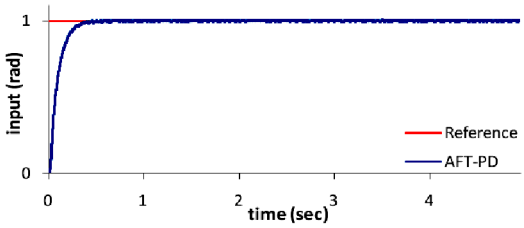


Fig. 18. AFT-PD controller response.

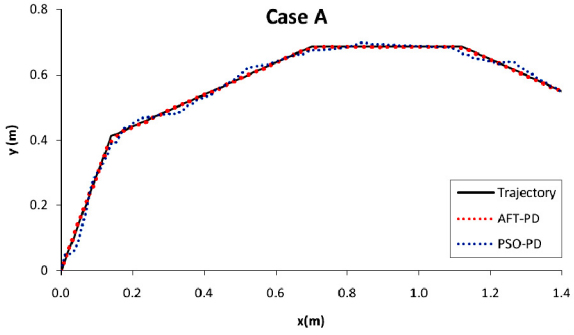
Table 7. Summary of controller step-response.

Controller	Overshoot	Rise time (s)	Steady-state error (rad.)
AFT-PD	None	0.32	±0.008
PSO-PD	None	0.92	±0.079

The poly-line trajectory shown in Fig. 4 is used to further test the tracking performance and validate the effectiveness of the proposed AFT-PD controller over the PSO-PD controller. The following three unique test-cases are employed.

a. *Case A:* The WMR moves on the path normally.
b. *Case B:* The WMR moves on the path while carrying an additional mass of 0.25 kg (as a mechanical disturbance).
c. *Case C:* The WMR moves on the path while bounded perturbations are added in its angular velocity commands.

The trajectories traversed by the WMR in real-time when analyzed using the three test-cases, under the influence of the AFT-PD and PSO-PD tracking controllers, are shown in Fig. 19.



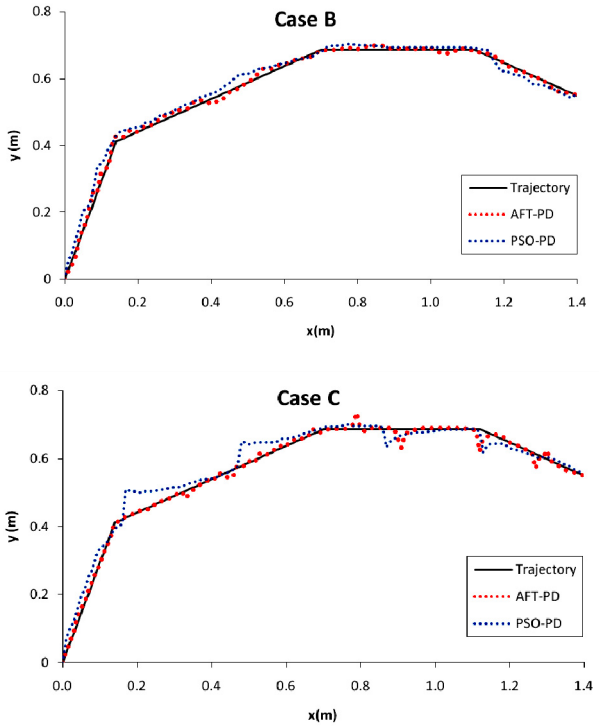


Fig. 19. Trajectory tracking in case A, B, and C.

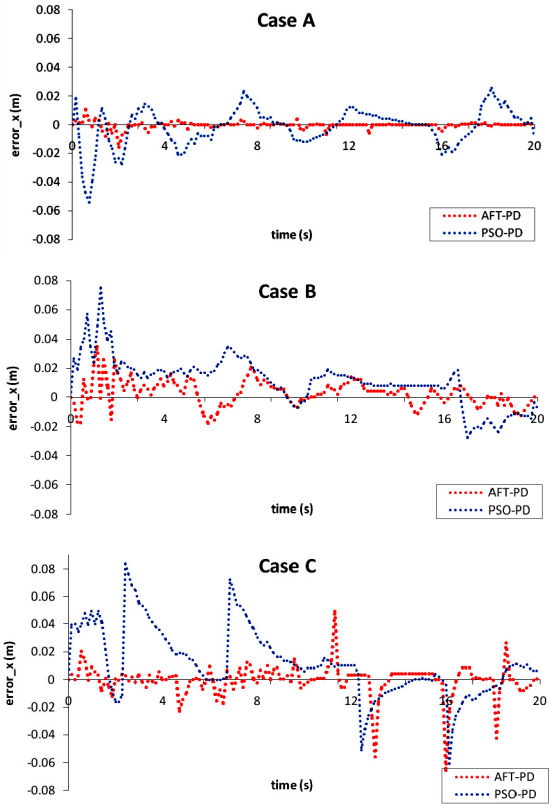


Fig. 20. Lateral errors (e_x) for case A, B and C.

The lateral errors (e_x) of the WMR for the case A, B, and C are shown in Fig. 20. The longitudinal errors (e_y) of the WMR for the case A, B, and C are shown in Fig. 21.

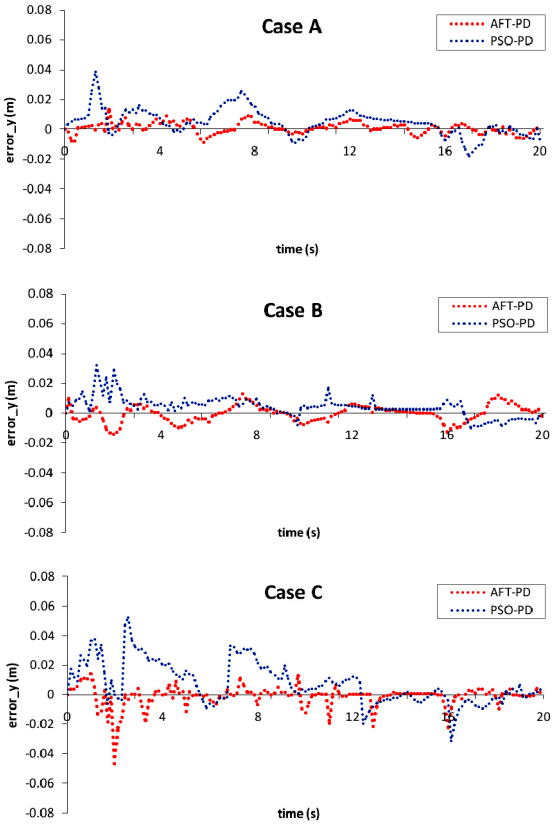
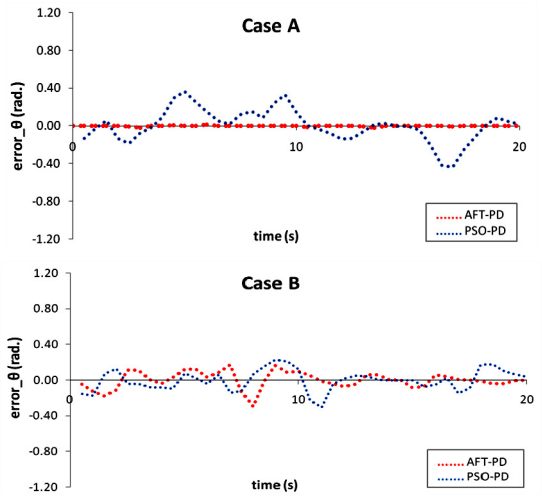


Fig. 21. Longitudinal error (e_y) in case A, B, and C.



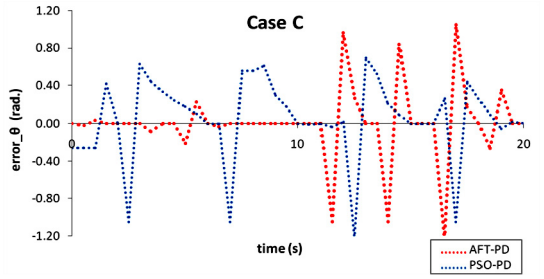


Fig. 22. Orientation error (e_θ) for case A, B, and C.

The orientation errors (e_θ) of the WMR, for the case A, B, and C are represented in Fig. 22. The Table 8 summarizes the results of a total of six real-time experiments conducted using the three test-case for each of the two high-level trajectory tracking controllers. The tracking controller implemented with the PSO-PD controller demonstrates a slower convergence rate, and exhibits comparatively larger trajectory deviations in all test-cases. Consequently, a lot of time is spent on nullifying these trajectory errors. The AFT-PD controllers have proven to be most efficient and robust. They exhibit faster convergence rate and comparatively smaller deviations from the desired trajectory. For AFT-PD, in case A and B, the absolute maximum values of lateral and longitudinal errors stay under 0.031 m and the maximum absolute values of orientation error is under 0.35 rad. The maximum absolute error in case C is induced due to artificially generated perturbations. However, the AFT-PD controller efficiently rejects these disturbances and returns the WMR to path within 1.20 s. in the worst case scenario. The controller assists the WMR to converge to the nearest desired way-point very quickly. Using the proposed control strategy, the WMR stays mostly on the path.

5. CONCLUSION

This paper addresses the validity of an adaptive fuzzy tuned classical control solution for the precise trajectory tracking by WMRs. It considers only error dynamics of the system. Unlike the model-based techniques, the proposed system does not rely upon the knowledge of the dynamic model of the WMR.

Table 8. Summary of experimental results.

High-level controller	Case	e_x		e_y		e_θ	
		AME (m)	RMSE	AME (m)	RMSE	AME (rad.)	RMSE
PSO-PD	A	0.058	0.013	0.039	0.012	0.490	0.042
	B	0.076	0.036	0.052	0.019	0.349	0.072
	C	0.084	0.067	0.060	0.026	1.152	0.182
AFT-PD	A	0.017	0.003	0.013	0.004	0.003	0.002
	B	0.031	0.009	0.027	0.008	0.345	0.045
	C	0.071	0.018	0.045	0.009	1.117	0.084

The intelligent robotic system presented in this paper contains two layers of control system integrated with a robust adaptation mechanism to make the overall design more robust and responsive to changes in WMRs posture caused by parametric uncertainties and random disturbances. It efficiently converges the tracking error (e_q) to zero. The Table 6 authenticates the introduction of an additional PI controller in the LLC to turn the WMR precisely on the path as well as to synchronize the motor velocities. The efficacy of the adaptive fusion of control outputs to fasten the convergence and optimize the trajectory tracking is clearly illustrated in Fig. 16. The WMR equipped with the proposed AFT-PD as well as the PSO-PD based high-level controller is tested via the three test-cases on the same trajectory; the normal case (A), the mechanically disturbed case (B) and the randomly perturbed case (C). The results of real-time experiments clearly demonstrate the optimal performance of the proposed high-level tracking controller even in the presence of bounded exogenous disturbances. In the PSO-PD controller, the tracking error exceeds abnormally at certain occasions and converges very slowly afterwards. This leads to an overall slower system response. The observations summary shown in Table 7 and 8 clearly manifests that the high-level controller with AFT-PD controller is superior to the PSO-PD controller. The high-level controller's performance can be further optimized by analyzing and retrofitting different soft-computing, evolutionary and intelligent adaptation mechanisms with the conventional classical controllers. A hybrid of advanced non-linear control schemes involving back-stepping, sliding-mode, linear quadratic regulators and model-predictive controllers can also be implemented to enhance the tracking performance.

REFERENCES

Bequette, B.W. (2003). *Process Control: Modeling, Design, and Simulation,* pp. 195-209. Prentice Hall, New Jersey, USA.

Bhatti, O.S., Mehmood-ul-Hassan, K., Imtiaz, M.A. (2015). Attitude Control and Stabilization of a Two-Wheeled Self-Balancing Robot. *Control Engineering and Applied Informatics*, 17(3), pp. 98-104.

Braunl, T. (2008). *Emedded Robotics*, pp. 83-101. Springer, Heidleberg, Germany.

Djoewahir, A., Tanaka, K., and Nakashima, S. (2013). Adaptive PSO-Based Self-Tuning PID Controller for Ultrasonic Motor. *International Journal of Innovaive Computing, Information and Control*, 9(10), pp. 3903-3914.

Elshazly, O., El-bardini, M., and El-Rabaie, M. (2014). Adaptive Fuzzy Iterative Learning Controller for X-Y table position control. In: *IEEE International Conference on Automation, Quality and Testing, Robotics*, Cluj, Napoca, pp. 1-6.

Elsheikh, E.A., El-Bardini, M.A., and Fkirin, M.A. (2016). Dynamic path planning and decentralized FLC path following implementation for WMR based on visual servoing. In: *3rd MEC International Conference on Big Data and Smart City*, Muscat, Oman, pp. 1-7.

Gharghory, S.M., and Kamal, H.A. (2012). Optimal Tuning of PID Controller using Adaptive Hybrid Particle Swarm Optimization Algorithm. *International Journal of Computers, Communication and Control*, 7(1), pp. 101-114.

Girirajkumar, S.M., Jayaraj, D., and Kishan, A.R. (2010). PSO based Tuning of a PID Controller for a High Performance Drilling Machine. *International Journal of Computer Applications*, 1(19), pp. 12-18.

Gonzales, J., and Taha, Z. (2008). Mobile robot navigation using open computer vision with fuzzy controller. *Journal of Advanced Computational Intelligence and Intelligent Informatics,* 12(4), pp. 336–337.

Hammouda1, I., Mekki, H., Kaaniche1, K., and Chtourou, M. (2014). Improving Mobile Robot Robustness in Visual Servoing Application. *International Journal of Control and Automation*, 7(10), pp. 331-342.

Horvath, G., and Engedy, I. (2010). Global camera-based localization and prediction of future positions in mobile robot navigation. In Rudas, I.J., Fodor, J., and Kacprzyk, J. (ed.), *Computational Intelligent in Engineering*, pp. 61-73. Springer-Verlag, Berlin-Heidelberg.

Keighobadi, J., and Menhaj, M.B. (2012). From Nonlinear to Fuzzy Approaches in Trajectory Tracking Control of Wheeled Mobile Robots. *Asian Journal of Control,* 14(4), pp. 960-973.

Killpack, M., Deyle, T., Anderson, C., and Kemp, C.C. (2010). Visual odometry and control for an omni-directional mobile robot with a downward-facing camera. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems,* Taipei, China, pp. 139-146.

Nitulescu, M. (2007). Solutions for modeling and control in mobile robotics. *Control Engineering and Applied Informatics*, 9(3;4), pp. 43-50.

Pacheco, L., Ferrer, J., and Luo, N. (2008). Local model predictive control experiences with differential driven wheeled mobile robots. *Control Engineering and Applied Informatics*, 10(2), pp. 59-67.

Persson, S.M., and Sharf, I. (2014). Sampling-based A* algorithm for robot path-planning. *International Journal of Robotics Research*, 33(13), pp. 1683-1708.

Rossomando, F.G., Soria, C., and Carelli, R. (2012). Neural network-based compensation control of mobile robots with partially known structure. *IET Journal of Control Theory and Applications*, 6(12), pp. 1851-1860.

Rossomando, F.G., Soria, C., and Carelli, R. (2014). Sliding Mode Control for Trajectory Tracking of a Non-holonomic Mobile Robot using Adaptive Neural Networks. *Control Engineering and Applied Informatics*, 16(1), pp. 12-21.

Shojaeipour, S., Gholami, E., Haris, SM., and Shojaeipour, A. (2010). Webcam-based mobile robot path planning using voronoi diagrams and image processing. In: *9th WSEAS International Conference on Applications of Electrical Engineering*, Penang, Malaysia, pp. 151-156.

Solea, R., Filipescu, A., and Nunes, U. (2009). Sliding-Mode Control for Trajectory-Tracking of a Wheeled Mobile Robot in Presence of Uncertainties. In: *7th Asian Control Conference*, Hong Kong, pp. 1701-1706.

Tandan, N., and Swarnkar, K.K. (2015). PID Controller Optimization By Soft Computing Techniques-A Review.

International Journal of Hybrid Information Technology, 8(7), pp. 357-362.

Wang, B., and Wang, C. (2012). Tracking control for nonholonomic mobile robots with visual servoing feedback. In: *10th World Congress on Intelligent Control and Automation,* Beijing, China, pp. 3864-3869.

Wang, K., Liu, Y., and Li, L. (2014). Visual Servoing Trajectory Tracking of Nonholonomic Mobile Robots without Direct Position Measurement. *IEEE Transactions on Robotics*, 30(4), pp. 1026-1035.

Yang, F., Wang, C., Chen, H., and Zhang, D. (2012). Adaptive tracking control for uncertain dynamic nonholonomic mobile robots based on visual servoing. In: *31st Chinese Control Conference*, Hefei, China, pp. 3047-3051.

Yazici, A., Kirlik, G., Parlaktuna, O., and Sipahioglu, A. (2014). A dynamic path planning approach for multirobot sensor-based coverage considering energy constraints. *IEEE Transactions on Cybernetics*, 44(3), pp. 305-314.

Ye., J. (2015). Tracking control of a nonholonomic wheeled mobile robot using improved compound cosine function neural networks. *International Journal of Control,* 88(2), pp. 364-373.

Yu, C.J., Chen, Y.H., and Wong, C.C. (2011). Path planning method design for mobile robots. In: *2011 SICE Annual Conference*, Tokyo, Japan, pp. 1681-1686.

Yu, H., Tang, G.Y., Su, H., Tian, C.P., and Zhang, J. (2014). Trajectory tracking control of wheeled mobile robots via fuzzy approach. In: *33rd Chinese Control Conference*, Nanjing, China, pp. 8444-8449.

Zhengcai, C., Yingtao, Z., and Qidi, W. (2011). Adaptive Trajectory Tracking Control for a Nonholonomic Mobile Robot. *Chinese Journal of Mechanical Engineering*, 24(3), pp. 1-7.