Algorithms for Controlling a Nonholonomic Differential Drive Wheeled Mobile Robot

Ali ALOUACHE, School of Automation, Beijing Institute of Technology, China, alouache15@yahoo.fr
QingHe WU, School of Automation, Beijing Institute of Technology, China, qinghew@bit.edu.cn

Abstract- This paper discusses different algorithms for controlling an autonomous nonholonomic differential drive wheeled mobile robot (WMR) on the horizontal X-Y plane. The presented algorithms are used for controlling the WMR to perform several useful tasks like the following: moving to a goal point, moving to specific pose, line following and path following. Each algorithm designs the necessary input velocities for the robot kinematic model such that the WMR can accomplish the corresponding task efficiently. The delicate task here for the WMR is to follow any given reference path, hence a fuzzy PD controller is proposed in this paper to improve the effectiveness of the path following controller. The fuzzy PD controller is designed based on the fuzzy logic approach. The presented techniques are simple and efficient therefore they can be used in real time applications like control and navigation of mobile robots. Finally, Matlab simulations are given to test the different algorithms.

I. INTRODUCTION

Mobile robots are robots that can move from one place to another autonomously, that is, without assistance from external human operators [1]. Unlike the majority of industrial robots that can move only in a specific workspace, mobile robots have the special feature of moving around freely within a predefined workspace to achieve their desired goals. This mobility capability makes mobile robots suitable for a large repertory of applications in structured and unstructured environments. Ground mobile robots are distinguished in wheeled mobile robots (WMRs) [2, 3, 4, 5] and legged mobile robots (LMRs) [6, 7]. Mobile robots also include unmanned aerial vehicles (UAVs) [8, 9, 10], and autonomous underwater vehicles (AUVs) [11, 12, 13]. WMRs are very popular because they are appropriate for typical applications with relatively low mechanical complexity and energy consumption.

Robot kinematics deals with the configuration of robots in their workspace, the relations between their geometric parameters, and the constraints imposed in their trajectories. The study of kinematics is a fundamental prerequisite for the study of dynamics, the stability features, and the control of the robot. The development of new algorithms for kinematic controllers is still a topic of ongoing research [14, 15, 16, 17, 18, 19, 20], toward the end of constructing robots that can perform more sophisticated and complex tasks in industrial and societal applications.

This paper proposes different algorithms for controlling an autonomous nonholonomic differential drive wheeled mobile robot (WMR) moving on the X-Y plane. The WMR is desired to perform several tasks, the first control approach is moving to a goal point \((x^*, y^*)\), where the linear and angular velocities of the kinematic model are controlled to drive the WMR to the goal point. The second control approach is moving to specific pose \((x^*, y^*, \theta^*)\), which is solved by using polar coordinates transformation. Another control approach is following a target line defined by the equation \((ax+by+c)\). The last approach but the most important one is path following. The path following algorithm designs the input velocities such that the tracking error function is reduced to zero. The linear velocity input of the WMR is firstly controlled by a PD controller, but the efficiency of the PD controller is limited, therefore a fuzzy PD controller is developed in this paper to improve the effectiveness of the path following controller. The fuzzy PD controller is developed based on the fuzzy logic approach [21]. The fuzzy PD controller advantageous because it provides more stability to the path following control system and the WMR can track faster the reference path in the X-Y plane.
This paper is structured as follows: section II establishes the differential drive kinematics of the wheeled mobile robot (WMR). Moving to goal point and moving to specific pose controllers design are presented in section III and Section IV respectively. Section V presents the line following approach. Section VI presents the path following approach and designs the fuzzy PD controller. Section VII demonstrates the simulation results and section VIII gives some general conclusions about the work.

II. DIFFERENTIAL DRIVE KINEMATICS

Fig.1 displays a nonholonomic differential drive wheeled mobile robot on the X-Y plane with the different parameters and center of mass C.

Remark 1: the WMR has two driving wheels mounted on the same axis and a free front wheel. The two driving wheels are derived to achieve both the orientation and translation pose.

The derived nonlinear kinematic model which expresses the motion of the WMR is

\[
\begin{bmatrix}
\dot{x}(t) \\
\dot{y}(t) \\
\dot{\theta}(t)
\end{bmatrix}
= 
\begin{bmatrix}
\cos(\theta(t)) & 0 \\
\sin(\theta(t)) & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
v(t) \\
w(t)
\end{bmatrix}
\]  

(1)

Where \(q(t) = (x(t), y(t), \theta(t))^T\) is the robot pose cartesian coordinates at instant time \(t\). \(q_0 = (x_0, y_0, \theta_0)\) is the initial pose coordinates of the robot center of mass \(C\). \(v(t), w(t)\) and \(\theta(t)\) are respectively the linear velocity, the angular velocity, and the heading angle of the robot where \((-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2})\).

The kinematic model of (1) describes the velocities of the vehicle but not the forces or torques that cause the velocity. The mechanical structure of the WMR is nonholonomic, it satisfies the following constraint

\[
\dot{x}(t)\sin(\theta(t)) - \dot{y}(t)\cos(\theta(t)) = 0
\]

(2)

This constraint means that WMR cannot move in the direction of the wheel axis (i.e. y).
III. MOVING TO GOAL POINT

Consider the problem of moving toward a goal point \((x^*, y^*)\) in the plane. Control the robot’s velocity to be proportional to its distance from the goal

\[
v_1 = \sqrt{(x^*-x)^2 + (y^*-y)^2}
\]

and to steer toward the goal which is at the vehicle-relative angle

\[
\theta'(t) = \tan^{-1}\left(\frac{y^*-y}{x^*-x}\right)
\]

using a proportional controller

\[
w_1 = k_1 \text{ang_diff}(\theta'(t) - \theta(t))
\]

which turns the steering wheel toward the target.

IV. MOVING TO SPECIFIC POSE

This is to drive the mobile robot to a specific pose \((x^*, y^*, \theta^*)\). The controller design is based on polar coordinates transformation as shown in fig 2. We apply a change of variables using the notation indicated in fig 2:

\[
\begin{align*}
\rho &= \sqrt{\Delta_x^2 + \Delta_y^2} \\
\alpha &= \tan^{-1}\frac{\Delta_y}{\Delta_x} - \theta \\
\beta &= -\theta - \alpha
\end{align*}
\]

where \(\rho\) is the distance to the goal, \(\beta\) is the angle of the goal vector with respect to the world frame, and \(\alpha\) is the angle of the goal vector with respect to the vehicle frame. Which results in

\[
\begin{pmatrix}
\dot{\rho} \\
\dot{\alpha} \\
\dot{\beta}
\end{pmatrix} =
\begin{pmatrix}
\cos \alpha & 0 \\
\sin \alpha & 0 \\
-\frac{\sin \alpha}{\rho} & -\frac{\cos \alpha}{\rho}
\end{pmatrix}
\begin{pmatrix}
0 \\
-1 \\
0
\end{pmatrix}
\begin{pmatrix}
v \\
w \end{pmatrix}
\]

if \(\alpha \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]\)

and assumes the goal \(\{G\}\) is in front of the vehicle. The linear control law is given as follows

\[
\begin{align*}
v_2 &= k_{\rho} \rho \\
w_2 &= k_\alpha \alpha + k_\beta \beta
\end{align*}
\]

Figure 2. Polar coordinate notation for the WMR moving to specific pose
The dynamics of the closed loop systems is

\[
\begin{pmatrix}
\dot{p} \\
\dot{\alpha} \\
\dot{\beta}
\end{pmatrix} =
\begin{pmatrix}
-k_p \cos \alpha \\
-k_\alpha \alpha - k_p \beta \\
-k_\alpha \sin \alpha
\end{pmatrix}
\]  

(8)

The closed loop system is stable as long as \( k_p > 0, k_\beta < 0, k_\alpha - k_p > 0 \). Is assumed that the goal is in front of the WMR then \( \alpha \in \left[ -\frac{\pi}{2}, \frac{\pi}{2} \right] \).

V. LINE FOLLOWING

A wheeled mobile robot moving on the \( X-Y \) plane with a constant velocity \( v_3(t) \), then it supposed to follow a target line defined by the equation \( ax + by + c \). The distance between the WMR and the line is defined by the following

\[
d = \sqrt{(x - x_l)^2 + (y - y_l)^2}
\]

(9)

The orientation angle of the line is constant and it is computed by

\[
\theta_d = \tan^{-1}\left(\frac{a}{b}\right)
\]

(10)

The control law for the angular velocity \( w_3 \) is

\[
w_3 = -k_2 d + k_3 \text{angle_difference}(\theta', \theta)
\]

(11)

VI. PATH FOLLOWING

Instead of target line following control task, which is presented in the previous section, here the mobile robot is desired to follow reference path.

Definition 1: the reference path is a sequence of coordinates \( q_r(t) \) given as follows:

\[
q_r(t) = (x_r(t), y_r(t), \theta_r(t))^T
\]

(12)

It is generated by motion planning techniques or in real time from the WMR’s sensors.

The general structure of the control system is shown in fig. 3. The objective of controller is to generate the input commands \( v(t) \) and \( w(t) \) for kinematic model of the WMR such that the WMR moves toward the reference point.

The tracking error vector \( \hat{X}(t) \) indicated in fig.3, is defined as follows

\[
\hat{X}(t) = q_r(t) - q(t)
\]

(13)

Where \( q_r(t) \) is the reference path and \( q(t) \) is the actual state of the WMR measured by the corresponding navigation system of the WMR

\[
q(t) = (x(t), y(t), \theta(t))^T
\]

(14)

In matrix form the tracking error vector \( \hat{X}(t) \) is

\[
\hat{X}(t) = \begin{bmatrix} x_r(t) - x(t) \\ y_r(t) - y(t) \\ \theta_r(t) - \theta(t) \end{bmatrix} = \begin{bmatrix} \hat{X}_x(t) \\ \hat{X}_y(t) \\ \hat{X}_\theta(t) \end{bmatrix}
\]

(15)
The tracking error \( e(t) \) is formulated as follows

\[
e(t) = E(\ddot{x}(t))
\]  

(16)

where \( E(\ddot{x}(t)) \) is a function defined as follows

\[
E(\ddot{x}(t)) = \sqrt{\ddot{x}_x^2(t) + \ddot{x}_y^2(t) - d_0}
\]  

(17)

The value of \( d_0 \) in (17) is a small positive constant.

Remark 2: the role of \( d_0 \) in (17), is to let the WMR maintains a small distance behind the reference path point \( q_r(t) \).

It is desired that the limit of the tracking error \( e(t) \) tends to zero as the time increases:

\[
\lim_{t \to \infty} e(t) = \lim_{t \to \infty} \ddot{x}(t) \approx 0
\]  

(18)

The tracking error \( e(t) \) is regulated to zero by adjusting the robot linear velocity \( v(t) \) with a PD controller

\[
v_4(t) = k_p e(t) + k_d \frac{de(t)}{dt}
\]  

(19)

\( k_p, k_d \) are non-negative coefficients, denote the proportional, and the derivative terms respectively.

The reference path angle is computed as follows

\[
\theta_r(t) = \tan^{-1}\left(\frac{y_r(t) - y(t)}{x_r(t) - x(t)}\right)
\]  

(20)

The angular velocity input is

\[
w_4(t) = k_4 \ddot{x}_\theta(t)
\]  

(21)

Where \( k_4 \) is a proportional gain.

A. Fuzzy PD Controller Design

To improve the performances of the path following controller, a fuzzy PD controller is developed. The fuzzy PD controller is designed based on fuzzy logic approach. Fig. 4 shows the complete structure of the path following controller.
In Fig. 4 the PD controller is used first then disconnected from the system and the developed fuzzy PD controller is connected to the switch to control the linear velocity of the WMR. The angular velocity of the WMR is controlled by a simple proportional controller.

The overall fuzzy PD controller is used for regulating the tracking error $e(t)$ and the change rate error $ec(t)$, where the error $e(t) = E(\hat{X}(t))$ is calculated using (16) and $ec(t)$ is the derivative of $e(t)$ computed as follows

$$ec(t) = \frac{de(t)}{dt}$$  \hspace{1cm} (22)

At each instant time $t=k$: the linear velocity $v_s(k) = u(k)$. The real control signal $u(k)$ is:

$$u(k) = u(k - 1) + \Delta u(k)$$  \hspace{1cm} (23)

The fuzzy PD controller receives the inputs $e(k)$ and $ec(k)$ and produces the incremental control signal $\Delta u(k)$. Here we consider that the output of the fuzzy PD controller is $u(k)$.

The fuzzy logic PD controller is built with four distinct components, fuzzification unit, fuzzy rule base (FRB), fuzzy inference mechanism and defuzzification unit.

a) Input Fuzzification / Output Defuzzification Units

A fuzzification unit is used to transform the nonfuzzy input values $e(t)$ and $ec(t)$ into a form which can be expressed by the fuzzy rule base and can be compared with the other rules.

The defuzzification step is used to transform the conclusion derived from the fuzzy inference mechanism into output value $u(k)$ of the fuzzy PD controller.

Fuzzification and defuzzification units require heuristic rules and membership functions to encode the desired system.
b) Membership Functions Design

The membership functions of the inputs \((e(k), ec(k))\) and the output \(u(k)\) linguistic variables are displayed in fig. 5 and fig. 6 respectively. Each linguistic variable is decomposed into seven fuzzy partitions expressed as NB, NM, NS, ZO, PS, PM, PB.

\[ e(k), ec(k) \text{ (7 Mememberships)} \]

**Figure 5. Membership functions of input variables**

\[ u(k) \text{ (7 Mememberships)} \]

**Figure 6. Membership functions of output variable**

The linguistic meanings of the fuzzy partitions are: NB: negative big, NM: negative middle, NS: negative small, ZO: zero, PS: positive small, PM: positive middle, PB: positive big.

The universe of discourse for the inputs \(e(k)\) and \(ec(k)\) membership functions are taken between the intervals \([-6, +6]\) as shown in fig. 5.

The universe of discourse for the output \(u(k)\) membership functions is taken between the interval \([-4, +4]\) as shown in fig. 6.

c) Fuzzy Rule Base Design

The fuzzy controller rules can be represented as a mapping from input linguistic variables \(e(t)\) and \(ec(t)\) to output linguistic variable \(u(t)\).

At instant time \(t=k\) the control signal \(u(k)\) is

\[ u(k) = \text{fuzzy PD controller} \ (e(k), ec(k)) \quad (24) \]
The fuzzy rule base system is composed with collection of the *IF-THEN* rules, where the \( j \text{th} \) rule, is in the following form:

\[
R^j : \text{if } x_1 \text{ is } F^j_1 \text{ and } x_n \text{ is } F^j_n \text{ THEN } y^j \text{ is } C^j, \text{ where } x = [x_1, x_n]^T \text{ is the input vector and } y \in R \text{ is the output.}
\]

By using the singleton fuzzification, the product inference engine and the weighted average method for the defuzzification strategy, the output value of the fuzzy system is given as follows

\[
y(x) = \frac{\sum_{j=1}^{m} y^j \left( \mu_{u_j}^i \left( x \right) \right)}{\sum_{j=1}^{m} \left( \mu_{u_j}^i \left( x \right) \right)} \tag{25}
\]

where \( j=1, \ldots, 49 \), the number of fuzzy rules in the fuzzy rule base system.

\( y^j \) Represents a crisp value at which the output membership function \( \mu_{u_j}^i \) reaches its maximum value,

\[
\mu_{u_j}^i (y^j) = 1 \tag{26}
\]

The complete structure of the Mamdani fuzzy inference system is shown in fig.7, it is structured as 49 fuzzy rules which are the combination result of the 7 memberships functions for the tracking error \( e(k) \) and the 7 membership functions of the error change rate \( ec(k) \).

The 49 fuzzy rules are designed as in table I. The Matlab fuzzy control toolbox is used to input the corresponding fuzzy rules.

The same for all the rules, the format of the fuzzy reasoning of rule 1 in the table I is: if the error \( e \) is \( NB \) and the error change rate \( ec \) is \( NB \), then \( u \) is \( ZO \).

![System structure: 2 inputs, 1 output, 49 fuzzy rules](image)

<table>
<thead>
<tr>
<th>( e(k) )</th>
<th>( ec(k) )</th>
<th>( NB )</th>
<th>( NM )</th>
<th>( NS )</th>
<th>( ZO )</th>
<th>( PS )</th>
<th>( PM )</th>
<th>( PB )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( NB )</td>
<td>( ZO )</td>
<td>( PS )</td>
<td>( PM )</td>
<td>( PB )</td>
<td>( PB )</td>
<td>( PB )</td>
<td>( PB )</td>
<td>( PB )</td>
</tr>
<tr>
<td>( NM )</td>
<td>( NS )</td>
<td>( ZO )</td>
<td>( PS )</td>
<td>( PM )</td>
<td>( PB )</td>
<td>( PB )</td>
<td>( PB )</td>
<td>( PB )</td>
</tr>
<tr>
<td>( NS )</td>
<td>( NM )</td>
<td>( NS )</td>
<td>( ZO )</td>
<td>( PS )</td>
<td>( PM )</td>
<td>( PB )</td>
<td>( PB )</td>
<td>( PB )</td>
</tr>
<tr>
<td>( ZO )</td>
<td>( NB )</td>
<td>( NM )</td>
<td>( NS )</td>
<td>( ZO )</td>
<td>( PS )</td>
<td>( PM )</td>
<td>( PB )</td>
<td>( PB )</td>
</tr>
<tr>
<td>( PS )</td>
<td>( NB )</td>
<td>( NB )</td>
<td>( NM )</td>
<td>( NS )</td>
<td>( ZO )</td>
<td>( PS )</td>
<td>( PM )</td>
<td>( PB )</td>
</tr>
<tr>
<td>( PM )</td>
<td>( NB )</td>
<td>( NB )</td>
<td>( NB )</td>
<td>( NM )</td>
<td>( NS )</td>
<td>( ZO )</td>
<td>( PS )</td>
<td>( PM )</td>
</tr>
<tr>
<td>( PB )</td>
<td>( NB )</td>
<td>( NB )</td>
<td>( NB )</td>
<td>( NM )</td>
<td>( NS )</td>
<td>( ZO )</td>
<td>( PS )</td>
<td>( PM )</td>
</tr>
</tbody>
</table>

**Table I**

**The Fuzzy Rules of The Fuzzy Inference System**
The tuning of the conventional PD controllers is made by selecting the gains $k_p$ and $k_d$ as in (19). But in fuzzy PD controllers, the tuning is done by selecting the position of the membership functions.

The performance of a fuzzy PD controller is illustrated pictorially by the phase portrait, shown in fig. 8, with axes $e^o(k)$ and $ec^o(k)$. The symbol "^o" denotes normalized values in the interval [-1, 1].

Definition 2: A phase portrait is a geometric representation of the trajectories of a dynamical system in the phase plane.

The various regions of the phase plane illustrated in fig. 8 correspond to positions in table I of the fuzzy PD controller.

Typically, the desired performance of a fuzzy PD controller is the one shown in Fig 8, that is, $(e^o(k), ec^o(k))$ converges asymptotically to the equilibrium point $(0,0)$.

![Phase Portrait](image.png)

Figure 8. Trajectory of the error dynamics state [$e(k)$, $ec(k)$] of the fuzzy PD controller.

VII. SIMULATION RESULTS

We have taken different examples to test the different algorithms presented in the previous sections for controlling the differential drive WMR on the X-Y plane.

A. Moving To Goal Point

Assume the initial pose of the WMR at $t=0$ is $q_0 = (8, \frac{\pi}{2})$.

The goal point is $q_{goal} = (3, 3)$. The input velocities are given by (3) and (5).

The simulation result for the move to a goal is illustrated in fig.9 where the WMR moves from its initial pose to the goal point.

See that the WMR effectively move toward to the goal point. Moreover the fig. 10 and fig. 11 display a good convergence of the inputs linear and angular velocities of the WMR.
B. Moving to a Specific Pose

Assume the initial pose of the WMR at $t=0$ is $q_0 = [6, 5, \pi]$. It is desired that the WMR moves to the specific pose $[G]$ with coordinates $q_g = [3, 2, \pi/2]$.

The controller parameters of equation 7 are $k_p = 3$, $k_\alpha = 8$, $k_\beta = -3$.

The values of the parameters $\alpha, \beta, \rho$ generated from the simulation results are illustrated in fig.12. The WMR’s path is shown in fig. 13. The WMR moves smoothly from its starting pose to the goal pose $q_g$.

The time history of angle displayed in fig.14 demonstrate that the WMR angle starts from the initial heading value and ends at the goal pose heading value.
C. Line Following

Assume a WMR moving on the X-Y plane by a constant linear velocity $V$. It is desired to follow a target line defined by the equation $x - 2y + 3 = 0$. Assume the initial pose of the robot at $t=0$ is $q_0(0) = [8, 8, \pi]$.

Simulation result of fig.15 shows that the WMR smoothly follows the target line. Note in fig 16 and fig.17, the time histories of the distance and heading angle of the WMR demonstrate good results.

![Figure 12. The polar coordinates values](image12)

![Figure 13. Moving to specific pose path](image13)

![Figure 14. The time history of the WMR angle](image14)

![Figure 15. Target line following](image15)
D. Path Following

Two paths have been chosen for the simulation: a circle path and an infinity shape path. For each path, the WMR is controlled firstly by a PD controller as shown in fig.4 then disconnected, and the fuzzy PD controller is connected to the switch.

D.1 First Path (Circle Shape)

Reference path is defined by the following equations

\[
\begin{align*}
    x_r(t) &= R \cos(wt) \\
    y_r(t) &= R \sin(wt)
\end{align*}
\]

Where the radius \( R=1 \), and the angular frequency \( w = 2\pi f = \frac{n}{5} \text{rad/s} \).

The initial state of the WMR at \( t=0 \) is \( q_0(0) = \left[ \frac{1}{2}, 0, \frac{n}{4} \right] \).

Fig. 18 and fig. 19 illustrate the time histories for the variables of configuration. Note that \( x \) and \( y \) can track the reference path \( x_r \) and \( y_r \), respectively. The plot of the tracking errors function \( e(t) \) of (16) are illustrated in fig 20. See that the errors converge to very small values as time increases.

Fig. 21 displays the WMR path following for the reference path on the \( X-Y \) plane for both PD and fuzzy PD controllers.
Remark 3: at the initial state, it is considered that the WMR is at rest and stable. It is desired that the whole system presents a stable behavior such that the WMR leaves its initial state then follows the reference point.

Note in fig.21, in case of the PD controller, the WMR leaves its initial state but goes over the reference path. However in case of the fuzzy PD, the WMR leaves its initial state and keeps following the reference path.

D.2 Second Path (Infinity Shape)

We take the initial pose of the WMR at \( t = 0 \) is \( q_0(0) = \begin{bmatrix} 0 & \frac{1}{2} & \frac{5\pi}{2} \end{bmatrix} \).

The infinity path is defined by the following equations

\[
\begin{align*}
x_r(t) &= R\sin(wt) \\
y_r(t) &= R\sin(2wt)
\end{align*}
\]
Where $R=1$ and $\omega = \frac{\pi}{7}$ rad/s.

Fig. 22 and fig. 23 display the time histories for the variables of configuration. It is obvious that fuzzy $PD$ controller can track the reference path $x_r$ and $y_r$ along the $x$-axis and $y$-axis respectively.

The plot of the tracking errors function $e(t)$ of (16) are illustrated in fig 24. It is observed that the fuzzy $PD$ presents good error convergence.

Fig. 25 displays the $WMR$ path following for the reference path on the $X-Y$ plane for both $PD$ and fuzzy $PD$ controllers.
The analysis of the results shown in fig.25, indicates that the WMR becomes instable when it is controlled by a PD controller, because it leaves the initial state and goes away from the reference path. However satisfied results obtained using a fuzzy PD controller, where the WMR follows smoothly the reference infinity path.

VIII. CONCLUSIONS

This paper presented different algorithms for controlling a nonholonomic differential drive wheeled mobile robot. Each algorithm designed the appropriate linear and angular input velocities for the WMR kinematics model such that the WMR can successfully accomplish the following tasks: driving the WMR to a goal point, moving to a specific pose, target line following and path following. Simulation examples are performed to test each algorithm.

The algorithm for driving the WMR to a goal point \((x^*, y^*)\) on the \(X-Y\) plane is simple and effective, it is verified by simulation example. Polar coordinates transformation are used to design the suitable controller for moving the WMR to a specific pose \((x^*, y^*, \theta^*)\) on the \(X-Y\) plane, also it is verified by simulation results. Another algorithm is following a target line given by any equation of the form \((ax+by+c)\), the algorithm is efficient while the WMR is moving with constant speed, as demonstrated by simulation example.

Compared to the previously presented tasks, the challenging task for the WMR is path following. At the beginning the input velocities are designed such that the tracking error function is reduced to zero. The linear velocity input of the WMR is firstly controlled by a PD controller, but the efficiency of the PD controller is limited, therefore a fuzzy PD controller is developed in this paper to improve the effectiveness of the path following controller. The fuzzy PD controller is designed based on fuzzy logic approach. The main advantages of the fuzzy PD are the following: providing more stability to the path following control system and the WMR can track faster the reference path in the \(X-Y\) plane. Simulation results demonstrated the effectiveness of the fuzzy PD controller compared to the PD controller. Different paths and initial states for the WMR are taken as examples for testing the fuzzy PD controller and the obtained results showed better tracking performances using the developed fuzzy PD controller.

Generally, due to the simplicity and efficiency of the presented algorithms for controlling a nonholonomic differential drive WMR to perform several useful tasks, they can be used in real time applications like control and navigation of mobile robots. In the future works we may consider obstacle avoidance and motion planning.
REFERENCES