Comparative Evaluation of Fuzzy Maneuvering Controllers Robustness for Differential Drive Mobile Robots

Thiago A. Ushikoshi* Kamilla P. Peixoto**
Lucas L. Carneiro* Pedro H. S. Coutinho***
Thiago P. Chagas* Leizer Schnitman**

* Mechatronics Laboratory, Santa Cruz State University, BA, Brazil (e-mails: thiagoushikoshi@gmail.com, llcl2.lle@gmail.com, tpchagas@uesc.br).
** Graduate Program in Mechatronics, Federal University of Bahia, BA, Brazil (e-mail: kamillap.peixoto@gmail.com, leizer@ufba.br).
*** Graduate Program in Electrical Engineering, Federal University of Minas Gerais, MG, Brazil (e-mail: phcoutinho@ufmg.br).

Abstract:
This work presents a comparison between a Mamdani-type fuzzy controller recently proposed and its improved version applied to maneuvering control of a differential drive mobile robot. The performance of the controllers was verified through simulations, where the robot was subject to disturbances, such as variation of ground friction coefficients, mass and battery level. The robot’s kinematic and dynamic models are used for controller tuning and simulation. Results suggest the improved controller has a better performance in all three cases, since its responses are smoother and more accurate.

Keywords: Fuzzy control; Maneuvering trajectory tracking; Differential drive mobile robot; Nonholonomic robots; Disturbance.

1. INTRODUCTION

Mamdani-type fuzzy logic controllers (FLCs) incorporate expert’s skills knowledge through a set of conditional rules (Nguyen et al., 2019). This specific feature makes it intuitive to design and no mathematical model is required to do so (Nguyen et al., 2019; Levine, 1996). Besides, they usually perform better in nonlinear plants with complex dynamics or those that do not provide sufficient quantitative information, increasing parametric uncertainty (Lee, 1990).

FLCs have been widely applied to trajectory tracking of differential drive mobile robots (DDMRs). For instance, Mac Thi et al. (2015) successfully applied a multi-input and multi-output FLC for trajectory tracking and navigation control of a DDMR, Abadi and Khooban (2015) designed an optimal Mamdani-type fuzzy PID controller, and their results shown adequate precision, even in the presence of parametric uncertainties. Similarly, Ushikoshi et al. (2018) analyzed two Mamdani-type FLCs (F-Controller and FB-Controller) for maneuvering tracking of a DDMR. The FB-Controller has the ability to propose bidirectional movements, which can reduce trajectory tracking error, time to complete the path, and control effort.

This paper aims to compare the FB-Controller, previously proposed by Ushikoshi et al. (2018) for DDMRs, with its improved version, called here, FBm-Controller. In contrast to FB-Controller, FBm-Controller has Sigmoidal, Pi and Gaussian membership functions (MFs). These new MFs are employed to provide smoother movements possibly leading to reduction of tracking error. Thus, this study presents a comparison in terms of tracking error, control effort, and ability to compensate disturbances. The analyzed disturbances are variation of ground friction coefficients, robot’s mass variation and reduction of battery level, which are problems frequently found in real applications of mobile robotics. Simulation results suggest that FBm-Controller can indeed perform better than FB-Controller, leading to smoother movements and increasing trajectory accuracy. It is important mentioning that both controllers are designed through simulations performed considering the physical limitations of the robot, which contributes to obtaining suitable controllers for practical applications.

In this paper, the kinematic and dynamic models are used to controller design and to evaluate the effectiveness of the proposed controllers in practical applications. Although only the kinematic model could be acceptable to simulate the robot current position, as in Akka and Khaber (2018), the inclusion of dynamic model provides more realistic results, because it contains the effects of inertia, friction and the actuator’s dynamics. Therefore, by combining both models, as in Leena and Saju (2016) and Ali et al. (2016), the accuracy of the simulations can be improved.

* This work was supported by FAPESB through project RED0004/2014.
This paper is organized into six sections. Section 2 presents the mathematical modelling of the DDMR. Section 3 describes both FLCs, including their MF settings. The performed simulations are detailed in Section 4. In Section 5, the results are presented and discussed. Finally, Section 6 summarizes and concludes this article.

2. DIFFERENTIAL DRIVE MOBILE ROBOT

A DDMR illustration is shown in Figure 1, where global and robot frames are shown.

The DDMR mathematical model presented by Dhaouadi and Hatab (2013) combines kinematics and dynamics, including kinematic constraints, inertia and actuators dynamics. The kinematic model is given as follows (Dhaouadi and Hatab, 2013):

\[
\begin{bmatrix}
\dot{x}_a \\
y_a \\
\dot{\theta}
\end{bmatrix} = \frac{R}{2} \begin{bmatrix}
\cos \theta & \cos \theta & 0 \\
\sin \theta & \sin \theta & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
\omega_r \\
\omega_l \\
\omega_l
\end{bmatrix} = A_{\text{cin}} \eta, \tag{1}
\]

where \( R \) and \( L \) are robot dimensions according to Figure 1, and \( \omega_r = \dot{\phi}_r \) and \( \omega_l = \dot{\phi}_l \) are the right and left wheel angular velocities, respectively. The dynamic model can be obtained by using Lagrange formalism and it is defined as follows (Dhaouadi and Hatab, 2013):

\[
M\ddot{\eta} + V(\dot{\eta}) = \tau, \tag{2}
\]

where \( q = [x_a, y_a, \theta, \phi_r, \phi_l]^T \) with \( x_a \) and \( y_a \) being the position coordinates of the center-point \( A \) in the global frame, \( \theta \) the robot orientation, \( \phi_r \) and \( \phi_l \) are the right and left wheel angular positions, \( \tau = [\tau_r, \tau_l]^T \) is the input torque,

\[
M = \begin{bmatrix}
I_w + \frac{R^2}{4L^2} (mL^2 + I_1) & \frac{R^2}{4L^2} (mL^2 - I_1) \\
\frac{R^2}{4L^2} (mL^2 - I_1) & I_w + \frac{R^2}{4L^2} (mL^2 + I_1)
\end{bmatrix},
\]

\[
V(\dot{q}) = \begin{bmatrix}
0 & \frac{R^2}{4L^2} m_c d \dot{\theta} \\
0 & -\frac{R^2}{4L^2} m_c d \dot{\theta}
\end{bmatrix},
\]

\( m = m_c + 2m_w \) is the total mass of the robot, \( m_c \) is the core mass, \( m_w \) is the mass of each wheel, \( I_1 = I_c + m_c d^2 + 2m_w L^2 + 2I_m \) is the total equivalent inertia, \( I_c \) is the core inertia, and \( I_m \) is the wheel’s inertia.

As in Ushikoshi et al. (2018), the actuators are DC permanent-magnet motors. Thus, the right and left actuator models can be defined according to Nise (2007), and including the viscous friction coefficient \( b \) (Okuyama et al., 2017) and a continuous and differentiable model of friction \( \tau_F \) (Makkar et al., 2005), which results in:

\[
\tau_m = A_{\tau m} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} + B_{\tau m} \begin{bmatrix} e_r \\ e_l \end{bmatrix} - \tau_F, \tag{4}
\]

where

\[
A_{\tau m} = \frac{1}{N} \begin{bmatrix} -K_c \omega_r \omega_l + b & 0 \\ 0 & -K_c \omega_r \omega_l + b \end{bmatrix}, \quad B_{\tau m} = \begin{bmatrix} \frac{K_i}{R_a} & 0 \\ 0 & \frac{K_i}{R_a} \end{bmatrix},
\]

\[
\tau_F = F_{\phi r} \tan(\alpha_k \phi_d) + F_{\phi l} \tan(\alpha_k \phi_d), \quad F_{\phi r}, F_{\phi l} = mg,
\]

\( K_c \) is the back EMF constant, \( K_i \) is the torque constant, \( R_a \) is the armature resistance, \( N \) is the gear ratio from the rotor to the wheel, \( F_{\phi r} \) and \( F_{\phi l} \) are static friction coefficients for right and left wheels respectively, \( \alpha_k \) and \( \alpha_k \) are the difference between static and kinetic friction coefficients for right and left wheels, respectively. Then, the DDMR model can be represented in a concise way by combining (1), (2) and (4) into a single augmented state-space. The complete mathematical model of the DDMR, including the PI velocity controllers, is defined as follows:

\[
\dot{\chi} = A_{\chi} \eta + B_{\chi} \eta + F_{\chi}, \tag{6}
\]

where

\[
A_{\chi} = \begin{bmatrix} 0_{3 \times 3} & A_{\text{cin}} & 0_{1 \times 3} \\ 0_{2 \times 3} & A_{\tau d} - B_{\tau d K_\phi} & B_{\tau d K_\phi} \\ 0_{2 \times 3} & -I_2 & 0_{2 \times 2} \end{bmatrix}, \quad B_{\chi} = \begin{bmatrix} 0_{2 \times 3} \\ B_{\tau d K_\phi} \end{bmatrix}, \quad F_{\chi} = [0_{1 \times 3} F_{\tau d} 0_{1 \times 2}]^T, \quad \eta = [\omega_r, \omega_l, \phi_d]^T, \quad K_\phi \quad \text{and} \quad K_\phi \quad \text{are the proportional and integral gain matrices of PI controllers,} \quad \eta \quad \text{is the input vector composed by both right \( \omega_r \) and left \( \omega_l \) angular wheel velocity references, and} \quad I_2 \quad \text{is the identity matrix of order 2.}
\]

3. FUZZY CONTROLLERS SETTINGS

This section presents the FB-Controller and the FBm-Controller, describing their universes of discourse, MFs and the considered defuzzification method. Some design details are omitted for brevity, but they can be found in Ushikoshi et al. (2018).

3.1 FB-Controller

The FB-Controller was proposed to allow bidirectional maneuvering for DDMRs. It receives geometric distance \( l \) and angle \( \Phi \) between the robot current position and the setpoints, and it computes the required linear \( v \) and angular \( \omega \) velocities for the robot tracking the given trajectory. The universes of discourse are defined as: \( 0 \leq l \leq 5 \text{ m} \) and \( -3.141 \leq \Phi \leq 3.141 \text{ rad} \), \( -0.1 \leq v \leq 0.1 \text{ m/s} \) and \( -0.5 \leq \omega \leq 0.5 \text{ rad/s} \). The MFs and inference rules are shown in Figure 2 and Table 1, respectively. For defuzzification, the Mean of Maxima method is considered.

\[ DOI: 10.17648/sbai-2019-111607 \]
(b) Input \( \Phi \).

(c) Output \( \omega_F \).

(d) Output \( v_F \).

Figure 2. FB-Controller membership functions parameters.

<table>
<thead>
<tr>
<th>Outputs</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_F )</td>
<td>( \Phi )</td>
</tr>
<tr>
<td>( \omega_F )</td>
<td>( \Phi )</td>
</tr>
<tr>
<td>R</td>
<td>N</td>
</tr>
<tr>
<td>P</td>
<td>CCW</td>
</tr>
<tr>
<td>P</td>
<td>CW</td>
</tr>
</tbody>
</table>

3.2 FBm-Controller

Aiming to provide smoother robot motion than FB-Controller, the MFs of \( \Phi \) in Figure 2 are changed to gaussian functions and the MFs of \( v \) are changed to sigmoidal (sig) and pi-shaped ones, as illustrated in Figure 3. To obtain a smoother controller response, intermediate levels of quantification are added to \( \Phi \), represented by the MFs PNR, PNF, PPF and PPR. The universe of discourse of \( v \) was adjusted to \(-0.15 \leq v \leq 0.15 \) m/s and Centroid method was used for defuzzification. The MFs of \( \Phi \) and \( v \) are illustrated in Figure 3. The inference rules were not modified, remaining the same as FB-Controller shown in Table 1. Only \( \Phi \) and \( v \) were adjusted because \( l \) is used only as stop criteria and \( \omega \) does not concern to the softness of the translational motion.

3.2 FBm-Controller

Aiming to provide smoother robot motion than FB-Controller, the MFs of \( \Phi \) in Figure 2 are changed to gaussian functions and the MFs of \( v \) are changed to sigmoidal (sig) and pi-shaped ones, as illustrated in Figure 3. To obtain a smoother controller response, intermediate levels of quantification are added to \( \Phi \), represented by the MFs PNR, PNF, PPF and PPR. The universe of discourse of \( v \) was adjusted to \(-0.15 \leq v \leq 0.15 \) m/s and Centroid method was used for defuzzification. The MFs of \( \Phi \) and \( v \) are illustrated in Figure 3. The inference rules were not modified, remaining the same as FB-Controller shown in Table 1. Only \( \Phi \) and \( v \) were adjusted because \( l \) is used only as stop criteria and \( \omega \) does not concern to the softness of the translational motion.

4. METHODOLOGY

This section presents the adopted methodology to test and evaluate the FB-Controller and FBm-Controller.

4.1 Simulation

Matlab/Simulink was used to simulate the system according to the simplified diagram shown in Figure 4. First, a previously defined trajectory is given as a position reference \((x(t), y(t))\). Then, the position error is computed and converted into: angular error \( \Phi \), by applying trigonometry to the robot direction \( \theta \) and the setpoint direction; and distance \( l \), by computing the Euclidean distance between the current position and the next setpoint position. These are the FLC inputs. Then, the FLC receives them and returns the required angular \( \omega \) and linear \( v \) velocities to achieve the current setpoint. These outputs are converted into right \( \omega_r \) and left \( \omega_l \) wheel angular velocity setpoints (Dhaouadi and Hatab, 2013), which are used as references for the PI controllers.

Notice that open loop velocity controllers would neglect small disturbances such as inclined floor and misalignment of the wheels. So, these PI controllers are indicated in an inner-loop. Their outputs are left and right armature voltage of the motor. A \( \pm 12 \) volts saturation is assumed to each voltage supply. Once applied to the motors, these voltages lead the wheel to develop torques which moves the robot considering friction.

Model Parameters

Some simulation parameters were defined based on a commercial DDMR, whereas others were arbitrarily chosen due to the lack of specification. The DDMR wheels and platform were assumed as solid cylinders whose central axis moment of inertia is calculated as \( I = \frac{1}{2} m_1 R_1^2 \), where \( m_1 \) is the cylinder’s mass and \( R_1 \) is its radius.

Considering \( d = 0 \), platform mass \( m_c = 3.44 kg \), radius \( L = 1.75 \cdot 10^{-1} m \), wheel mass \( m_w = 1.72 \cdot 10^{-1} kg \), and radius \( R = 3.4 \cdot 10^{-2} m \), the dynamic matrices \( M(q) \) and \( V(q, \dot{q}) \) (2) become

\[
M = \begin{bmatrix}
2.40 \cdot 10^{-3} & -1.09 \cdot 10^{-4} \\
-1.09 \cdot 10^{-4} & 2.40 \cdot 10^{-3}
\end{bmatrix}
\quad V(q) = 0.
\]

Moreover, the friction coefficients were defined as \( F_{s,R} = F_{s,L} = 0.025 \), \( F_{k,R} = F_{k,L} = 0.02 \), and \( \alpha_s = \alpha_k = 1 \). Actuators parameters were taken from a commercial 12 V DC motor and the PI gains were adjusted to be \( K_p = 3.5 \cdot I_2 \) and \( K_i = 15 \cdot I_2 \). Finally, for simplicity, the gear ratio was assumed to be \( N = 1 \) for both wheels.
4.2 Case Scenarios

The three types of disturbance applied to the DDMR are load mass variation, ground surface change, and low battery level condition. These disturbances were chosen attempting to address situations possibly experimented by DDMRs. First, load mass variation is fundamental for transportation applications. Second, there are several applications that require the robot to travel over multiple types of ground surfaces, and, sometimes, the surfaces are different for each wheel. Furthermore, the crucial need of a DDMR to move through the operating environment requires a portable power supply, such as a battery. However, the battery eventually discharges and it causes a voltage drop that impacts directly in the robot operation (Shepherd, 1965).

Load Mass Variation  The total mass of the robot, \( m \), is modified from the original (3.79 kg) to 11.37 kg at \( t = 10 \) s.

Ground Surface Change  A transition between different types of surface is simulated. It is done by increasing in 10 times both \( F_{s,r} \) and \( F_{k,r} \) of the right wheel, at \( t = 10 \) s.

Low Battery Condition  The system performance is analyzed under low battery level conditions. At \( t = 10 \) s, the system battery voltage is changed by reducing the applied armature voltage to 50%.

4.3 Controller Evaluation

The trajectory reference used is a crossed zig-zag path because it allows analyzing the robot performance in straight lines, and in smooth and sharp curves. This reference path is composed by setpoints, which include position \((x, y)\) and time \((t)\). In other words, the setpoint defines where the robot is expected to be and in each time. An RMSE, \( e \), is calculated based on the position setpoint and the robot current position at the respective time setpoint. However, defining setpoints does not guarantee the robot to follow a specific trajectory. So, an auxiliary trajectory was created to define the outline of the reference path, and it is composed of 4 sections (2 straight lines and 2 exponential curves). Then, an error \( e_{\text{traj}} \) (RMS value) was calculated between the performed and the auxiliary trajectory.

5. RESULTS AND DISCUSSION

The comparison between FB-Controller and FBm-Controller is presented in this section. Their performances are evaluated in the scenarios described in Section 4. The robustness of the controllers are evaluated via \( e \) and \( e_{\text{traj}} \) errors and visual remarks are made in both trajectories performed and FLCs outputs.

5.1 Load Mass Variation

Figure 5 shows both controlled trajectories. The non-smooth controller led the robot through a trajectory with very high accuracy on the first straight line. However, the robot passes the setpoint, and then, starts to deviate in the first sharp curve. Meanwhile, the smooth controller led it to make a more precise trajectory, only deviating when close to the sharp curve edges, and regaining precision right after them. Here, the smooth controller softness is evidenced.

Figure 5. Load mass variation: trajectory performed by the DDMR using FB and FBm controllers.

Table 2 shows the \( e \) and \( e_{\text{traj}} \) errors. Once the FBm-Controller has a larger \( v \) universe of discourse, it can achieve the setpoints better, what reduced \( e \). Besides, it led the robot to stay at reference neighborhood (as expressed by \( e_{\text{traj}} \)) unlike the FB-Controller, which keeps an offset error.

Table 2. Load mass variation: setpoint and trajectory errors using FB and FBm controllers.

<table>
<thead>
<tr>
<th></th>
<th>( e )</th>
<th>( e_{\text{traj}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB</td>
<td>0.2165</td>
<td>1.0616</td>
</tr>
<tr>
<td>FBm</td>
<td>0.0763</td>
<td>0.5715</td>
</tr>
</tbody>
</table>

Both FLCs have linear velocity \( v \) and angular velocity \( \omega \) as outputs, and their time-series are shown in Figures 6 and 7, respectively. Note that, only the FBm-Controller has an immediate response to the disturbance. Besides, this controller presents less high-frequency oscillations. Yet, the FB-Controller does not respond immediately, thus, it leads the robot to get delayed from the trajectory which increases the error.

Figure 6. Load mass variation: tested controllers output \( v \).

5.2 Ground Surface Change

The trajectories performed by the robot using the tested controllers are shown in Figure 8. The FBm-Controller leads the robot to perform the trajectory with a small error at the first sharp curve, but it fluctuates near the end. This oscillation continues, but gradually diminishes...
Figure 7. Load mass variation: tested controllers output $\omega$. Its amplitude, indicating stable damping. However, in this case, the FBm-Controller corrects the motion, as the change of friction makes it deviating from the path. Because of this, it is unable to complete the trajectory.

Figure 8. Ground surface change: trajectory performed by the DDMR using both controllers.

The performance indicators $e$ and $e_{\text{traj}}$ for both controllers are shown in Table 3. The $e_{\text{traj}}$ is more expressive in this case because the FB-Controller does not even finish the trajectory and the FBm fluctuates. Even though, the FBm-Controller led to $e_{\text{traj}}$ about 5 times smaller than the FB's.

Table 3. Ground surface change: setpoint and trajectory errors using the tested controllers.

<table>
<thead>
<tr>
<th></th>
<th>$e$</th>
<th>$e_{\text{traj}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB</td>
<td>0.4349</td>
<td>3.6579</td>
</tr>
<tr>
<td>FBm</td>
<td>0.1179</td>
<td>0.6984</td>
</tr>
</tbody>
</table>

The FLCs outputs and their time-series for this case are shown in Figures 9 and 10, respectively. Here, we see both FLCs reacting immediately after the applied disturbance. However, FBm-Controller reacts with less switching and faster, what leads to a predominant better performance. Yet, its angular velocity is less representative overall and presents a low-frequency oscillation. This means that the FBm-Controller makes minor corrections (as a result of being faster) to stay closer to the trajectory reference. It also explains the fluctuation seen in Figure 8.

Figure 9. Ground surface change: FB and FBm output $v$.

Figure 10. Ground surface change: FB and FBm output $\omega$.

5.3 Low Battery Conditions

The resulting trajectories performed by the robot using both FLCs are shown in Figure 11. Here, the disturbance effects do not affect the trajectory until the first sharp curve edge, where both controllers fail to track the path. However, the FBm-Controller corrects itself early and manages to complete the trajectory presenting a damped-oscillation nature. The FB-Controller also fails to complete the trajectory.

The errors $e$ and $e_{\text{traj}}$ for both controllers are shown in Table 4. One of the reasons of this scenario difficulties rely on the controllers arriving late at the setpoints, evidenced by a high $e$. Similar to what happens in ground surface change scenario. Again, high $e_{\text{traj}}$ is a result of fluctuations (FBm-Controller) and not completing the trajectory (FB-Controller).

Figure 11. Low battery conditions: trajectory performed by the DDMR using FB and FBm controllers.

Table 4. Low battery conditions: setpoint and trajectory errors using FB and FBm controllers.

<table>
<thead>
<tr>
<th></th>
<th>$e$</th>
<th>$e_{\text{traj}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB</td>
<td>0.3894</td>
<td>2.8193</td>
</tr>
<tr>
<td>FBm</td>
<td>0.1393</td>
<td>1.0909</td>
</tr>
</tbody>
</table>
The time-series of the FLCs outputs \( v \) and \( \omega \) are presented in Figures 9 and 10, respectively. Once more, the FBm-Controller reacts earlier than the FB-Controller, emphasizing its faster profile. The FBm-Controller softness is again evidenced, as well as its angular velocity free from switching.

![Figure 12. Low battery conditions: FB and FBm output \( v \).](image)

![Figure 13. Low battery conditions: FB and FBm output \( \omega \).](image)

6. CONCLUSIONS

In this paper, a comparison between two FLCs, the FB-Controller (Ushikoshi et al., 2018) and FBm-Controller was made. The case studies analyzed the FLCs behavior over disturbed scenarios often faced by DDMRs. The FBm-Controller performed better both in load mass and ground surface variation and also in low-battery conditions cases. This controller provided smoother responses with almost none low-frequency oscillations, unlike the FB-Controller.

Due to its higher error sensitivity, the FBm-Controller caused the robot to oscillate. However, it provided faster responses to disturbances, regaining the trajectory tracking. The ability to attain to the given path is promoted by the larger linear velocity universe of discourse.

Future works would lead to better results if the FBm-Controller fluctuation was improved. Also, the FLC adjustment could be enhanced by optimization techniques and the reliability of the simulations could be increased by refining the system identification.

REFERENCES
