Real-Time Fuzzy-PID for Mobile Robot Control and Vision-Based Obstacle Avoidance

Sabrina Mohand Saidi, L2CSP Laboratory, Algeria
Rabah Mellah, L2CSP Laboratory, Algeria
Arezki Fekik, Akli Mohand Oulhadj University, Bouira, Algeria & Electrical Engineering Advanced Technology Laboratory (LATAGE), Mouloud Mammeri University, Tizi-Ouzou, Algeria & Automated Systems and Soft Computing Lab (ASSCL), Riyadh, Saudi Arabia*

Ahmad Taher Azar, College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia & Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt

ABSTRACT

In this work, the differential mobile robot is controlled utilizing fuzzy PID speed control, which combines fuzzy control with conventional PID control in real time. The path may be convoluted, and the surrounding environment may contain a range of arbitrary shape and size obstacles. A monocular camera is used to detect obstacles during the navigation process. To enable a robot to travel within an indoor space while avoiding obstacles, a basic image processing approach based on area of interest was used. The goal of this research is to develop a fuzzy PID speed controller on a real robot, as well as a simple and efficient visual obstacle avoidance system. MATLAB is used to implement the control system. GUIDE (graphical user interface development environment) has enabled the creation of graphical user interfaces. These interfaces make it easy to manipulate the system in real time and capture live video. The proposed methodologies are tested on a non-holonomic dr robot i90 mobile robot, and the results are satisfactory.

KEYWORDS

Differential Mobile Robot, Fuzzy PID Control, Speed Control, Trajectory Tracking, Visual Obstacle Avoidance

1. INTRODUCTION AND RELATED WORK

Despite the fact that the first tests with mobile robots date back to the late 1960s, the topic did not attract major attention until the 1990s. A significant amount of research has been published. Mobile Robots have proven their worth on land, in the air, at sea, and in submarines, and are now used in industries, security, first aid, personal support, and exploration (Khamis et al., 2022, 2021; Ajeil et al., 2020a,b; Ibraheem et al., 2020a,b; Ammar et al., 2020a,b; Barakat et al., 2020). Mobile robotics is clearly at the forefront of technological innovation, with companion robots, personal assistance robots, and even robotic transportation systems. The majority of the research is devoted to creating and developing robot motion (Krić et al., 2011; Mahfouz et al., 2013), path planning (Kayacan and Chowdhary, 2019; Matoui et al., 2019), map building (Jia et al., 2010), obstacle avoidance (Trinh
et al., 2022), object tracking (Yilmaz et al., 2006, and speed control (Ng et al., 2012; Sharma and Jain, 2016). Mobile robots, for example, can map their surroundings, develop collision-free dynamic settings, and plan safe approaches to humans. The majority of research has been focused on applying dynamic models to build and implement mobile robot control (Sharma and Jain, 2016; Park et al., 2017; Sun et al., 2016). Others may not see the need to use the dynamic model or may lack access to specific information essential for its use, hence the kinematic model is widely used (Krivić et al., 2011; MohandSaïdi and Mellah, 2019; Lang et al., 2010).

The differential mobile robot is becoming increasingly vital and broadly used in human daily lives. Because of its ease of assembly and intriguing kinematic features, this sort of robot is quite popular. As a result, they are widely used. In recent years, there has been an upsurge in and interest in research on differential mobile robots (Krivić et al., 2011; Lopez-Franco et al., 2015). Because lateral translation is impossible, non-holonomic mobile robots have just two degrees of freedom on a plane (Krivić et al., 2011; Matoui et al., 2019). Control with PID is a technique used in engineering to reduce process variability by altering some quantifiable system variables with feedback and compensation. Since its conception in 1910 and the presentation of the tuning rules by Ziegler–Nichols in 1942, PID control has risen in popularity significantly. PID Control is still used in almost every field. A great amount of time and effort has gone into determining the best PID settings for various process models. Ahmad et al. (2019) and Li et al. (2006) conducted a comparative study of PID controller tuning approaches. PID parameter fine tuning remains a difficult task. The majority of research is focused on the use of intelligent methods to modify the settings of this controller, such as neural networks (Rossomando and Soria, 2015; Fan and Chen, 2013) and fuzzy logic (Kumbasar and Haigras, 2014) or both techniques (Zerfa and Nouiba, 2013).

The hybrid fuzzy PID controller combines the advantages of the fuzzy controller with the ease of use and resilience of the standard PID controller. It is utilized in a variety of applications, including pesticide residue detection (Jun et al., 2015) and system control for water purification (Mashhadi et al., 2016). It is also frequently used to control differential mobile robots, particularly in speed control. A fuzzy-PID double closed-loop speed control system was proposed by Wu et al. (2013). ANFIS, fuzzy, and PID speed controllers for wheeled mobile robots are developed and compared in Khan et al. (2022). Yousfi et al. (2021) proposed a hybrid fuzzy logic PID based dynamic model controller to assure target achievement and trajectory tracking. Yousfi et al. (2021) offered two control strategies for non-holonomic mobile robots evolving in environments with multiple external effects, one based on a PD fuzzy logic controller and the other on a smart PID optimized neural networks-based controller. A non-holonomic autonomous wheeled robot is proposed by Ben Jabeur and Seddik (2021) that follows a predefined path using a combination of back stepping and an adaptive fuzzy PID approach. Zanga et al. (2021) presented a fuzzy fractional-order PID controller for path following, which use fuzzy logic to dynamically set PID parameters. Ammar and Azar (2020b) proposed a fractional order PID to control Pioneer-3 Mobile Robot path tracking. Ratnayake et al. (2019) disucssed the effectiveness of the PID control and the Fuzzy control in wall following for differential drive mobile robots. Campos et al. (2018) proposed an optimal fuzzy PD + I controller for the trajectory of the unicycle robot. To optimize and obtain the ideal controller parameter values, the PSO (Particle Swarm Optimization) approach is used. All of the preceding work on wheeled mobile robot control has been validated in simulation, but the behavior of a mobile robot in a real environment is not always the same in simulation, and there are few validated studies with experimental data. Boukens et al. (2018) presented a self-tuning method based on the ant colony methodology and fuzzy control for real-time controller parameter change for mobile robots.

Hartonos and Nizar (2019) developed a fuzzy logic speed controller for a wheeled robot that can follow predetermined points and respond swiftly to achieve a predefined speed. The system’s input is the robot speed error with variation, and the system’s output is 8-bit pulse width modulation translated to fuzzy forms. Later research focused on robots with additional sensing systems to construct autonomous guidance systems to maneuver a wheeled mobile robot securely in its surroundings.
(Alhalabi et al., 2021). Guzel and Bicker (2011) described obstacle avoidance methods based on vision. Detecting objects is a critical part of obstacle avoidance (Cyganek, 2013). The difficulty of computing avoidance algorithms and the cost of sensors are the most relevant factors for real-time applications. Range finders with lasers, ultrasonic sensors, and stereo vision technologies are often used to identify barriers. All of them, however, have drawbacks. Ultrasonic sensors have an issue with angular resolution. The price of a laser range finder plus a stereo vision system is quite significant. Many mobile robots use cameras since they are sensors that provide additional information about their environment.

The method of control based on information provided by one or more cameras is referred to as visual servonning. Pasteau et al. (2016) demonstrated vision-based electric wheelchair control. Tsalatsanis et al. (2007) presented stereo vision for real-time object tracking and collision avoidance. Stereo vision Guidance for Robot is also used by Woodward et al. (2010). Benn and Lauria (2012) proposed a monocular vision to navigate a robot in dynamic and changing situations. Visual feedback is utilized to track a moving target by Boumehraz et al. (2018). Nadour et al, (2019 a) used a mobile robot for navigation and obstacle avoidance. There are two types of vision-based obstacle avoidance techniques. The first is based on the optical flow principle (Nadour et al., 2019 a,b), which suffers from shifting light and motion discontinuities created by objects moving in respect to their surroundings. The second step entails identifying pixels that differ from the appearance of the ground and categorizing them as obstacles (Adar et al., 2015). The two main advantages of this strategy are its ease of use and availability for real-time applications.

The goal of our method is to develop and implement a speed controller for a real wheeled mobile robot that uses real-time self-adjustment of PID parameters via a fuzzy regulator with two inputs: speed error and variation of speed error, and three outputs: \( k_p \), \( k_i \), and \( k_d \), which are the PID regulator parameters. The regulator’s settings will change with each sample interval in order to adapt to the change in speed detected by the quadratic encoder sensor. This research also proposes a novel image-based technique for obstacle avoidance of a mobile robot traveling within an interior space. The simplicity of this technique is what makes it unique. This approach is based on image processing and detects impediments by using background subtraction. When an impediment is spotted, the robot comes to a halt and pulls the information needed to calculate the avoidance movement from the image. The main advantage of the proposed method is that it does not require determining the robot’s distance from the obstacle.

This paper is organized as follows. Section 2 of this work provides modeling and a brief description of the dr robot i90. Section 3 outlines the suggested fuzzy PID controller’s design. Section 4 introduces trajectory tracking, and Section 5 introduces collision avoidance. Section 6 presents and discusses experimentally obtained results. The seventh section summarizes some major findings and viewpoints.

2. MODELING OF THE MOBILE ROBOT

The use of robots has grown dramatically in recent years. They are now utilized for a wide variety of purposes and can be found in a variety of settings, including industrial, home, educational, and healthcare settings (Vaidyanathan and Azar, 2020; Azar and Kamal, 2021; Koubaa and Azar, 2021a,b; Ajel et al., 2021; Al-Qassar et al., 2021a,b; Bansal et al., 2021; Azar et al., 2022, 2021a,b,c,d,e,f,g,h,i,j, 2020a,b,c,d; Kazim et al., 2021a,b; Najim et al., 2021a,b, 2020; Elkholy et al., 2020a,b; Mohamed et al., 2020; Ibrahim et al., 2020; Sayed et al., 2020; Samanta et al., 2018 ; Mukherjee et al., 2014). In the case of mobile autonomous robots, they must be able to navigate through an unknown area while also doing the purpose for which they were built. As a result, the robot must be capable of creating an environment model, calculating its present position and orientation inside the environment using this model, and navigating throughout the environment to reach the target spots.
2.1 The Dr. Robot i90 Description

The Dr. Robot i90 is a differential wheeled mobile robot designed for the development of advanced applications with completely wireless connectivity. It weighs only 5 kg but can carry a payload of up to 15 kg. It measures 43 cm in width, 38 cm in length, and 30 cm in height. It is capable of reaching a top speed of 75 centimeters per second. This robot has a high-resolution camera and two DC motors that allow it to navigate about its environment, as well as quadratic encoders placed in the driving wheels that offer incremental angle measurements (Dr Robot i90, 2010). The driving pilot element of the Dr. Robot i90 is the PMS5005 card, which is designed to work with the WiRobot system. The WiRobot software development kit allows PC programs to communicate with the robot (Wi Robot SDK, 2010; PMS 5005, 2006). Figure 1 depicts a photograph of Dr. Robot i90 as well as the communication architecture between the robot and the computer.

2.2 Unicycle Type Mobile Robot Description

A robot is defined as a unicycle when it has two independent wheels and eventually some of idle wheels to ensure its stability. Figure 2 depicts a unicycle-type robot. Idle wheels are not included because they do not affect the kinematics if set correctly.
2.3. Model of Kinematics

This robot is a two-wheeled platform with differential wheels that allows it to move. DC motors serve as actuators for the robot’s two wheels, allowing it to move and be orientated. To keep the platform balanced, a caster wheel is employed. The kinematics equations of a robot moving at linear velocity \( v \) and angular velocity are supplied by:

\[
\dot{x}(t) = v(t) \cos(\theta(t))
\]

\[
\dot{y}(t) = v(t) \sin(\theta(t))
\]

\[
\dot{\theta}(t) = \omega(t)
\]

\( x \) and \( y \) indicate the world coordinates of the robot gear’s center and \( \theta \) is the orientation angle. This model has been utilized in several articles, including (Krivić et al., 2011; Matouï et al., 2019; Serrano et al., 2016; Scaglia et al., 2009; Shamsfakhr and Bigham, 2017).

The actuation model shows the robot’s velocities as a function of its driving wheel velocities and geometric properties, and it is given by equations (4) and (5).

\[
v(t) = \frac{v_r(t) + v_l(t)}{2}
\]

\[
\bar{E}(t) = \frac{v_r(t) - v_l(t)}{D}
\]

The distance between steered wheels is denoted by the letter \( D \). \( v_r \) and \( v_l \) are the velocities of the right and left wheel respectively, reflecting inputs of the kinematic model (Mahfouz et al., 2013). The robot’s mobility is defined by two non-holonomic restrictions (Matouï et al., 2019; Corke, 2017), which are determined by two primary hypotheses:

Hypothesis 1:

There is no lateral slippage. It simply indicates that the robot is unable to move laterally in its coordinate system, as expressed by equation (6):

\[
\dot{y}_{robot}(t) = 0
\]

Hypothesis 2:
Each of the robot’s wheels retains one point of contact with the ground. The wheel does not slide in either of its longitudinal or orthogonal axes.

At discrete time $t = nT_0$, $x(t), y(t), v(t)$ and $\omega(t)$ will be designated as $x_n, y_n, v_n$ and $\omega_n$ respectively (Ng et al., 2012; Serrano et al., 2016; Scaglia et al., 2009). It follows from The kinematics equations that

$$\begin{align*}
x_{n+1} &= x_n + \int_{nT_0}^{(n+1)T_0} v(t) \cos \theta \, dt \\
y_{n+1} &= y_n + \int_{nT_0}^{(n+1)T_0} v(t) \sin \theta \, dt \\
\theta_{n+1} &= \theta_n + \int_{nT_0}^{(n+1)T_0} w(t) \, dt
\end{align*}$$

The following equations (8) are generated using the Euler approximation of the mobile robot’s kinematic model (Serrano et al., 2016). The use of discrete-time equations simplifies the computer system implementation in real-time.

$$\begin{align*}
x_{n+1} &= x_n + T_0 v_n \cos \theta_n \\
y_{n+1} &= y_n + T_0 v_n \sin \theta_n \\
\theta_{n+1} &= \theta_n + T_0 \omega_n
\end{align*}$$

3. DESIGN OF THE PROPOSED CONTROL

Many significant advances in the design of nonlinear systems for a wide range of practical applications have occurred in recent decades. Several inspiring approaches, such as optimal control, nonlinear feedback control, adaptive control, sliding mode control, nonlinear dynamics, chaos control, chaos synchronization control, fuzzy logic control, fuzzy adaptive control, fractional order control, and robust control, as well as their integrations, have been proposed (Fekik et al., 2022, 2021a,b,c, 2020a, 2019, 2018a,b,c; Daraz et al., 2021; Pilla et al., 2021a; Abdul-Adheem et al., 2021, 2020a,b; Liu et al., 2020; Bouchenha et al., 2021; Gorripotu et al., 2021; Drhorhi et al., 2021; Alimi et al., 2021; Kumar et al., 2021; Acharyulu et al., 2021; Hamiche et al., 2021; Mittal et al., 2021; Pham et al., 2021, 2018, 2017a,b; Sambas et al., 2021a,b; Khan et al., 2020a,b; Khennaoui et al., 2020a,b; Kammogne et al., 2020; Alain et al., 2020, 2019, 2018; Ouannas et al., 2021, 2020a,b,c,d,e,f, 2019a,b,c, 2017a,b,c,d,e,f,g,h,i,j,k, 2016; Ammar et al., 2019, 2018; Meghni et al., 2017a,b; Singh et al., 2018a,b,c, 2017; Ben Smida et al., 2018; Lamamra et al., 2017; Grassi et al., 2017; Ghouldebhour et al., 2016; Mekki et al., 2015; Tolba et al., 2017a,b; Soliman et al., 2017; Wang et al., 2017; Humaidi et al. 2022, 2020; Abdelmalek et al., 2018a,b; Shawla et al., 2018; Azar et al., 2017; Azar and Serrano, 2015; Vaidyanathan et al., 2021a,b,c,d,e,f, 2019, 2018a,b,c,d,e,f, 2017a,b,c, 2015a,b,c; Vaidyanathan and Azar, 2021, 2020, 2016a,b,c,d,e,f, 2015a,b,c,d; Santoro et al., 2013).

The traditional PID (Proportional-Integral-Derivative) controller has a simple structure; it is widely used in the control community due to its simplicity, robustness, and familiarity (Ng et al., 2012; Nguyen et al., 2016; Pilla et al., 2021b, 2020, 2019; Ibraheem et al., 2021; Rana et al., 2021; Najm et al., 2021c; Soliman et al., 2020; Sallam et al., 2020; Gorripotu et al., 2019; Ammar et al., 2018).
However, it is not suitable for a large inertial system, and matching the optimal parameters is
difficult. A PID’s coefficients $k_p$, $k_i$, and $k_d$ can be modified experimentally via test and error (Sharma
and Jain, 2016; MohandSaidi and Mellah; 2022). First and foremost, understand that attempting to
alter all three coefficients at the same time is pointless. There are far too many conceivable
combinations and obtaining a nice triplet would be difficult. Traditional PID controllers are notoriously
difficult to tune to achieve the perfect state. The PID controller’s mathematical form is

$$c(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}$$

(9)

$k_p$, $k_i$, and $k_d$ are PID’ control parameters. $e(t)$ is the error between desired and measured value and$c(t)$ is the signal of control.

The PID controller’s parameters are determined by the self-tuning fuzzy logic controller. Other
than mathematics, fuzzy logic provides a significantly more pragmatic approach. In particular,
where mathematics fails owing to the impossibility or complexity of modeling, fuzzy logic performs
admirably. It enforces a set of rules that we all follow on a daily basis. The fuzzy logic controller
(FLC) is a system that is inspired by human reasoning abilities and is based on linguistic data.

The concept of fuzzy logic control is not new; it arises from efforts to develop artificially intelligent
decision-making and inference systems (Sain et al., 2022 ; Shalaby et al., 2021; Ghouldebourk et al.,
2021; Humaidi et al., 2021; Nasser et al., 2021; Mohanty et al., 2021; Fekik et al., 2021d, 2020b;
Ahmadian et al., 2021; Abdelmalek et al., 2021; Ananth et al., 2021; Azar et al., 2013; Mjahed et al.,
2020; Djeddi et al., 2019; Amara et al., 2019; Vaidyanathan and Azar, 2016; Kumar et al., 2018; Khettab
et al., 2018; Pintea et al., 2018; Banu et al., 2017; Emary et al., 2014; Gharbia et al., 2014; Giove et al.,
2013; Elshazly et al., 2013; Jothi et al., 2013). The structure of fuzzy inference is comparable to that of
a human judgment process: incoming signals are evaluated subjectively and in a fuzzy manner. Output
signals are influenced by inference rules, which are thought to be a set of fuzzy rules developed from
personal knowledge of the topic under study. Fuzzy rules are developed based on the system’s various
operating conditions, taking into account the proportional, integral, and derivative coefficients and
changing the three parameters with the fuzzy rule. The accuracy of a fuzzy control system is influenced
by expert rules and membership functions. As a result, they must be picked with caution.

The primary goal of this research is to design a PID control self-tuned by a fuzzy regulator with
two inputs, the speed error e and the variation of the speed error $e$ acquired via quadratic encoder sensor,
at each sampling period, and three outputs, $k_p$, $k_i$, and $k_d$ parameters of the PID controller adjusted in
real-time at each sampling period. Each of the left and right wheel DC motors is controlled by a fuzzy
PID controller. Figure 3 depicts a schematic of the fuzzy PID control applied to each wheel. The stability
of the PID controller is based on the positivity of the PID parameters where the design of a stable fuzzy
PID controller for DC motors system was done based on the passivity theorem (Sio and Lee, 1998).

$$k_p > 0 , k_i > 0 , k_d > 0$$

The linguistic values of $e$ and "$e$ are [NB, NM, NS, Z, PS, PM, PB], which correspond to
[Negative Big, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium, Positive
Big]. The range domains of $e$ and "$e$ are [-50 50] and [-100 100], respectively. The linguistic value
$k_p$, $k_i$, and $k_d$ outputs of the fuzzy controller are [Z, S, M, B] corresponding to [Zero, Small, Medium,
Big]. The range domains of $k_p$, $k_i$, and $k_d$ are [0 25], [0 50], [0 10], respectively.
Standard triangle membership functions are utilized for the two inputs and three outputs. Figures 4 and 5 depict the membership functions of the fuzzy inputs and fuzzy outputs, respectively.

The fuzzy rules used in the fuzzy controller’s inference mechanism are of the Mamdani If-Then type (Mamdani and Assilian, 1975), and they are based on works of the form:

\[
\text{if } P1 \text{ is } X1 \text{ and } P2 \text{ is } X2, \text{ then } C1 \text{ is } Y1 \text{ and } C2 \text{ is } Y2 \text{ and } C3 \text{ is } Y3
\]
The values of the gains are determined using the center of gravity defuzzification approach. Table 1 includes the guidelines for output parameters based on the experience of experts and references (Wei et al., 2010; Miloud and Abdelouahab, 2016).

Table 1. Rule Table of \( k_p / k_i / k_d \)

<table>
<thead>
<tr>
<th>( k_p / k_i / k_d )</th>
<th>( e )</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>B/B/S</td>
</tr>
<tr>
<td>NM</td>
<td>B/B/S</td>
</tr>
<tr>
<td>NS</td>
<td>M/B/Z</td>
</tr>
<tr>
<td>Z</td>
<td>M/M/Z</td>
</tr>
<tr>
<td>PS</td>
<td>S/M/Z</td>
</tr>
<tr>
<td>PM</td>
<td>S/Z/B</td>
</tr>
<tr>
<td>PB</td>
<td>Z/Z/B</td>
</tr>
</tbody>
</table>


4. TRAJECTORY TRACKING

Path planning refers to the robot’s movement along a predefined path. Before the robot may move, path planning, and trajectory generation must be performed. The robot should move from one location in the workplace to another. There has been much debate on how to determine the trajectory of an autonomous mobile robot. When the environment is completely known and movement can be planned ahead of time, it is said to be structured. When there are uncertainties and movements must be planned on the fly, it is categorized as partially structured. The ability to obtain ambient information via external sensors such as vision, distance, or proximity sensors is referred to as autonomous navigation. The two major groups of approaches that we generally recognize are deliberative and reactive procedures. The idea behind deliberative approaches is to utilize a model of the environment in which the system operates to determine a whole movement (a route or trajectory) between an initial and final location. Reactive approaches rely only on sensor data collected by the robotic system at all times to decide mobility for the next step (Miloud and Abdelouahab, 2016).

The purpose of trajectory tracking problems is to find the control actions that bring the robot to the desired position with a predetermined orientation at each step time. As a result of these combined operations, the robot follows the intended course. To accomplish this purpose, only two control variables are available: the robot’s linear velocity \( \nu \) and rotational velocity \( \omega \). The goal is to identify and values that allow the mobile robot to follow a predetermined path with the least amount of error. The reference velocity vector has the following form based on the matrix domain and equations 1, 2, and 3.

\[
\hat{\mathbf{p}}_{ref} = \begin{bmatrix}
\dot{x}_{ref} \\
\dot{y}_{ref} \\
\dot{\theta}_{ref}
\end{bmatrix} = 
\begin{bmatrix}
v_{ref} \cos \theta_{ref} \\
v_{ref} \sin \theta_{ref} \\
\omega_{ref}
\end{bmatrix}
\]

(10)

The coordinates of the differential wheeled mobile robot are in the global frame and can be transformed to local coordinates using the rotation matrix shown below (Ibraheem et al., 2020):
\[ R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \]  

(11)

In the local robot frame, the vector of position and orientation error \( E \) is (Pawlowski et al., 2001):

\[ E = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = R(\theta)(p_{\text{ref}} - p) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\text{ref}} - x \\ y_{\text{ref}} - y \\ \theta_{\text{ref}} - \theta \end{bmatrix} \]  

(12)

Where \( v_{\text{ref}} \) and \( \omega_{\text{ref}} \) are the reference linear velocity and the reference angular velocity of the robot, respectively, \( p_{\text{ref}} \) and \( p \) are the desired and current position and orientation vectors respectively of the robot.

Based on Luyapunov function (Zanga et al., 2021):

\[ L = \frac{(e_x^2 + e_y^2)}{2} + \frac{(1 - \cos e_\theta)}{k_2} \]  

(13)

with \( k_2 > 0 \) and \( L > 0 \). The planned system achieves asymptotic stability when \( L < 0 \).

\[ \dot{L} = (v_{\text{ref}} \cos e_\theta - v)e_x - \left( \frac{\omega - \omega_{\text{ref}}}{k_2} - v_{\text{ref}} e_y \right) \sin(e_\theta) \]  

(14)

The control law will be designed as

\[ \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} k_1 e_x + v_{\text{ref}} \cos e_\theta \\ \omega_{\text{ref}} + k_2 v_{\text{ref}} e_y + k_3 \sin e_\theta \end{bmatrix} \]  

(15)

That gives

\[ \dot{L} = -k_1 e_x^2 - \frac{k_3}{k_2} \sin^2(e_\theta) < 0 \]  

(16)

\( k_1, k_2 \) and \( k_3 \) are positive real-valued parameters
5. OBSTACLE AVOIDANCE

To be safe, the robot must be able to operate without colliding with or hitting any form of impediment. This necessitates an effective control strategy that guarantees the environment’s inherent unpredictability is overcome. The operation is guided by a perception-action process that is repeated at a high frequency. The camera gathers data about the surroundings (obstructions) and the robot, which is used to compute the avoidance movement. The operation is restarted when the robot completes the movement. As a result, a series of moves leads the robot to the goal without any collisions. A rudimentary vision command is used in this part to attempt to answer this issue.

5.1 Visual Feedback

In many applications, equipping mobile robots with cameras is very advantageous because all information about the surrounding environment is stored in a single image frame. The method of control based on information provided by one or more cameras is referred to as visual servoing. This paper describes the construction of a real-time visual obstacle avoidance system that allows a robot to move safely in an interior setting. The most significant aspect of visual servoing is image processing. Image processing is a calculation that converts one or more input photographs into output data. The most significant aspect of visual servoing is image processing. This converts one or more input images into output data. In this investigation, visual data is collected using a monocular camera mounted directly on the robot. As seen in Figure 6, its motion is induced by the robot’s mobility.

To track features in image sequences, a balance must be achieved between the precision of motion prediction and the related computing cost. Despite the fact that computing performance has improved substantially in recent years, monitoring systems that can record, analyze, and publish real-time findings typically use some sort of simplification to speed up the process. Image processing allows you to change the content of photographs in order to extract important information for a specific purpose. The camera images are 640x400 pixels in size, however viewing the complete image is not required. As a result, the shot is cupped to create a point of interest. It is only interesting to detect impediments on the path to follow while keeping in mind that the camera is fixed and focused at the area of the environment in front of the robot. This area of interest is chosen on the basis of experimentation. As illustrated in Figure 7, the length I is determined by the size of the robot, while the width I is determined by the speed of the robot.

Figure 6. The mobile robot's and the monocular camera's coordinate systems
5.2 Obstacle Detection

Image processing is the process of altering the properties of a photograph in order to either boost its graphical information for human interpretation or make it more suitable for autonomous machine perception. The current image and the reference image, which is the image of the ground on which the robot moves without obstructions, have been removed pixel by pixel. The output is binarized and converted to an absolute value. The obstacle detection method works on the basis of calculating the sum \( S \) of the intensities of the pixels on the white surface and comparing it to an empirically determined threshold indicating the existence of an obstruction \( S_p \). After detecting the object, the binary picture was subjected to two morphological operations: erosion followed by dilatation to address the issue of noise induced by the robot’s movement and variations in illumination. The obstruction’s center of gravity is located, and the obstacle is surrounded by a radius \( R_o \) that will help calculate the radius \( R_m \) of the obstacle avoidance movement.

5.3 Obstacle Avoidance Process

There are four steps to this process.

Step 1: Detect obstacle
Step 2: Compute the radius of avoidance movement, which is in circular form
Step 3: avoid obstacles
Step 4: return to its path

As previously stated, image processing is used to detect obstacles. Its center of gravity will be identified, and the obstacle will be surrounded by a circle of radius \( R_o \), which is depicted in Figure 8 by the arrow in green color. The radius of the obstacle avoidance movement \( R_m \) is computed using equation (17).

\[
R_m = R_o + R_s + D / 2
\]  

\( R_o \) is radius of circle surrounding obstacle, \( R_s \) is the security margin, and \( D \) is the distance between steered wheels, as shown in Figure 8. The robot follows the path drawn by blue line to avoid the obstacle and gets back on track. \( R_o \) is measured using a pinhole camera model (Corke.P,2017).
6. EXPERIMENTAL RESULTS

The dr Robot i90 is used to test the proposed methods’ effectiveness. The robot has two quadratic encoders that measure $v_r$ and $v_l$ at each sampling step $T_s = 0.05$ second. $v_r$ and $v_l$ are velocities of the right and left wheels, respectively. It also has a high-resolution camera that can take 25 400x640-pixel photos per second. Using the actuation model given by equations 4 and 5, Angular velocity $\omega$ and linear velocity $v$ can be calculated knowing that the distance between steered wheels is denoted $L$ equally to 0.315 m. The control system is implemented by MATLAB. GUIDE was used to develop interactive interfaces that allow users to make changes. This interface makes the system simpler to modify in real-time, and live capture is also visible.

6.1 First Experiment

The first experiment is carried out to validate the fuzzy PID controller’s effectiveness in keeping the mobile robot’s velocities constant. The control interface for the fuzzy PID speed control is depicted in Figure 9. Because certain robots operate worse at high speeds, two linear speeds were chosen as the ideal value: an average speed and a high speed. The speed curves of the robot are given in Figure 10(a) and 10(b) for $v_{ref} = 200$ pulses per second and Figure 10(c) and 10(d) for $v_{ref} = 500$ pulses per second.

In this experiment, the speed control of the robot dr robot i90 was compared using a standard PID and a fuzzy PID, and the speed curves clearly indicate the improvement that the fuzzy control brings in terms of stability and precision in maintaining the robot at the required speed. We also notice that, even at high speeds, the robot performs better with fuzzy PID control. The speed curves of the fuzzy PID-controlled robot exhibit minor oscillations, which could be caused by a small difference in radius between the left and right wheels, or by the ground being dusty and not completely level.

A second experiment was carried out to evaluate the effectiveness of the recommended controller. The goal is to allow the mobile robot to follow a predefined trajectory, and for that a set of equations has been created using Euler’s approximation of the mobile robot’s kinematic model. The use of a discrete kinematic model to calculate the robot’s location simplifies the suggested approach’s implementation. A circular trajectory was utilized as the planned route. The robot begins its trajectory at (0, 0) with an initial orientation of zero degree. It progresses with $v_r = 500$ pulses / second and
\[ v_l = 350 \text{ pulses per second}, \quad \text{where} \quad v_r \quad \text{and} \quad v_l \quad \text{are velocities of the right and left wheels}, \quad \text{respectively, with the sampling time of} \quad T_s = 0.05 \text{ second.} \]

Two tests were performed to validate trajectory tracking by comparing the tracking of a conventional PID-controlled robot to the tracking of a fuzzy PID-controlled robot. The results show that the proposed control tracking system performs effectively in tests. This test demonstrates the difference between the PID controls in Figure 10 (e) and the fuzzy PID control in Figure 10 (f), as well as the nearly flawless tracking of the desired trajectory in the fuzzy PID control test in Figure 10. (f).

**Figure 9. User interface created for the experiment of the fuzzy PID controller**

6.1 Second Experiment

As previously stated, dr robot i90 is equipped with a high-resolution camera. To prove that using vision as a system sensing tool is a good idea was to let the robot navigate safely from a starting point to a final position by following a well-defined path without hitting obstacle. The purpose of this test is to have the robot navigate by following a straight line itinerary while simultaneously recognizing the existence of an obstacle in its path. When an obstacle is detected the robot stops and extracts the visual information necessary for the process of obstacle avoidance. Figure 11 shows the user interface created by GUI of MATLAB for this test.

The background subtraction approach is used to detect obstacles. Figure 7 shows how the image was cropped to create an interesting zone with a fixed size of 200x100 pixels. Figure 12 depicts the free obstacles environment as the background image, the image with the obstacle, the binary reference sub-image, and the binary sub-image with the obstacle, as well as the result of the difference between the binary sub-images and the result after applying the morphological filters. Figure 13 displays the identification of the obstruction as well as the location of its center of gravity as indicated by the circle that surrounds it.
Figure 14 displays the experiment used to validate the suggested obstacle avoidance method. It shows a series of images taken throughout the experiment that depict the obstacle avoidance method carried out by Dr. Robot i90. In frame 1, the robot advances and follows its pre-planned trajectory; in frame 2, the robot detects the obstacle and stops; in frame 3, the robot changes direction and follows the circular trajectory calculated from information extracted from the image of the obstacle detected in frames 4 and 5, and stops; in frame 6, the robot changes direction and rejoins its trajectory in frame 7; and in frame 7, the robot changes direction and rejoins its trajectory. In frame 8, the robot continues on its path. The photographs show that the obstacle avoidance challenge was successfully accomplished.

Figure 10. Real time robot velocity using PID control in (a) and using fuzzy PID control in (b) for $v_{ref}=200$ pulses/sec. Real time robot velocity using PID control (c) and using fuzzy PID control (d) for $v_{ref}=500$ pulses/sec. Trajectory tracking of Dr. Robot i90 velocity using PID control (e). Trajectory tracking of Dr. Robot i90 velocity using fuzzy PID control (f).
Figure 11. User interface created for the experiment of obstacle avoidance

Figure 12. Image processing steps

Figure 13. Identification of the obstacle’s center of gravity and circumference
7. CONCLUSION

The proposed controller could be used to control a variety of systems. The application of this controller to a genuine non-holonomic mobile robot showed the usage of fuzzy PID control. This control automatically modifies the PID settings in real time, increasing the sensitivity and accuracy of the system. In terms of precision, stability, and robustness, the results reveal that the fuzzy PID controller surpasses the simple PID controller. Future work attempts to improve the system by shortening the sample duration, despite the fact that a more complex computer and processing system is required. The second experiment, based on image processing by region of interest, allows the mobile robot to travel autonomously. The primary advantage of this method is that it does not require mathematical models of the robot and its surroundings, nor does it require the use of another odometer system. Despite the fact that this method is simple, the trial results have proved its usefulness. Future research should concentrate on extending this method to dynamic impediments.

FUNDING

This research has no funding.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of Prince Sultan University, Riyadh, Saudi Arabia. Special acknowledgement to Automated Systems & Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia. In addition, the authors wish to acknowledge the editor and anonymous reviewers for their insightful comments, which have improved the quality of this publication.
REFERENCES


---

Rabah Mellah received the Doctoral degrees in Automatic Control from USTHB University of Algeria in 2006. He is a professor at the Department of Automatic, UMMTO University, Tizi-Ouzou, Algeria. His research interests include nonlinear system, robotics, teleoperation, fuzzy logic and neural networks.

Arezki Fekik received his BSc, MSc, PhD degrees in Electrical Engineering from Mouloud Mammeri University, Tizi-Ouzou, Algeria, in 2011, 2013 and 2018, respectively. He became a Lecture in 2019 and member in the international group of system control (IGCS) since 2018, member of the Springer conference committee (AISI2018, AML2018, AISI2019, AML2019, AISI2020, AML2020, AICV_2020) and member in the IEEE SMARTTECH 2020 conference. His current research interests include power electronics and its applications such as wind turbines, photovoltaic systems, reliability, harmonics, microgrids and variable speed drives. He has published 32 journal and conference articles and book chapters in the fields of power electronics and its applications. He is the guest editor of the journal WSEAS Transactions on Systems and Control for a special issue entitled Modeling, Control and Optimization in Renewable Energy Systems.