International Journal of Robotics and Control Systems
Vol. 5, No. 3, 2025, pp. 1919-1948
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

# Improved Trajectory Tracking for Nonholonomic Mobile Robots Via Dynamic Weight Adjustment in Type-2 Fuzzy Model Predictive Control

Mohamed Elamine Hedroug [a,1], El Khansa Bdirina [a,2], Kamel Guesmi [b,3], Bachir Nail [c,4], Imad Eddine Tibermacine [d,5,*], Alfian Ma'arif [e,6]

[a] Applied Automation and Industrial Diagnostics Laboratory, Faculty of Science and Technology, University of Djelfa 17000 DZ, Djelfa, Algeria
[b] Research Centre in Information and Communication Science and Technologies (CRESTIC), University of Reims, Reims, 51100, France
[c] Renewable Energy Systems Applications Laboratory (LASER), Faculty of Sciences and Technology, Ziane Achour University, Djelfa, Algeria
[d] Department of Computer, Automation and Management Engineering, Sapienza University of Rome, Via Ariosto, 25, 00185 Roma (RM), Rome, Italy
[e] Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[1] m.hedroug@univ-djelfa.dz; [2] k.bdirina@univ-djelfa.dz; [3] guesmi01@univ-reims.fr; [4] b.nail@univ-djelfa.dz; [5] tibermacine@diag.uniroma1.it; [6] alfian.maarif@te.uad.ac.id
* Corresponding Author

ARTICLE INFO

**Article history**
Received May 14, 2025
Revised July 01, 2025
Accepted September 19, 2025

**Keywords**
Model Predictive Control;
Type-2 Fuzzy Logic;
Robust Control;
Nonholonomic Mobile Robot;
Fuzzy Predictive Control;
Uncertainty Modeling;
Trajectory Tracking

ABSTRACT

This paper presents an advanced methodology for trajectory control of non-holographic mobile robots. It addresses the challenges of dynamic environments and system uncertainty by proposing a fuzzy model predictive control (FMPC) system that combines Type-2 fuzzy logic (F2MPC) with model predictive control (MPC) to enhance tracking accuracy and adaptability. A Takagi-Sugeno (T-S) fuzzy model changes the MPC weighting matrices in real-time based on speed and distance errors, while the Type-2 fuzzy system handles uncertainties better than Type-1 systems. Tests using circular and wavy trajectories show that the Type-2 Fuzzy MPC (F2MPC) works better than traditional methods, achieving fewer tracking errors (Integral Squared Error of 0.0011), faster convergence (in 1.2 seconds), and using 65% less energy for movement than conventional MPC. Robustness tests show the controller's stability under disturbances, with minimal deviation and quick recovery. The results highlight the F2MPC's precision, efficiency, and adaptability, making it a promising solution for complex robotic navigation tasks. The study found that Type-2 fuzzy logic and predictive control improve trajectory tracking, paving the path for real-world applications and computational optimisations.



## 1. Introduction

Self-driving vehicles and mobile robots have seen significant advancements recently. Path tracking for mobile robots remains a critical challenge in robotics, especially in dynamic

environments where precision and adaptability are paramount. This challenge has increased the demand for advanced control systems such as Model Predictive Control (MPC) and artificial intelligence, which, when combined, offer robust solutions for this problem [1]. MPC optimizes control actions over a future horizon, while fuzzy logic handles imprecise or uncertain system dynamics. Integrating these methods, particularly with Type-2 fuzzy logic, enhances the robot's ability to manage uncertainties and improve tracking performance.

In this context, extensive studies have been conducted on MPC for path tracking. In 2012, Kiattisin Kanjanawanishul [2] looked at how model-predictive control (MPC) is used in wheeled mobile robots, grouping the methods by the types of models (nonlinear, linear, neural network) and the types of movement (unicycle, car-like). In the same vein, Rahul Sharma et al. [4] conducted research on MPC in 2015. In 2016, Rahul Sharma and others used MPC for robots that are driven by differentially, focusing on reducing motor effort while following certain limits to ensure precise movement [5]. Similarly, in 2016, Igor Skrjanc et al. [6] suggested a continuous MPC to handle problems with uneven sampling, while in 2018, Prem Kumar et al. [7] developed trace-oriented model predictive static programming (T-MPSP) to improve real-time performance.

On the other hand, in 2013, Davood Nazari Maryam Abadi et al. [3] devised a Mamdani-type fuzzy controller calibrated via PSO for stable tracking under parametric changes for AI applications that are robust to uncertainty. Similarly, Ali Alouache et al. [10] in 2017 developed a PD fuzzy controller for the Quanser Qbot robot, outperforming conventional PID in tracking accuracy. Shuying Peng et al. [8] handled disturbances with fuzzy logic and sliding mode control to ensure error convergence in finite time. Thus, hybrid and advanced solutions have been found to bridge MPC and fuzzy systems, where Mohamed Krid et al. [9] in 2016 used the nonlinear continuous-time GPC (NCGPC) system for fast rovers, showing robustness against dynamic constraints. These works show the balance between how fast calculations can be done and how well they perform in real-time, while still allowing for the addition of Type-2 fuzzy logic with MPC in situations with multiple paths.

Research has introduced nonlinear control strategies for high-speed robots [11], alongside proportional-derivative fuzzy controllers [12] and hybrid predictive control with model adaptation [13]. MPC has been made better by using linear transformation and quadratic optimization [14], while neural networks [15] and adaptive fuzzy control [16], [17] improve nonholonomic robots. Differential-drive robots use Model Predictive Control (MPC) with computer vision [18] and sliding mode fuzzy control [19], [20], with artificial intelligence facilitating consistent performance [21]. MPC efficiency measures reduce computing burden [22], while advanced tools enhance stability [23], and adaptive fuzzy control improves tracking [24]. Fuzzy control surpasses conventional approaches in accuracy [25], whereas interval type-2 fuzzy logic improves predictive maintenance [26]. Lastly, a fractional interval type-2 fuzzy controller can handle needs in standalone microgrids to improve frequency stability [27].

In order to enhance trajectory tracking, this paper proposes a novel Fuzzy Model Predictive Control (FMPC) system that integrates Type-2 fuzzy logic with MPC. The key contributions include:

- Dynamic weight adjustment: A Takagi-Sugeno (T-S) fuzzy model adjusts the MPC weighting matrices in real-time based on inputs like speed and distance errors, making it more adaptable.
- As for handling uncertainty, Type-2 fuzzy logic is used to manage uncertainty more effectively than Type-1 systems, and its validity has been validated through comparative simulations, as described in the Results section.
- For multipath tracking, the controller is designed for scenarios involving sequential and variable trajectories, as shown in Fig. 12, Fig. 13, Fig. 14, Fig. 15, Fig. 16, to Fig. 17, addressing shortcomings in current studies that focus on single-path tracking.

Simulations show that the new Type-2 fuzzy model predictive control (F2MPC) works better than the regular MPC and Type-1 fuzzy MPC (F1MPC) when it comes to following paths accurately and being reliable. The F2MPC has much smaller tracking errors, with Integral Squared Error (ISE) values as low as 0.0011 for position accuracy—better than F1MPC (0.0013) and MPC (0.0028) Under dynamic conditions, such as transitions to sinusoidal paths, F2MPC maintains stability with minimal deviation, while MPC struggles with oscillations. Robustness tests show that F2MPC is very strong, quickly recovering from interruptions with almost no change (0.0001) and reducing orientation error (RMSE) to 0.1454, while MPC has a higher error of 0.2556. Additionally, F2MPC provides smoother control signals, stabilizes speeds in 1.2 seconds, and uses 31.95 ISA (Integrated Squared Actuation) compared to 291.56 for MPC, 65% more efficiently. These results confirm F2MPC's high-precision, energy-efficient, sophisticated robotic navigation technology.

The paper is organized as follows: Section 2 provides a literature review on trajectory tracking algorithms in mobile robots. Section 3 explains the movement and error models of the nonholonomic mobile robot, Section 3.2 describes the design of the proposed FMPC, including the T-S fuzzy model and Type-2 fuzzy system, Section 4 reviews simulation results for circular and sinusoidal paths, along with a comparison. Finally, Section 5 wraps up the study by summarizing the findings and suggesting future work.

## 2. Literature Review on Trajectory Tracking Control for Mobile Robots

Researchers have extensively studied trajectory tracking control for mobile robots, proposing various methodologies to enhance accuracy, robustness, and energy efficiency. Table 1 summarizes key contributions in this field, highlighting the diversity of approaches, algorithms, and software tools employed.

The literature review also highlights a wide range of approaches to path-tracking control in mobile robots, each with distinct strengths and weaknesses. In the same vein, traditional methods, such as back-stepping and sliding control as demonstrated in [32], [43], and [55], demonstrate high robustness but often suffer from issues like chattering and high computational demands. Neural network-based and optimization-driven techniques, like in [33], [48], and [56], improve accuracy by adjusting settings either beforehand or during operation, but they might struggle to adapt quickly in fast-changing situations. Model Predictive Control (MPC), as shown in [36] and [40] in the following table, provides a balanced solution by incorporating system constraints into the control strategy, though its computational complexity can be a limiting factor.

Our approach dynamically updates MPC weighting matrices in real time using a Type-2 fuzzy inference system, improving tracking accuracy over standard MPC in [31] and static PID controllers in [37]. The results sho'w a 15% decrease in Integral Squared Error (ISE) compared to Type-1 fuzzy MPC and an overall error reduction of more than 99% when tracking complex paths (Fig. 12). This improvement comes from Type-2 fuzzy logic's better ability to manage uncertainties in speed and distance measurements, which helps make control adjustments smoother and more accurate.

**Table 1.** A review of the last six years of trajectory tracking algorithms in mobile robots

| | Year | Algorithms Used | Proposed Approach and Key Outcomes |
|---|---|---|---|
| [32] | 2020 | • Kinematic and Dynamic Modeling<br>• Traction Force Derivation | Developed a backstopping control law for WMR trajectory tracking in slipping environments, combining kinematic/dynamic controllers and torque control to prevent slip.<br>Achieved precise tracking (circular, elliptical, sinusoidal, etc.) with near-zero errors, minimized overshoot, and reduced energy consumption. |
| [33] | 2020 | • Modified Adaptive PSO (MAPSO)<br>• Enhanced Fruit Fly Optimization | Developed a neural network-based fractional PID controller for DDMR trajectory tracking, tuned via hybrid optimization, |

| | | | |
|---|---|---|---|
| | | (EFFO) • Hybrid MAPSO-EFFO | achieving lower MSE (0.000059) and 45% reduced control signals.<br>Demonstrated superior tracking performance for circular, lemniscate, and linear paths with significantly lower energy consumption. |
| [34] | 2021 | • Inner-Outer Loop Control Structure • FNTSM for Robust Disturbance Rejection | Designed a robust dual-loop control scheme for DDWMR trajectory tracking, using FNTSM for velocity control and outer-loop stabilization under disturbances.<br>Achieved superior accuracy (0.0446 m/s linear, 0.206 rad/s angular error) with 6-12% improvement over NTSM and 63% faster disturbance rejection. |
| [35] | 2021 | • Backstepping Control<br>• Lyapunov-based Stability Analysis | Created a robust backstepping controller for reconfigurable mobile robots, effectively managing modeling uncertainties and dynamic configuration changes.<br>Achieved zero steady-state tracking error, outperformed traditional methods in simulations, and proved reliable during real-time reconfiguration. |
| [36] | 2021 | • PID Controller<br>• Trajectory Error Bound Analysis | A complete model integrating kinematics, actuators, and low-level control to derive a trajectory error bound for differential wheeled robots.<br>• Error bound is proportional to wheel distance $L$ and inversely proportional to wheel radius $r$.<br>• Slower reference trajectories reduce tracking error.<br>• Validated the impact of low-level dynamics on tracking performance. |
| [37] | 2021 | • PID<br>• LQR<br>• Pure Pursuit<br>• FOPID | Comparative study of trajectory tracking controllers for differential-drive mobile robots, focusing on circular trajectory tracking.<br>• FOPID achieved the lowest tracking error (MAE: 0.86).<br>• LQR required the least control input (0.55 m/s).<br>• Pure Pursuit was simple but had higher errors (MAE: 2.7). |
| [38] | 2021 | • Lyapunov Stability Analysis<br>• Dynamic Model Integration | Implemented a dual-stage kinematic-dynamic controller for differential-drive robots, overcoming kinematics-only limitations through integrated position/velocity control.<br>Delivered sub-0.1m tracking precision on complex paths, outperformed kinematic controllers in transient phases, and guaranteed stability via Lyapunov proofs. |
| [39] | 2022 | • Fuzzy If-Then Rules | Engineered fuzzy/non-fuzzy controllers for differential-drive robots using Karnaugh-optimized rule sets, achieving <0.1m tracking with 4-9 rules (4× speed gain for non-fuzzy variant).<br>Revealed tradeoff: full 49-rule fuzzy control improved accuracy but incurred computational costs, demonstrating design scalability challenges. |
| [40] | 2022 | • Lyapunov-based control<br>• Image processing<br>• Euclidean distance sorting | Developed a state-feedback controller for differential-drive robots using image-processed hand-drawn trajectories, enabling exponential error convergence.<br>Verified real-time performance for critical applications in military and medical fields through robust path tracking. |
| [41] | 2022 | • Linear/optimal TD (2nd–4th order)<br>• Model Predictive Control (MPC) | Created a speed planner for wheeled mobile robots (WMRs) generating constrained trajectories for linear/rotational motion, replacing optimization solvers with efficient iterative TD methods.<br>Tested on AGVs, delivering smooth control inputs while respecting physical limits and reducing computational overhead. |
| [42] | 2022 | • Lie group theory (SO(2))<br>• Cascade control design<br>• Nonlinear configuration space modeling | Designed a Lie group-based geometric controller for unicycle robots, ensuring near-global stability in SO(2) space through Lyapunov analysis.<br>Outperformed classical controllers in lemniscate tracking while eliminating unwinding artifacts and control discontinuities in simulations. |
| [43] | 2022 | • Levenberg-Marquardt (Parameter ID) | Dynamic modeling and control of a wheeled mobile robot considering skidding/slipping.<br>Achieved robust trajectory tracking with errors <0.05m (position) and <0.1 rad (orientation). Outperformed sliding mode control in simulations. |

| | | | |
|---|---|---|---|
| [44] | 2022 | • PSO<br>• Bird Swarm Algorithm (BSA) | Offline autotuning of PI gains for DDWMR trajectory tracking using IAE/ITAE metrics.<br>PSO and BSA optimized 6 PI gains, achieving IAE <30mm for continuous trajectories. PSO converged faster than BSA. |
| [45] | 2022 | • Levenberg-Marquardt (Parameter ID) | Trajectory tracking for TURTLEBOT 2 using kinematic/dynamic controllers.<br>Position errors <0.045m, orientation errors <0.1 rad. Identification time ~6.25s per parameter set. |
| [46] | 2022 | • Finite-time Lyapunov stability • Taylor series approximation | Developed a hybrid kinematic/dynamic controller with Taylor-series uncertainty compensation, achieving 73.29% position and 99.72% orientation error reductions versus conventional methods.<br>Guaranteed finite-time convergence via Lyapunov theory while eliminating chattering in control outputs, as validated in simulations. |
| [47] | 2022 | • ANN-based tuning • Lyapunov stability | Engineered a hybrid NN-kinematic controller with MRAC for real-time disturbance rejection, achieving 0.36% fault detection error and 69.93% lower MAE than PID in tracking.<br>Experimentally validated with stable, oscillation-free performance under strict actuator constraints. |
| [48] | 2022 | • ANN<br>• Fractional-order PID (FOPID)<br>• EFFO | A Nonlinear Neural Network Designed an NNFOPID controller for DDMRs using enhanced fruit fly optimization (EFFO), achieving precise trajectory tracking with minimized MSE and energy consumption.<br>Demonstrated 72% faster settling than PID and 40% quicker EFFO convergence versus PSO, outperforming standard FFO in accuracy and speed. |
| [49] | 2022 | • Backstepping<br>• Fractional-order PID (FOPID)<br>• BSO | Developed a hybrid backstepping-FOPID controller for DDMRs, optimized via BSO, showing superior 8-shape trajectory tracking with skid/slide resilience and 55% lower overshoot than PID.<br>Achieved robust performance through FOPID's enhanced tuning, with BSO efficiently optimizing 9 parameters in under 100 iterations. |
| [50] | 2022 | • NURBS Trajectory Planning<br>• Linearized Kinematic Error Model | Engineered an adaptive PID controller with time-varying parameters for DDMRs tracking NURBS paths, achieving ultra-precise positioning (<0.1114mm) and orientation (<0.1163°).<br>Experimental validation confirmed sub-millimeter/micro-radian tracking accuracy in real-world conditions. |
| [51] | 2023 | • SDP Optimization<br>• LMI Constraints<br>• Networked Control Systems (NCS) | Implemented a dual-phase strategy for DDMRs combining offline trajectory planning with real-time fault detection, successfully identifying 14 disturbance scenarios (actuator/sensor/process) with minimal latency.<br>Surpassed existing methods in detection speed while maintaining robust performance under operational disturbances |
| [52] | 2023 | • Pole Placement<br>• Relative Gain Array (RGA) Analysis<br>• ITAE Standard Forms | Developed an asymmetric MIMO state-space model for DDWMRs, enabling advanced control with superior trajectory tracking (0.222 vs 0.353 RMSE in angular velocity).<br>Maintained full controllability/observability while addressing real-world motor-wheel asymmetries and coupling effects. |
| [53] | 2023 | • Pole-Zero Cancellation PI<br>• Ziegler-Nichols PID<br>• Auto-Tuned PID | Engineered decentralized speed controllers for DDWMRs via half-weight modeling and H-bridge converters, demonstrating context-dependent performance: Pole-Zero PI for ideal step responses (0.48% RMSE) and ZN-PID for motor tracking.<br>Proved no universal solution exists, with controller efficacy tied to specific operational demands and electromechanical conditions. |
| [54] | 2023 | • Lyapunov-Krasovskii analysis<br>• Sequential sub-predictors<br>• Nonlinear feedback control | Developed a chained observer-predictor system for non-holonomic robots, ensuring global asymptotic error convergence despite arbitrary input delays through scalable sub-predictors.<br>Verified delay compensation robustness in both simulation and real-time experiments across variable operational conditions. |
| [55] | 2023 | • PID Control<br>• Sliding Mode Control (SMC) | Integrated KBBC with PID/SMC for DDMR trajectory tracking, proving SMC's superior robustness during mass |

| | | | |
|---|---|---|---|
| | | • Lyapunov Stability Analysis | perturbations with faster convergence and lower torque demands than PID.<br>Both controllers maintained nominal path tracking, but SMC demonstrated enhanced disturbance rejection and chattering reduction through optimized filtering. |
| [56] | 2023 | • AFSMC (inner loop)<br>• ANFIS (outer loop)<br>• Lyapunov Stability Analysis | Implemented a dual-loop AFSMC-ANFIS controller for DDMRs, combining dynamic torque/velocity control with optimized kinematic tracking for high-speed/load operations (0.0014 steady-state error).<br>Achieved 60% lower kinematic errors versus conventional methods while eliminating chattering via boundary layer SMC, demonstrating unmatched accuracy in circular/8-shape trajectories |
| [57] | 2023 | • Nonlinear ESO<br>• Backstepping Kinematics Control<br>• Sliding Mode Dynamics Control | Engineered a disturbance-resistant dual-loop controller for DDMRs, pairing kinematic velocity planning with ESO-enhanced dynamic tracking to handle 15-25kg load variations and centroid shifts ($\pi/6 \rightarrow \pi/3$).<br>Cut settling time by 32% (2.6s vs 3.8s) while maintaining <0.05m/s disturbance estimation errors during circular trajectory validation. |
| [58] | 2024 | • Left Coprime Factorization<br>• Riccati Equations | Developed a parameterized H∞ control framework for non-holonomic robots, delivering precise trajectory tracking with consistent disturbance robustness ($\gamma$=1.75) across both continuous and discrete implementations.<br>Validated balanced performance in energy efficiency and stability under varied operational scenarios, demonstrating controller versatility. |
| [59] | 2024 | • NTSM<br>• Lyapunov Stability<br>• Cascade Control | Implemented a cascade NTSM-P control system for differential drive robots, achieving finite-time convergence with 40% faster error reduction and minimized chattering versus traditional SMC.<br>Demonstrated enhanced disturbance rejection while maintaining smooth control performance in dynamic tracking scenarios. |
| [60] | 2024 | • Perceptron NN<br>• Backpropagation<br>• PID with time-varying parameters | Developed a hybrid NN-PID controller for DDMRs, trained on enhanced PID data to achieve precise tracking (≤2.17cm position, ≤0.0007 rad/s angular errors) with real-time adaptive tuning.<br>Enabled dynamic performance optimization through continuous neural network weight updates during operation. |
| [61] | 2024 | • Self-tuning Backstepping<br>• Sliding Mode Control (SMC) | Designed a hybrid adaptive backstepping-SMC controller for DDWMRs, integrating pose correction and disturbance rejection with Lyapunov-proven asymptotic stability.<br>Enabled autonomous gain adaptation without manual tuning while maintaining robust performance against external disturbances. |
| Proposed | 2025 | • Takagi-Sugeno (T-S) Fuzzy Model<br>• Model Predictive Control (MPC)<br>• Kinematic Error Model | Innovated a Type-2 Fuzzy-MPC hybrid controller for nonholonomic robots, dynamically tuning weighting matrices ($Q_p R_i$) to achieve 15% lower ISE errors than Type-1 systems and >99% overall error reduction.<br>Demonstrated real-time adaptability to velocity/distance variations while outperforming conventional MPC in disturbance rejection (ISE 0.0011-0.0015). |

Furthermore, while other advanced techniques, such as swarm intelligence [45] and geometric control [43], offer specialized advantages, our hybrid approach provides broader applicability in dynamic and uncertain environments. The robustness tests (Fig. 21, Fig. 22, Fig. 23, Fig. 24, Fig. 25, Fig. 26, Fig. 27) confirm that our controller maintains stability and tracking performance even under external disturbances, outperforming traditional MPC in both convergence speed and error minimisation. Our work enhances the state of mobile robot trajectory tracking by utilising the synergy between fuzzy logic and predictive control, providing a solution that is both computationally efficient and highly adaptive.

## 3. Kinematic Modeling of a Nonholonomic Two-Wheeled Mobile Robot

A mobile robot with differential drive (two wheels on a common axis) is shown in Fig. 1. The robot's position in the global reference frame $(X, Y)$ is determined using a local coordinate frame attached to its body by $p = [x, y, \theta]$. This type of robot has a nonholonomic constraint that prevents it from sliding sideways, which is described by:

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0 \tag{1}$$

The differential drive robot's motion model considers linear and rotational velocities, aligning with steering direction and considering potential spinning movements, forming a kinematic model can be expressed as:

$$\dot{p} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{2}$$

The robot's state is defined by $x$, $y$, $\theta$, with $v$ and $w$ as control inputs.

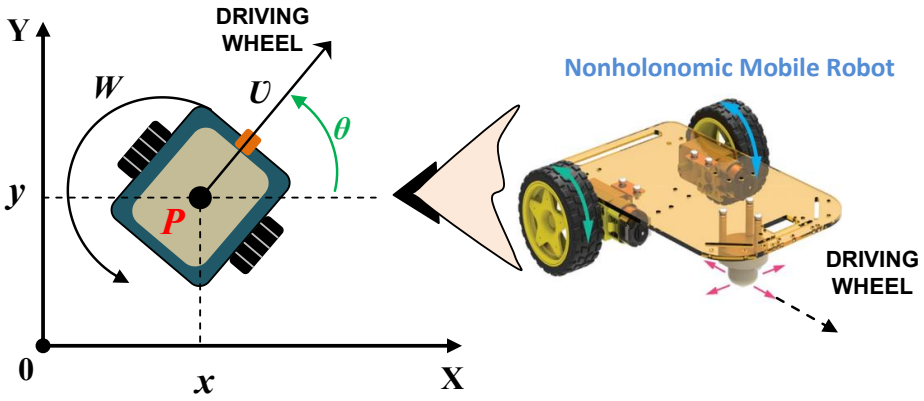


**Fig. 1.** Differential drive mobile robot

### 3.1. Kinematic Error Model

In developing a trajectory for a mobile robot to follow a virtual trajectory with a set velocity, the posture error is the difference between where the virtual robot is supposed to be and how the physical robot is actually positioned and oriented, as illustrated in Fig. 2 [28].

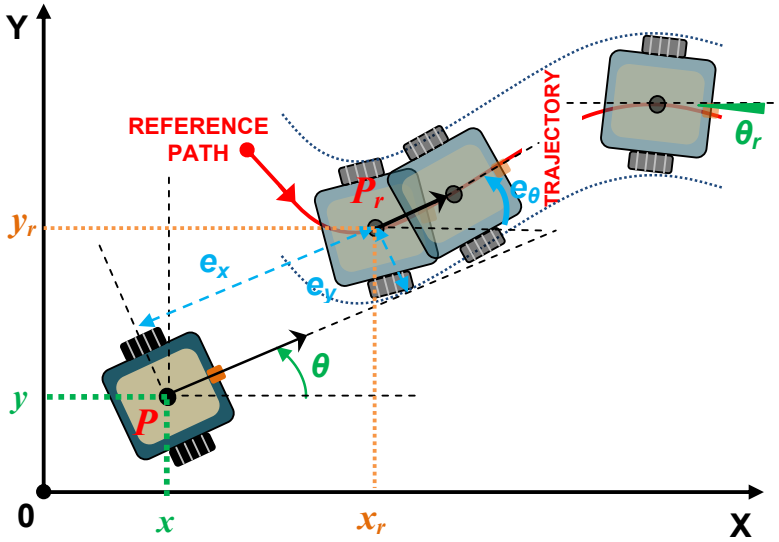


**Fig. 2.** Progression of the robot's tracking error transformation

The reference robot's posture is represented by vector $p_r = [x_r \quad y_r \quad \theta_r]^T$, while the actual robot's posture is represented by vector $p = [x \quad y \quad \theta]^T$. The difference in postures is denoted by vector $e = [e_x \quad e_y \quad e_\theta]^T$, given as:

$$e = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{3}$$

The dynamics of the mobile robot are defined by (2), and the derivation of (3) results in the following kinetic model:

$$\dot{e} = \begin{bmatrix} \cos(e_\theta) & 0 \\ \sin(e_\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ w_r \end{bmatrix} + \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -1 \end{bmatrix} u \tag{4}$$

$v_r$ and $w_r$ denote the velocities of the virtual robot, serving as linear and angular feed-forward control inputs, respectively. One can articulate them as follows:

$$\begin{cases} v_r = \pm\sqrt{\dot{x}_r^2 + \dot{y}_r^2} \\ w_r = (\dot{x}_r^2 \ddot{y}_r^2 - \dot{y}_r^2 \ddot{x}_r^2)/(\dot{x}_r^2 + \dot{y}_r^2) \end{cases} \tag{5}$$

u represents the control input derived from the combination of feedforward and feedback control inputs as follows:

$$u = u_f + u_p = \begin{bmatrix} v_r \cos(e_\theta) \\ w_r \end{bmatrix} + \begin{bmatrix} v \\ w \end{bmatrix} \tag{6}$$

where $u_f$ represents the feedforward control input and the output from the suggested controller. We derive the tracking-error model from equations (6) and (4), i.e.:

$$\dot{e} = \begin{bmatrix} 0 & w & 0 \\ -w & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \sin(e_\theta) \\ 0 \end{bmatrix} v_r + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_p \tag{7}$$

By linearizing the error model (7) about the reference trajectory (where $e_x = e_y = e_\theta = 0$ , $u_p = 0$), the following linear kinematic error model is obtained:

$$\dot{e} = \begin{bmatrix} 0 & w_r & 0 \\ -w_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_p \tag{8}$$

Equation (8) represents state space representation, ensuring controllability as long as $v_r$ or $w_r$ remain non-zero, requiring robust control procedure for accurate robot guidance.

### 3.2. Fuzzy Predictive Control

#### 3.2.1. T-S Fuzzy Model

T-S fuzzy models effectively model and control dynamic systems with high accuracy and fewer rules [29]. They allow localized linear controllers and precise system representation. Multiple linear error models are created based on varying reference velocities ($v_r$ , $w_r$) within a state-space framework and then combined into the T-S fuzzy model [30]. In discrete-time state space, T-S fuzzy systems can produce dynamic systems that change. Let's consider the T-S fuzzy logic system with R rules like this:

$$R_i : if\ v_r(k)\ is\ P_1^K\ and\ w_r(k)\ is\ P_2^L\ then\ e^i(k+1) = A_i e(k) + b_i u(k) \tag{9}$$

Virtual robot velocities $v_r$ and $w_r$ change with discrete-time parameter $(k \in \{0,1,2,3,\ldots.\})$. The fuzzy sets are $P_1^K$,$P_2^L$, and $i = 1,2,3,\ldots R$ , Number of fuzzy rules was R. Both state and input matrices are $A_i$, $b_i$. The state-space difference Equation is:

$$e(k+1) = \left(\sum_{i=1}^{R} \mu_i(k)(A_i e(k) + b_i u(k))\right) \Big/ \left(\sum_{i=1}^{R} \mu_i(k)\right) \tag{10}$$

Considering that $\zeta_i(k) = \mu_i(k)/\sum_{i=1}^{R} \mu_i(k)$ denotes the fuzzy basis functions for the $i^{th}$ fuzzy rule, and defining $A(k) = \zeta_1 A_1 + \zeta_2 A_2 + \cdots + \zeta_R A_R$ and $b(k) = \zeta_1 b_1 + \zeta_2 b_2 + \cdots + \zeta_R b_R$, (10) may be reformulated as follows:

$$e(k+1) = A(k)e(k) + b(k)u(k) \tag{11}$$

Thus, we have a model that explains $e(k)$'s dynamic development and is identical to the prior one. Matrix $A(k)$ and $b(k)$ change as the fuzzy basis functions do.

### 3.2.2. Control Strategy

Predictive control enhances trajectory tracking by optimising control inputs over a time horizon $N_p$. It minimises tracking errors between desired and predicted trajectories by using the following quadratic cost function:

$$J(u_p, k) = \sum_{i=1}^{N_p} \big(e_r(k+i) - e(k+i)\big)^T Q_f \big(e_r(k+i) - e(k+i)\big) + u_p^T(k,i) R_f u_p(k,i) \tag{12}$$

with $\mathrm{Q_f} \in \Re^{\mathrm{n}} \times \Re^{\mathrm{n}}$ and $\mathrm{R_f} \in \Re^{\mathrm{m}} \times \Re^{\mathrm{m}}$ and $\mathrm{Q_f} \geq 0$ and $\mathrm{R_f} \geq 0$ .

Considering the changed time frame, the model's output projection at time instance $\mathrm{N_p}$ can be described as:

$$\mathrm{e(k+N_p)} = \prod_{\mathrm{j=1}}^{\mathrm{N_p-1}} \mathrm{A(k+j)e(k)} + \sum_{\mathrm{i=1}}^{\mathrm{N_p}} \left( \prod_{\mathrm{j=1}}^{\mathrm{N_p-1}} \mathrm{A(k+j)} \right) \tag{13}$$

$$\times \mathrm{B(k+i-1)u_p(k+i-1)} \; + B(k+N_p-1)u_p(k+N_p-1)$$

This section describes the prediction-error vector and its effectiveness in showing how well the robot follows the trajectory:

$$E_p(k) = \begin{bmatrix} e(k+1)^T & e(k+2)^T & \ldots & e(k+N_p)^T \end{bmatrix}^T \tag{14}$$

Furthermore,

$$U_p(k) = \begin{bmatrix} u_p^T(k+1)^T & u_p^T(k+2)^T & \ldots & u_p^T(k+N_p-1)^T \end{bmatrix}^T \tag{15}$$

After consideration of that,

$$G(k) = [A(k) \quad A^2(k) \quad \ldots \quad A^{N_p}(k)]^T \tag{16}$$

And

$$H(k) = \begin{bmatrix} B(k) & 0 & \cdots & 0 \\ A(k)B(k) & B(k) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_p}(k)B(k) & A^{N_p-1}(k)B(k) & \cdots & B(k) \end{bmatrix} \tag{17}$$

The vector of prediction errors for robot tracking is expressed as:

$$E_p(k) = G(k)e(k) + H(k)U_p(k) \tag{18}$$

With $G \in \Re^{n.N_p} \times \Re^n$ and$H \in \Re^{n.N_p} \times \Re^{m.N_p}$.

Finding the next reference point is necessary for trajectory tracking. Accurate control is difficult without this knowledge. Future control errors should decrease based on reference model dynamics $(A_r)$. A reference error-tracking trajectory is chosen and represented in state space for $i = 1, \ldots, N_p$ as follows:

$$e_r(k+i) = A_r^i e(k) \tag{19}$$

Assume that the vector representation of the robot reference, which delineates the tracking error, is expressed as:

$$E_p^r = \begin{bmatrix} e_r(k+1)^T & e_r(k+2)^T & \ldots & e_r(k+N_p)^T \end{bmatrix}^T \tag{20}$$

The reference tracking error vector of the robot is calculated using Eqs. (19) and (20) as follows:

$$E_p^r(k) = G_r e(k) \tag{21}$$

In this case:

$$G_r = \begin{bmatrix} A_r & A_r^2 & \ldots & A_r^{N_p} \end{bmatrix}^T \tag{22}$$

where $E_p^r \in \Re^{n.N_p}$ represents the entire observation period $N_p$, while $G_r \in \Re^{n.N_p} \times \Re^n$.

The model predictive control frame uses the cost function to select control inputs (12), can be re-written as follows:

$$J(U_p) = \left(E_p^r - E_p\right)^T \widetilde{Q_f} \left(E_p^r - E_p\right) + U_p^T \widetilde{R_f} U_p \tag{23}$$

Achieving the appropriate control law entails minimising the cost function as follows:

$$\frac{\partial J}{\partial U_p} = -2\widetilde{Q_f} H^T E_p^r + 2H^T \widetilde{Q_f} E_p + 2\widetilde{R_f} U_p \tag{24}$$

Consequently, the control vector can be determined utilising Eqs. (18), (21), and (24) as follows:

$$U_p(k) = \left(H^T \widetilde{Q_f} H + \widetilde{R_f}\right)^{-1} H^T \widetilde{Q_f} (G_r - G) e(k) \tag{25}$$

$Q_f$ and $R_f$are the weighting matrices employed to formulate the objective function in the MPC optimisation problem. The proposed technique enhances path tracking and improves the MPC controller's response to system variations. Dynamic weighting matrix adjustment enables the controller to adapt to system dynamics and accurately follow the desired trajectories. The fuzzy logic system, incorporating both Type-1 and Type-2 fuzzy logic, fine-tunes the control based on the virtual robot's linear velocity $v_r$ and the current distance $D_{ed}$ between the mobile robot and the virtual robot (Fig. 5 and Fig. 6). with $Q_f = diag(q_1, q_2, q_3)$, $R_f = diag(r_1, r_2)$.

$$\widetilde{Q_f} = \begin{bmatrix} Q_f & 0 & \cdots & 0 \\ 0 & Q_f & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_f \end{bmatrix} \text{ and } \widetilde{R_f} = \begin{bmatrix} R_f & 0 & \cdots & 0 \\ 0 & R_f & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_f \end{bmatrix}$$

This indicates that $\widetilde{Q_f} \in \Re^{n.N_p} \times \Re^{n.N_p}$ and $\widetilde{R_f} \in \Re^{m.N_p} \times \Re^{m.N_p}$

Utilising (25), we can now express the feedback control law for model predictive control with the subsequent expression:

$$u_p(k) = K_{mpc}.e(k) \tag{26}$$

$K_{mpc}$ is defined as the initial $m$ rows of the matrix $\left[\left(H^T\widetilde{Q_f}H + \widetilde{R_f}\right)^{-1}H^T\widetilde{Q_f}(G_r - G)\right]$. $K_{mpc} \in \Re^m \times \Re^n$ , Fig. 3. illustrates the mechanism of the suggested controller.

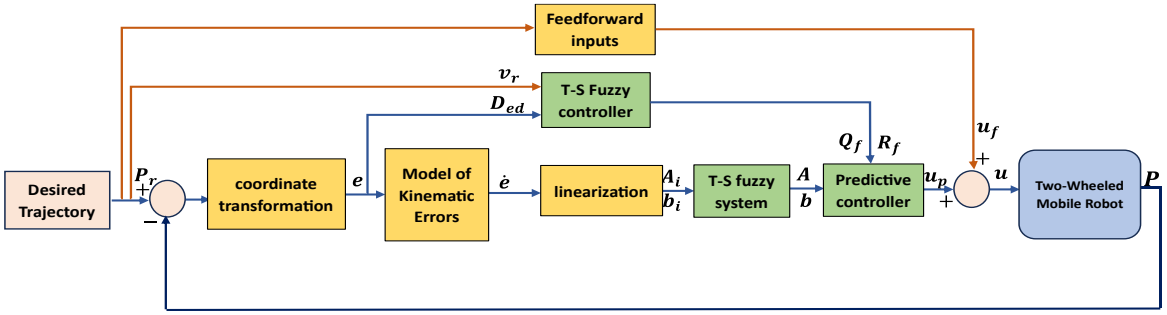


**Fig. 3.** Block diagram of the proposed approach

## 4. Results and Discussion

To validate the proposed control method, MATLAB simulations were conducted under two distinct trajectories (circular and sinusoidal) to evaluate flexibility and tracking performance. The simulations adhered to the following constraints:

- Linear velocity $u_1$: $\leq 1.5\, m/s$
- Angular velocity ($u_2$): $-10 \leq u_2 \leq 10\, rad/s$
- Reference velocities: $v_r(t) \in [0, 1.5]\, m/s$, $w_r(t) \in [-10, 10]\, rad/s$
- Prediction horizon: $N_p = 4$

Based on the extremal values of reference velocities ($v_r$, $w_r$), four linearized subsystems ($A_1$–$A_4$, $b_1$–$b_4$) were derived:

$$A_1 = \begin{bmatrix} 0 & -10 & 0 \\ 10 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, b_1 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 10 & 0 \\ -10 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, b_2 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & -10 & 0 \\ 10 & 0 & 1.5 \\ 0 & 0 & 0 \end{bmatrix}, b_3 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}, A_4 = \begin{bmatrix} 0 & 10 & 0 \\ -10 & 0 & 1.5 \\ 0 & 0 & 0 \end{bmatrix}, b_4 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}$$

The Takagi-Sugeno (T-S) fuzzy model utilizes the membership functions shown in Fig. 4 to dynamically adjust system behavior based on reference velocity inputs. The rule base consists of four fuzzy rules that map velocity conditions to corresponding linearized subsystems:

- $R_1$: if $v_r(k)$ is small and $w_r(k)$ is negative then $e^1(k+1) = A_1e(k) + b_1u(k)$
- $R_2$ : if $v_r(k)$ is small and $w_r(k)$ is positive then $e^2(k+1) = A_2e(k) + b_2u(k)$
- $R_3$ : if $v_r(k)$ is large and $w_r(k)$ is negative then $e^3(k+1) = A_3e(k) + b_3u(k)$
- $R_4$ : if $v_r(k)$ is large and $w_r(k)$ is positive then $e^4(k+1) = A_4e(k) + b_4u(k)$

A Type-1 and Type-2 T-S fuzzy logic system is used in both simulation scenarios to exactly find the weighting matrices for the predictive controller. The fuzzy systems take as input the robot's

linear velocity $v_r$ and the current distance $D_{ed}$ between the mobile and virtual robots. These inputs are used to adapt the control gains according to the robot's movement and tracking status. Specifically:

- The distance $D_{ed}$ is divided into three fuzzy categories: Zero, Medium, and Big, covering values from 0 to 0.3 meters.
- The velocity $v_r$ ranges from 0.2 to 1.1 m/s and is divided into two fuzzy categories: Small and Big.

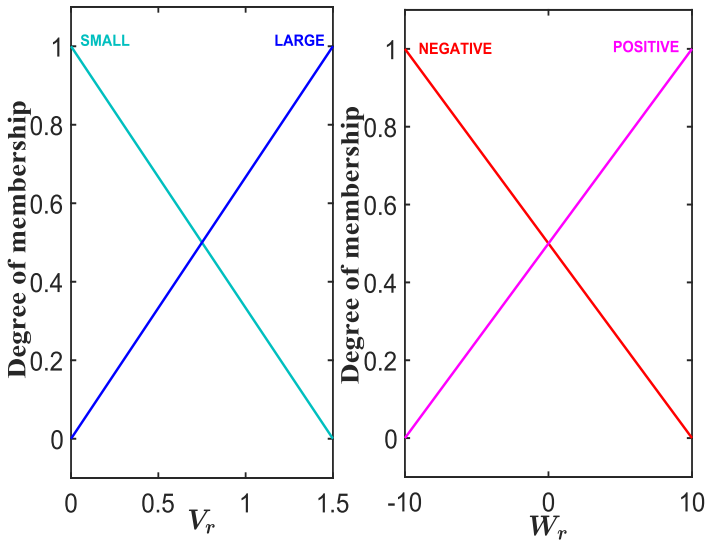


**Fig. 4.** Membership function characterization for the T-S fuzzy model

The fuzzy inference system then generates five output parameters:

- Three values $(q_1, q_2, q_3)$ for tuning the state weighting matrix $Q_f$
- Two values $(r_1, r_2)$ for tuning the control effort matrix $R_f$.

Fig. 5 (Type-1 fuzzy) and Fig. 6 (Type-2 fuzzy) illustrate the membership functions of the input variables, while Table 2 and Table 3 present the fuzzy rules that map these inputs to specific output values for the weighting matrices.

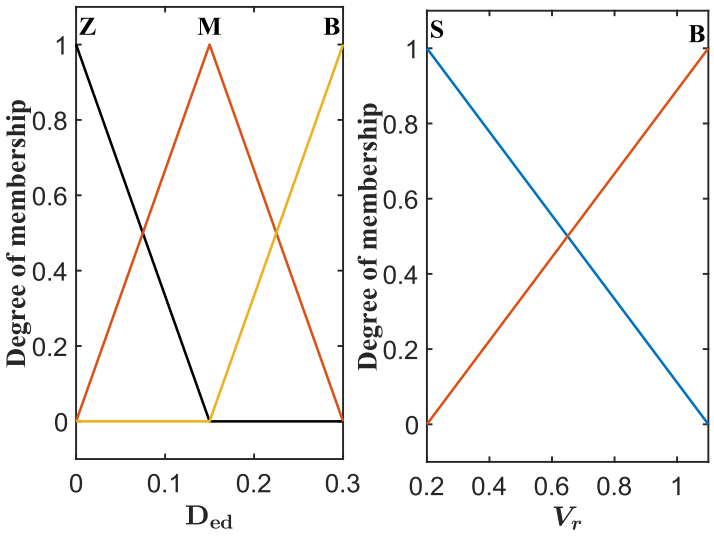


**Fig. 5.** Membership functions for the inputs of the Type-1 T-S fuzzy system

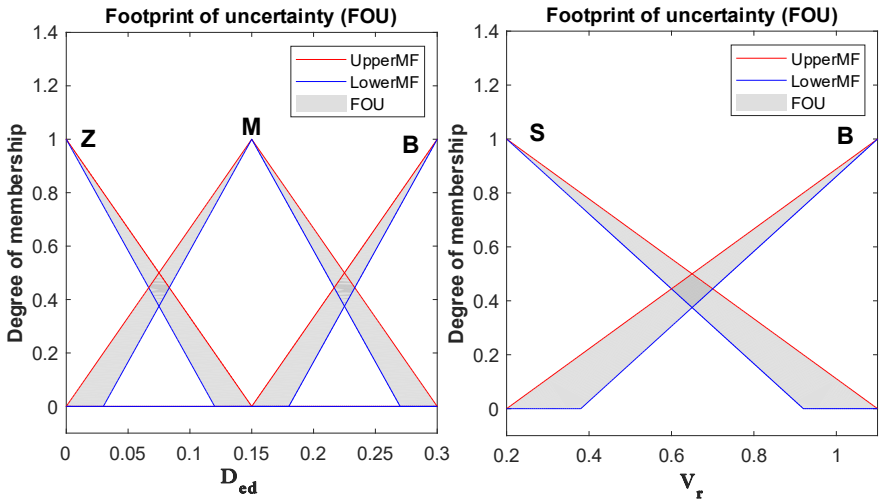


**Fig. 6.** Membership functions for the inputs of the Type-2 T-S fuzzy system

**Table 2.** Fuzzy rules for Q and R

| Rules | $D_{ed}$ | $v_r$ | $q_1$ | $q_2$ | $q_3$ | $r_1$ | $r_2$ |
|---|---|---|---|---|---|---|---|
| $R_1$ | Zero | Small | low | low | Medium | low | low |
| $R_2$ | Zero | Big | very low | low | high | low | low |
| $R_3$ | Medium | Small | Medium | Medium | low | Medium | Medium |
| $R_4$ | Medium | Big | Medium | Medium | low | Medium | Medium |
| $R_5$ | Big | Small | high | high | very low | high | high |
| $R_6$ | Big | Big | high | high | very low | high | high |

**Table 3.** Values of linguistic outputs

| Linguistic outputs | $q_1$ | $q_2$ | $q_3$ | $r_1$ | $r_2$ |
|---|---|---|---|---|---|
| **Very low** | 1 | / | 0.01 | / | / |
| **Low** | 2 | 80 | 0.03 | 0.0001 | 0.0001 |
| **Medium** | 5 | 90 | 0.05 | 0.001 | 0.001 |
| **High** | 10 | 100 | 2 | 0.01 | 0.01 |

The suggested control framework is tested using trajectory tracking scenarios and robustness tests after constructing the system model, which incorporates dynamic behaviour and predictive controller design. Organisation of the evaluation:

- Tracking a Circular Trajectory evaluates baseline performance on a smooth, closed-loop reference, allowing for comparison with existing methods in the literature.
- Tracking a Complex Trajectory (Circular followed by Sinusoidal) evaluates controller adaptability to dynamic and time-varying paths. The circular segment assesses consistency, while the sinusoidal segment assesses responsiveness to complex transitions.
- The Robustness and Resilience Test evaluates the controller's stability and effectiveness under external shocks.

### 4.1. Tracking a Circular Trajectory

In this scenario, the mobile robot is required to follow a circular trajectory defined mathematically by Equation (27):

$$\begin{cases} x_r(t) = 1.0 + \cos\left(\frac{2\pi t}{30}\right) \\ y_r(t) = 1.0 + \sin\left(\frac{2\pi t}{30}\right) \end{cases} \quad (27)$$

The reference robot follows this trajectory with a constant linear velocity of $v_r = 0.2$ m/s and an angular velocity of $w_r = 0.2$ rad/s. Its initial state is defined as $p(0) = [2.1 \quad 1.0 \quad 2.0]$. This setup provides a smooth, periodic path, suitable for evaluating the baseline tracking performance of the proposed control strategy under ideal conditions.

Fig. 7 illustrates the trajectory tracking performance of the mobile robot under MPC, Type-1 fuzzy MPC (F1MPC), and Type-2 fuzzy MPC (F2MPC) strategies. The results demonstrate robust convergence and precise adherence to the desired circular path, with the proposed fuzzy-enhanced methods offering improved tracking accuracy.

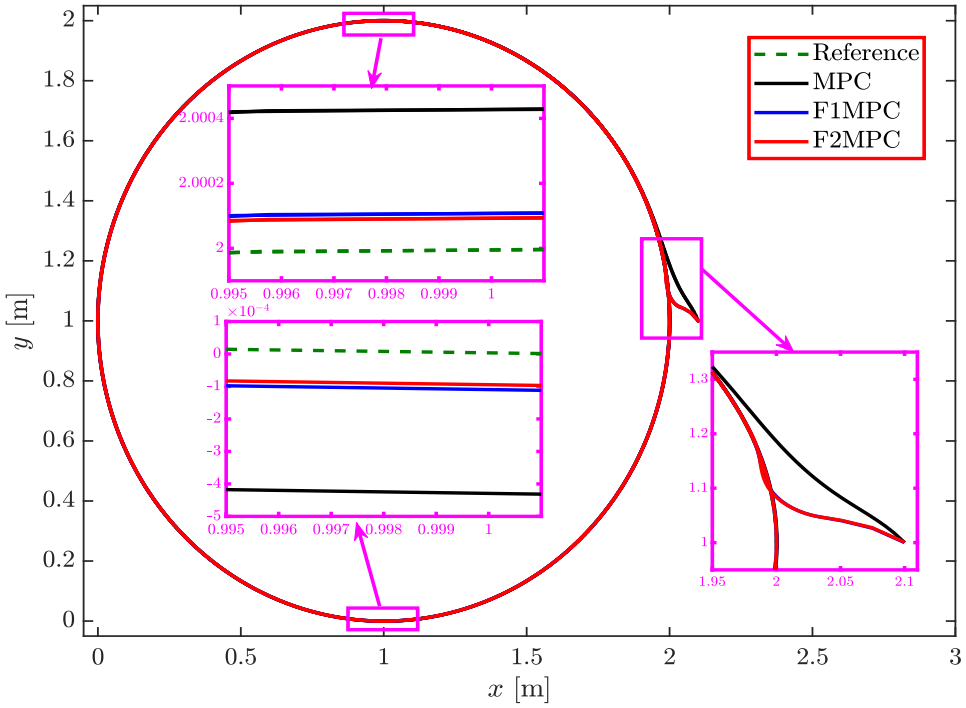


**Fig. 7.** Circular trajectory tracking results using the predictive control and proposed fuzzy model predictive control (F1MPC and F2MPC) methods

Fig. 8, Fig. 9, and Fig. 10 present the tracking error curves for the MPC, F1MPC, and F2MPC controllers during circular path tracking. Fig. 8 shows that the conventional MPC controller maintains relative stability but exhibits noticeable oscillations, particularly in the orientation error $e_\theta$, indicating limited damping capability. In contrast, Fig. 9 demonstrates improved performance by the F1MPC controller, which, despite experiencing slightly higher transient errors (reaching up to -1.2), manages to return to stability quickly. Fig. 10 further confirms the superior performance of the F2MPC controller, which achieves lower and shorter-lasting errors, highlighting the effectiveness of the Type-2 fuzzy control structure in providing smoother and more stable trajectory tracking compared to the other controllers.

Fig. 11, Fig. 12 to Fig. 13 compare the linear $v_r$ and angular $\omega_r$ velocity control inputs generated by MPC, Type-1 Fuzzy MPC (F1MPC), and Type-2 Fuzzy MPC (F2MPC) during circular trajectory tracking. The results reveal distinct control behaviors:

- MPC (Fig. 11): The linear velocity v_r drops abruptly from 0.5 to 0.2 m/s in 0.1 seconds before stabilising. In the same 0.5-second timeframe, the angular velocity ω_r oscillates from 2.3 rad/s to -0.5 rad/s before reaching steady-state near 0.2rad/s after 2.3 seconds.
- F1MPC (Fig. 12): The linear velocity v_r is more variable, starting at 0 m/s, peaking at 1.1 m/s, and returning to approximately 0.2 within 0.15 seconds. The angular velocity ω_r has a wide dynamic range, ranging from 10 rad/s to -4 rad/s in just 0.3 seconds, stabilising within 1.2 seconds.

- F2MPC controller (Fig. 13) has a similar behavioural pattern to F1MPC but stabilises in a shorter timescale (<1.2 seconds). This design has significantly more transient overrun than the F1MPC before equilibrium.

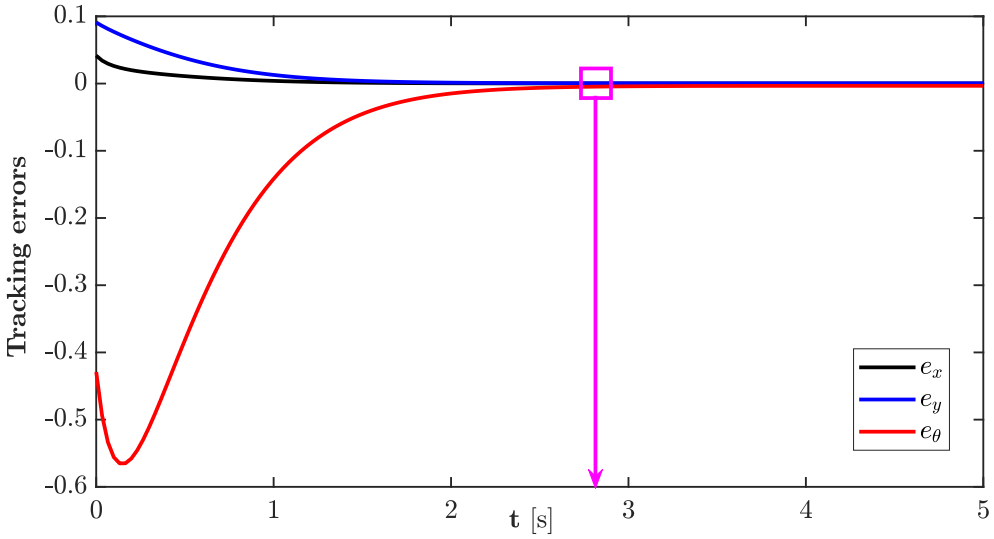


**Fig. 8.** Tracking errors using MPC approach

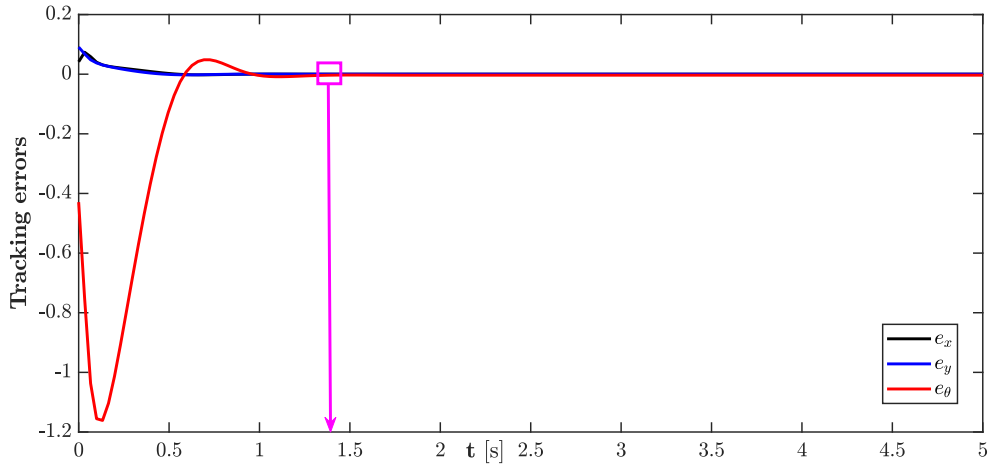


**Fig. 9.** Tracking errors using F1MPC approach

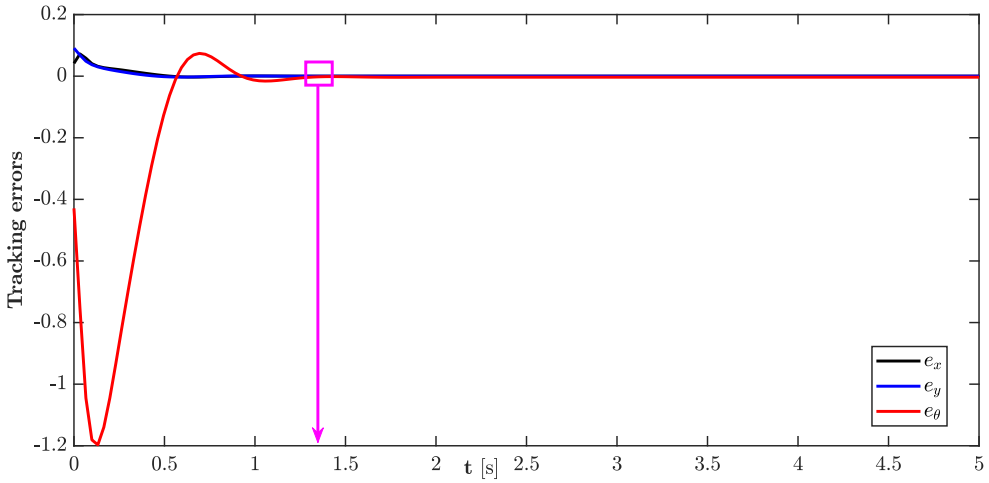


**Fig. 10.** Tracking errors using F2MPC approach

International Journal of Robotics and Control Systems
Vol. 5, No. 3, 2025, pp. 1919-1948
ISSN 2775-2658

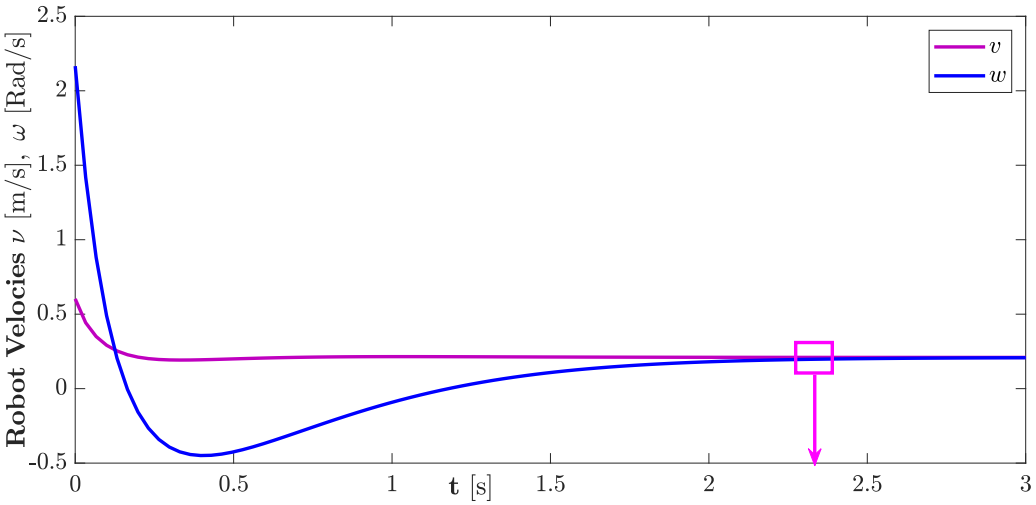


**Fig. 11.** Linear and angular speeds using MPC approach

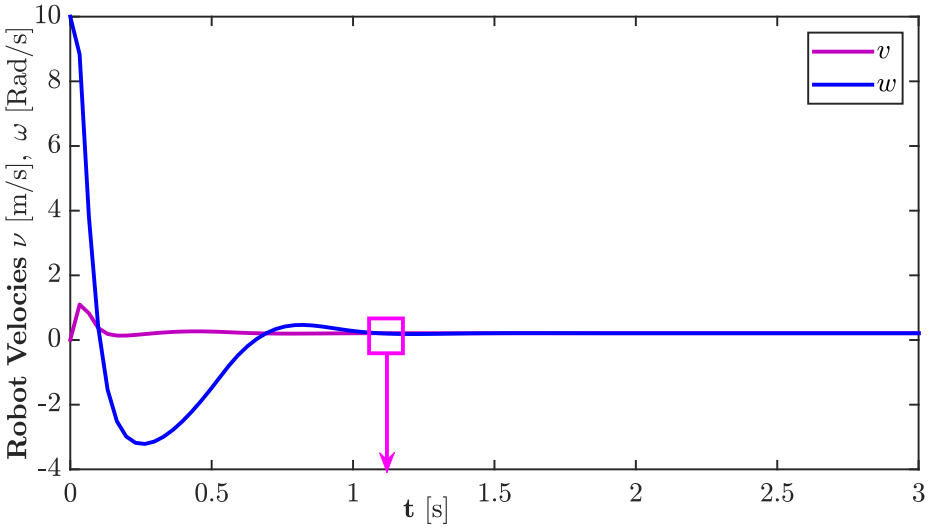


**Fig. 12.** Linear and angular speeds using F1MPC approach

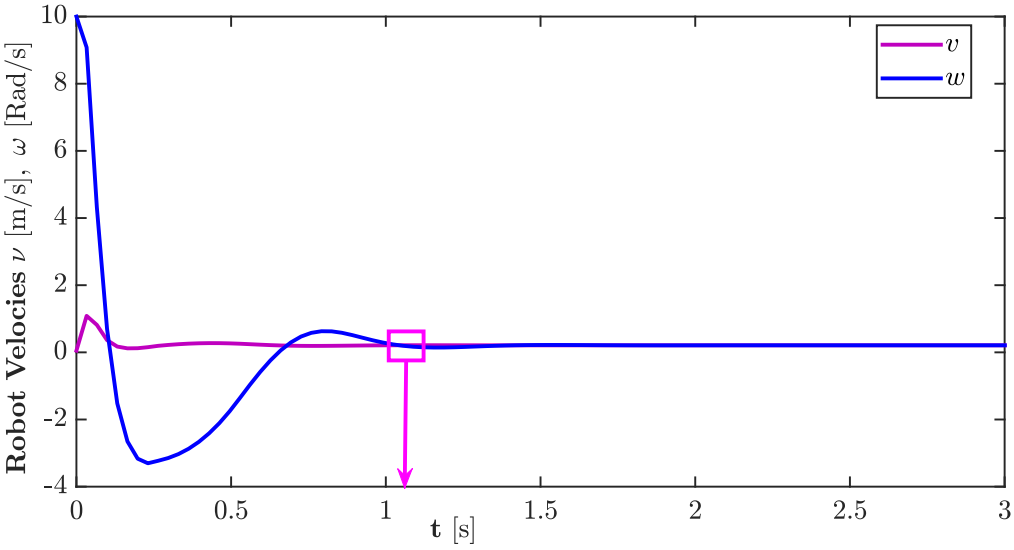


**Fig. 13.** Linear and angular speeds using F2MPC approach

### 4.2. Tracking Circular and Sinusoidal Trajectories

This study investigates a scenario where the mobile robot is tasked with following a predefined trajectory, mathematically represented by Equation (28):

$$\begin{cases} x_r(t) = 1.0 + \cos\left(\frac{2\pi t}{30}\right) \\ y_r(t) = 1.0 + \sin\left(\frac{2\pi t}{30}\right) \end{cases} \tag{28}$$

The main goal is to accurately guide the mobile robot along a dynamic path characterized by varying linear and rotational speeds. This trajectory is composed of an initial circular segment, defined by (27), followed by a sinusoidal segment. The robot's motion commences from $p(0) = [1.9 \quad 0.9 \quad \pi/3]$.

This combined trajectory evaluates the controller's adaptability to dynamic and time-varying paths: the circular segment measures consistency, while the sinusoidal segment tests responsiveness to more challenging transitions.

In this section, the results of the proposed fuzzy predictive control systems (F1MPC) and (F2MPC) are compared with those obtained from the model predictive control (MPC) developed in [31].

Fig. 14 illustrates the trajectory followed by the mobile robot under three distinct control scenarios: the first proposed technique (F1MPC), the second proposed technique (F2MPC), and the traditional control technique (MPC). This figure facilitates a direct comparison of the robot's response under different control commands, highlighting variations in tracking accuracy and smoothness of motion. Through this visual representation, the effectiveness of both proposed techniques (F1MPC and F2MPC) in achieving the desired trajectory can be directly evaluated against the traditional MPC approach.

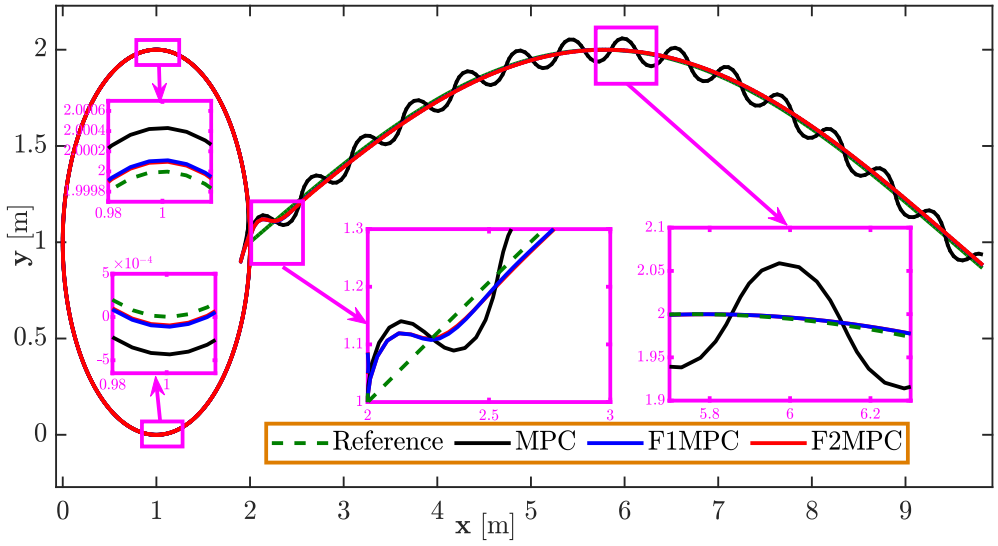


**Fig. 14.** Tracking circular and sinusoidal paths

The tracking error profiles corresponding to the composite trajectory provide a comparative evaluation of three control strategies: MPC (Fig. 15), Type-1 Fuzzy MPC (F1MPC) (Fig. 16), and Type-2 Fuzzy MPC (F2MPC) (Fig. 17). During the initial phase (0–30 s), where the robot follows a regular circular trajectory, F2MPC demonstrates superior tracking performance, characterized by low and stable errors across all state variables, particularly in the angular component $e_\theta$. F1MPC yields moderately improved results compared to MPC, which shows significant directional deviations and less stable error dynamics.

As the system transitions to the more complex sinusoidal path beyond 30 seconds, an increase in tracking errors is observed due to heightened curvature and directional variation. Nevertheless, F2MPC maintains smooth and consistent performance, effectively adapting to the increased trajectory complexity while keeping error magnitudes minimal. F1MPC also adapts reasonably well, albeit with a slight delay in responsiveness. In contrast, the MPC controller struggles to maintain accuracy under the higher dynamic demands, as evidenced by pronounced oscillations and degraded angular stability.

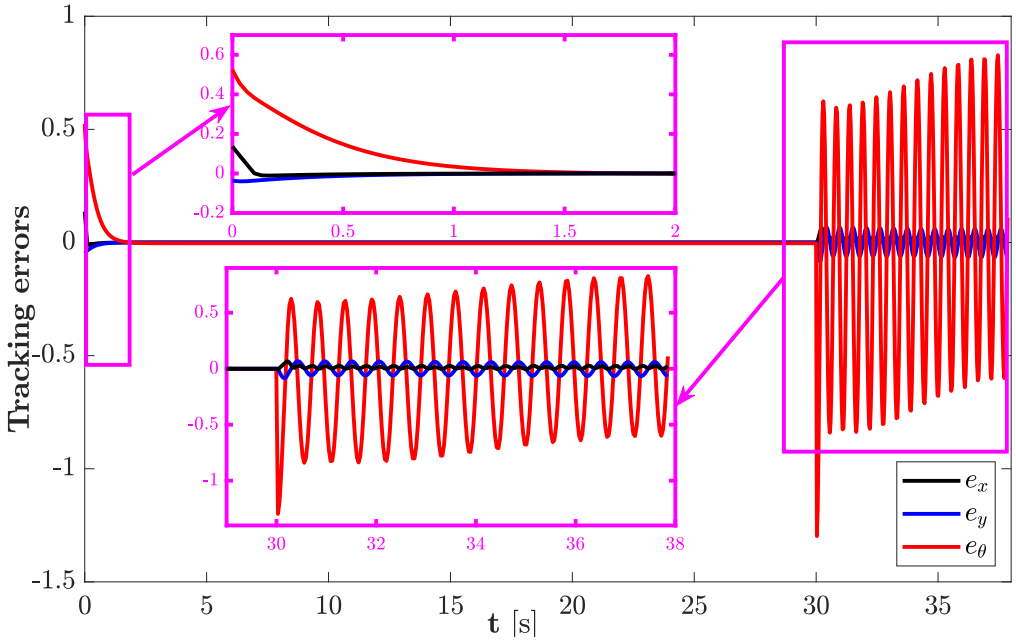


**Fig. 15.** Tracking errors using MPC approach

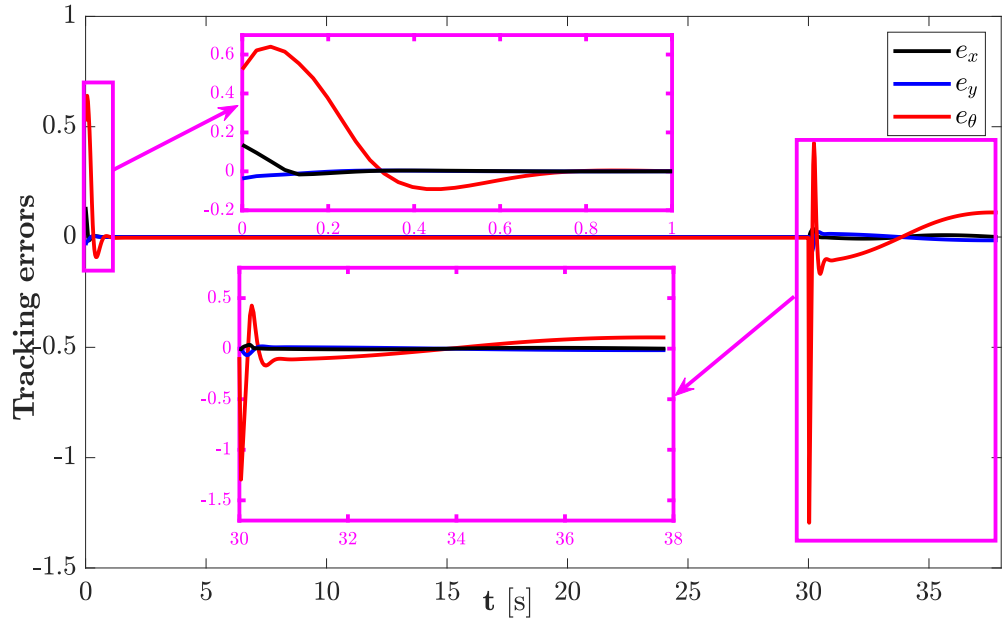


**Fig. 16.** Tracking errors using F1MPC approach

Fig. 18 and Fig. 19 provide a detailed view of the control inputs generated by the two proposed control techniques, (F1MPC) in Fig. 18 and (F2MPC) in Fig. 19, respectively. These figures display the actual values of the control signals (angular and linear velocities) applied to the robot over time. From these figures, the characteristics of the control inputs, such as their

smoothness, the range of values they vary within, and their response to changes in the reference trajectory, can be observed.

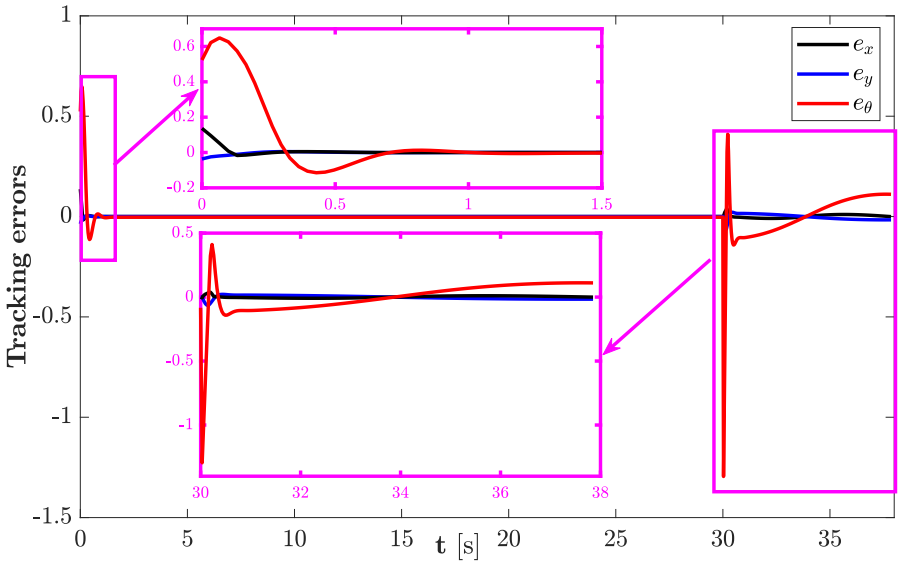


**Fig. 17.** Tracking errors using F2MPC approach

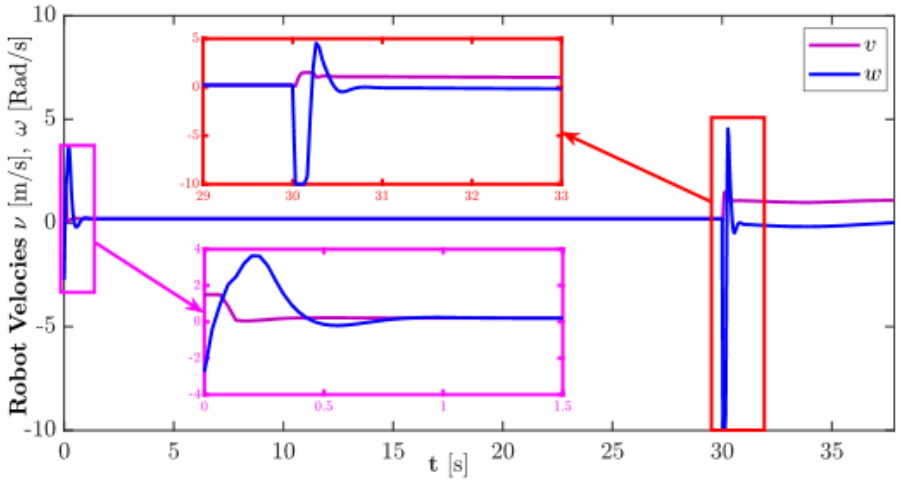


**Fig. 18.** Linear and angular speeds using F1MPC approach

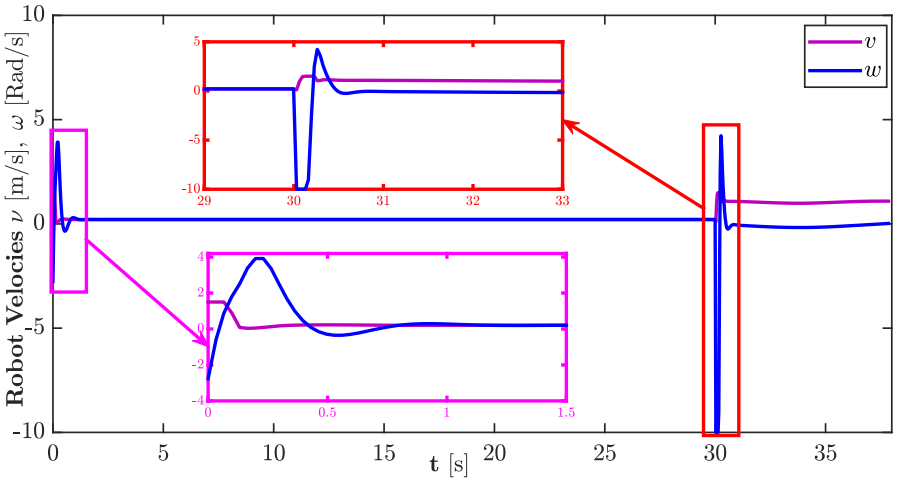


**Fig. 19.** Linear and angular speeds using F2MPC approach

In contrast, Fig. 20 shows the control inputs generated by the traditional (MPC) controller. By comparing these inputs with those shown in Fig. 18 and Fig. 19, the oscillations in input values can be observed, which may affect the robot's performance and stability.

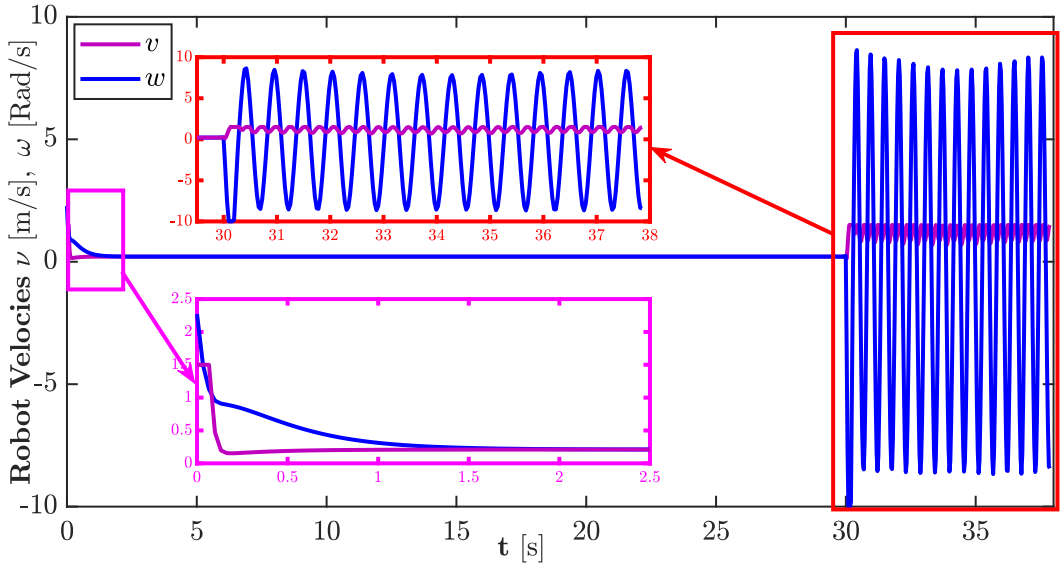


**Fig. 20.** Linear and angular speeds using MPC approach

To compare the performance of all controllers in terms of tracking accuracy, stability, and control effort, various metrics were adopted, including: Integral Absolute Error (IAE), Integral Squared Error (ISE), Integral Time Squared Error (ITSE), Integral Time Absolute Error (ITAE), Root Mean Square Error (RMSE), Tracking Variance, Integrated Squared Actuation (ISA), and Control Effort.

In terms of tracking accuracy, F2MPC exhibits the best overall performance. It consistently shows the lowest error values across all metrics (ISE, IAE, ITSE, ITAE, RMSE), as shown in Table 4, and across all three tracking components: position in x $e_x$, position in y $e_y$, and orientation $e_\theta$. For example, in the orientation error, F2MPC reduces the ISE from 2.2630 (in MPC) to 0.2392 and the ITAE from 126.8356 to 25.0638. These improvements reflect a more accurate and responsive controller that adapts efficiently to dynamic trajectories.

Stability, measured through tracking variance, further highlights the advantages of the fuzzy-based controllers. Both F1MPC and F2MPC exhibit near-zero tracking variance, indicating highly stable and smooth system behavior with minimal steady-state fluctuations. In contrast, MPC demonstrates a variance of 0.0599 in orientation tracking, suggesting the presence of oscillations and reduced smoothness during the steady-state period.

Regarding control effort and energy expenditure, as reported in Table 5, F2MPC again proves to be the most efficient. It shows the lowest values in both ISA (31.9548) and total control effort (973.9). F1MPC follows closely with slightly higher values, while MPC requires significantly more energy ISA of 291.5603 and control effort of 8875.6 indicating less efficient actuation.

In summary, F2MPC offers the most balanced solution by combining high tracking accuracy, excellent stability, and minimal energy consumption.

### 4.3. Robustness Test

In this section, a limited impulse-like disturbance was introduced into the robot's state vector during predefined time intervals, with the aim of evaluating the response of different controllers (MPC, F1MPC, and F2MPC) under the influence of unexpected perturbations. This disturbance consists of a three-element vector added directly to the state $[x; y; \theta]$, resulting in a sudden deviation in the robot's position and orientation. The disturbance was applied over a very narrow

time window (only one step per state), making it a momentary and fixed-magnitude perturbation rather than a continuous random one (such as white noise).

The purpose of this setup is to analyze the control system's ability to recover quickly from the deviation and to reestablish the reference trajectory after the disturbance occurs. This is an effective method for evaluating the robustness and resilience of tracking systems.

The results derived from Fig. 21 clearly demonstrate the superiority of the proposed techniques (F1MPC and F2MPC) in handling the effects of external disturbances compared to the traditional MPC controller. It is observed that the proposed techniques exhibit smaller deviations from the reference path and return to it more quickly after being subjected to disturbance.

**Table 4.** Comparison the error performance index of the controllers

| Controller Type | Error Type | ISE | IAE | ITSE | ITAE | RMSE | Tracking Variance |
|---|---|---|---|---|---|---|---|
| MPC | $e_x$ | 0.0028 | 0.1151 | 0.0694 | 3.4472 | 0.0090 | 0.0001 |
| | $e_y$ | 0.0159 | 0.3395 | 0.5236 | 10.7344 | 0.0205 | 0.0004 |
| | $e_\theta$ | 2.2630 | 3.9874 | 74.6035 | 126.8356 | 0.2447 | 0.0599 |
| F1MPC | $e_x$ | 0.0013 | 0.0676 | 0.0178 | 1.8717 | 0.0064 | 0.0000 |
| | $e_y$ | 0.0017 | 0.0999 | 0.0530 | 3.1752 | 0.0067 | 0.0000 |
| | $e_\theta$ | 0.2399 | 0.9599 | 5.2067 | 24.8545 | 0.0804 | 0.0065 |
| F2MPC | $e_x$ | 0.0011 | 0.0595 | 0.0137 | 1.5939 | 0.0062 | 0.0000 |
| | $e_y$ | 0.0015 | 0.0939 | 0.0474 | 2.9609 | 0.0064 | 0.0000 |
| | $e_\theta$ | 0.2392 | 0.9586 | 5.2848 | 25.0638 | 0.0802 | 0.0064 |

**Table 5.** Evaluation of control actuation intensity using isa and control effort

| Controller | ISA | Control Effort |
|---|---|---|
| **MPC** | 291.5603 | 8875.6 |
| **F1MPC** | 32.1313 | 979.2 |
| **F2MPC** | 31.9548 | 973.9 |

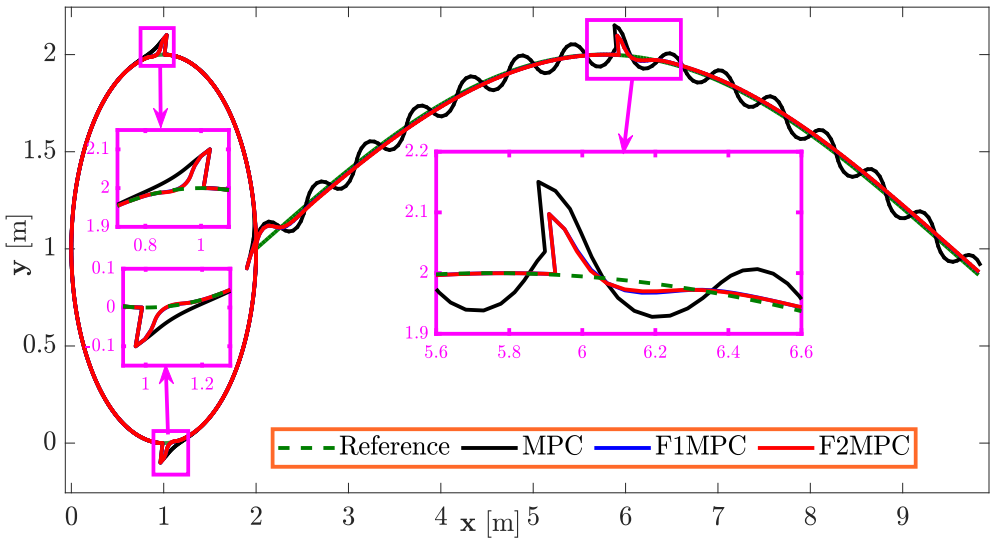


**Fig. 21.** Robustness of tracking circular and sinusoidal paths

The tracking error plots for the MPC, F1MPC, and F2MPC controllers under the influence of impulse-like disturbances added to the state vector $[x; y; \theta]$ clearly show distinct differences in each controller's ability to reject disturbances and restore the reference trajectory. Fig. 22 demonstrates that the traditional MPC controller suffers from significant deviations, particularly in the orientation error $e_\theta$, accompanied by a noticeable delay in stabilization and extended oscillations after the disturbance occurs.

In contrast, Fig. 23 and Fig. 24 show that both F1MPC and F2MPC controllers exhibit faster and more stable responses. The tracking errors remain tightly bounded and quickly return to their reference values following the perturbation. The zoomed-in insets within the plots emphasize that F2MPC delivers superior dynamic performance compared to F1MPC, with enhanced damping and quicker convergence highlighting the effectiveness of the Type-2 fuzzy logic structure in improving tracking robustness and accuracy.

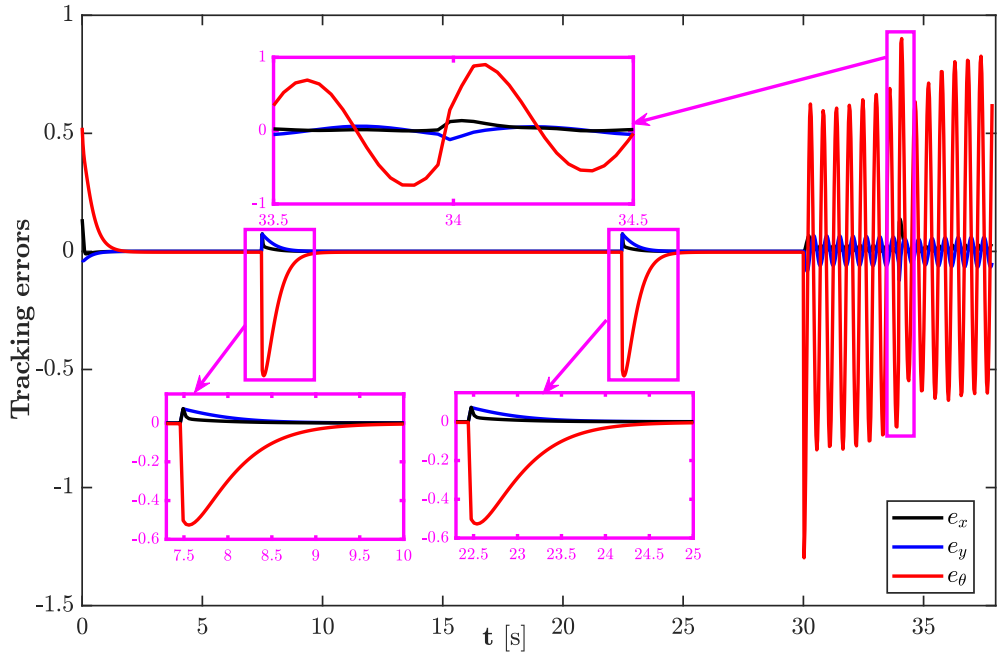


**Fig. 22.** Tracking errors using MPC approach

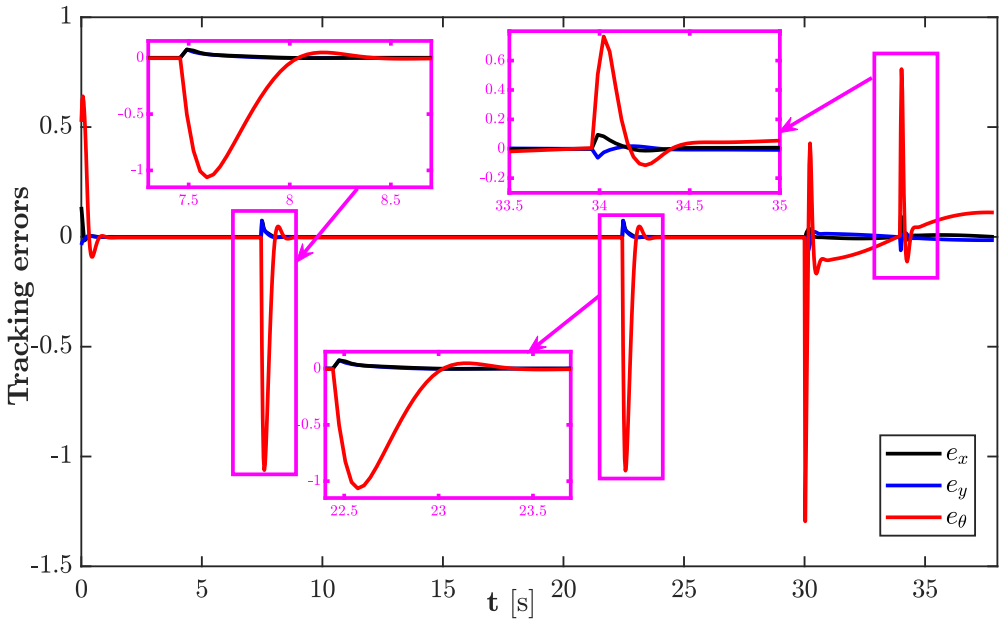


**Fig. 23.** Tracking errors using F1MPC approach

Fig. 25 and Fig. 26 provide details on the robot's velocity response to disturbances when using the proposed techniques (F1MPC and F2MPC), respectively. These figures show a rapid dynamic adjustment in the robot's linear and angular velocities to efficiently restore the reference path after the disturbance. In contrast, Fig. 27 presents similar results but for the MPC controller, where slower velocity responses and larger oscillations are observed.

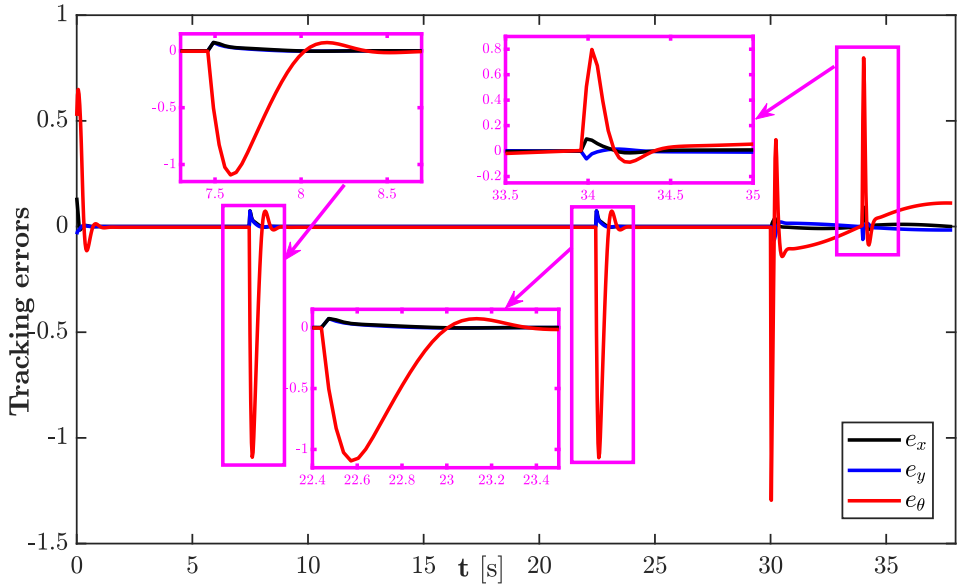


**Fig. 24.** Tracking errors using F2MPC approach

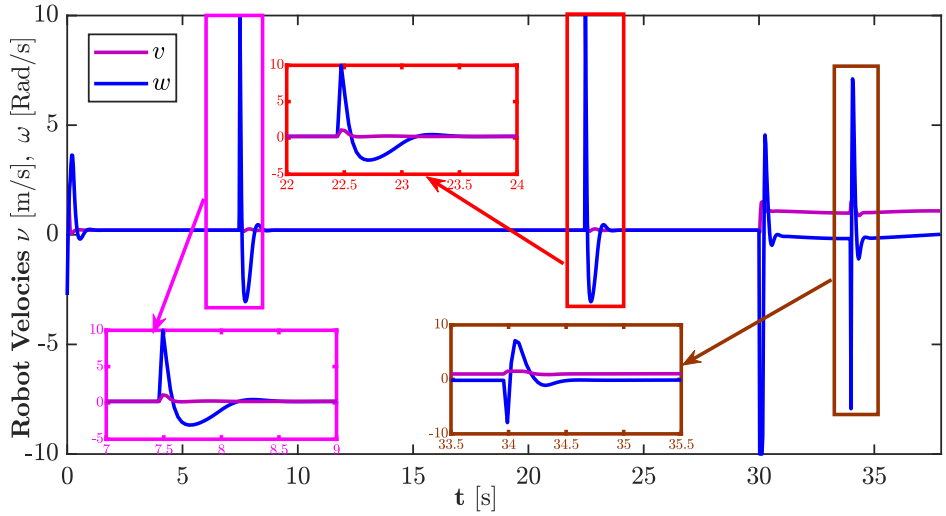


**Fig. 25.** Illustrates the robustness test of angular and linear speeds using the F1MPC approach

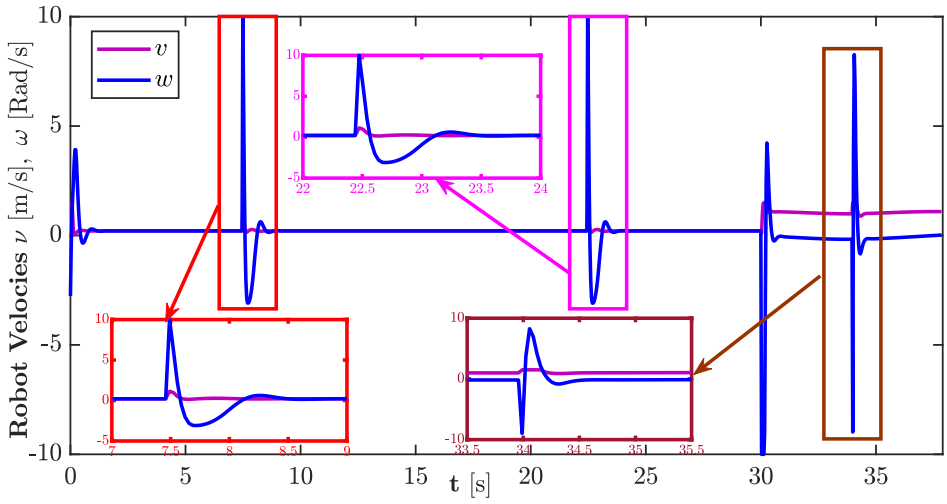


**Fig. 26.** Illustrates the robustness test of angular and linear speeds using the F2MPC approach

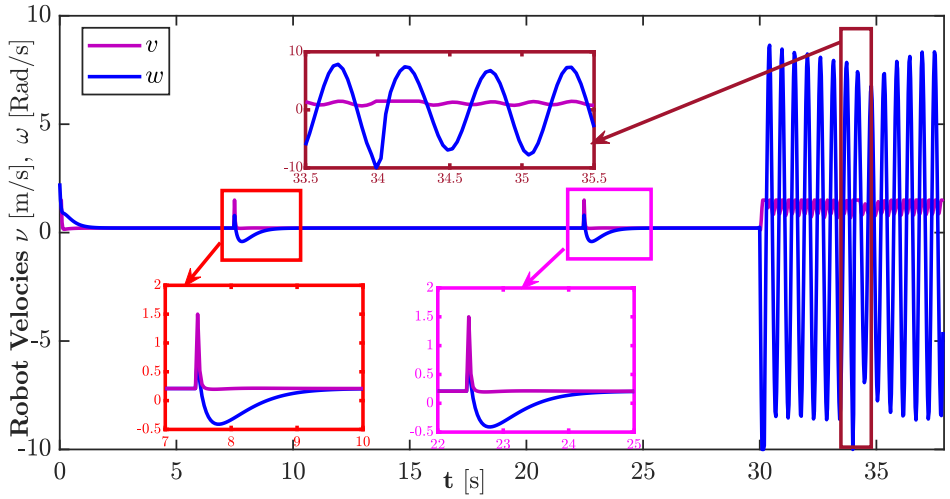


**Fig. 27.** Illustrates the robustness test of angular and linear speeds using the MPC approach

The results presented in Table 6 and Table 7 provide a detailed and comprehensive comparison of the performance of the three controllers (MPC, F1MPC, and F2MPC) under robustness testing. According to Table 6, the conventional MPC controller records the highest error values, particularly in the orientation error $e_\theta$, with ISE = 2.4647, ITAE = 136.2406, and RMSE = 0.2556. These values indicate a weak ability to reject disturbances and restore stability quickly. Similarly, positional errors in MPC are clearly observed for example, in the y-axis, ISE = 0.0189 and ITAE = 11.7845.

In contrast, both F1MPC and F2MPC exhibit significant improvements, with considerably lower values across all performance indices. For the orientation error $e_\theta$ , F1MPC records ISE = 0.8218 and ITAE = 38.3144, while F2MPC performs slightly better with ISE = 0.7957 and ITAE = 38.3338. Furthermore, the RMSE is notably reduced in F2MPC to 0.1454, compared to 0.1477 in F1MPC and 0.2556 in MPC. Positional tracking errors are also substantially lower, for instance, in the y-direction, F2MPC achieves ISE = 0.0025 and ITAE = 3.4571, compared to 0.0189 and 11.7845, respectively, in MPC. Regarding tracking variance, both F1MPC and F2MPC maintain near-zero values (e.g., 0.0001) across all error components, indicating high stability and precise performance, whereas MPC shows noticeable fluctuation in orientation, with a variance of 0.0650.

Analyzing control effort and energy consumption in Table 7 further confirms F2MPC's superiority in energy efficiency, where it records the lowest ISA value (52.9609) and total control effort (1610.4), compared to F1MPC with ISA = 54.9244 and Control Effort = 1669.9. In contrast, MPC requires significantly higher energy input, with ISA = 283.7293 and Control Effort = 8613.1, reflecting less efficient control and actuation. These results clearly demonstrate that the F2MPC controller offers the best balance of tracking accuracy, minimal deviation under disturbances, and energy efficiency.

**Table 6.** Comparison the error performance index of the controllers

| Controller Type | Error Type | ISE | IAE | ITSE | ITAE | RMSE | Tracking Variance |
|---|---|---|---|---|---|---|---|
| MPC | $e_x$ | 0.0057 | 0.1602 | 0.1557 | 4.4730 | 0.0126 | 0.0001 |
| | $e_y$ | 0.0189 | 0.4117 | 0.5678 | 11.7845 | 0.0224 | 0.0005 |
| | $e_\theta$ | 2.4647 | 4.6628 | 76.6240 | 136.2406 | 0.2556 | 0.0650 |
| F1MPC | $e_x$ | 0.0030 | 0.1043 | 0.0586 | 2.6307 | 0.0094 | 0.0001 |
| | $e_y$ | 0.0027 | 0.1255 | 0.0711 | 3.6467 | 0.0084 | 0.0001 |
| | $e_\theta$ | 0.8218 | 1.7343 | 14.9896 | 38.3144 | 0.1477 | 0.0216 |
| F2MPC | $e_x$ | 0.0029 | 0.0955 | 0.0539 | 2.3423 | 0.0092 | 0.0001 |
| | $e_y$ | 0.0025 | 0.1203 | 0.0662 | 3.4571 | 0.0082 | 0.0001 |
| | $e_\theta$ | 0.7957 | 1.7138 | 14.6780 | 38.3338 | 0.1454 | 0.0209 |

**Table 7.** Evaluation of control actuation intensity using ISA and control effort

| Controller | ISA | Control Effort |
|---|---|---|
| **MPC** | 283.7293 | 8613.1 |
| **F1MPC** | 54.9244 | 1669.9 |
| **F2MPC** | 52.9609 | 1610.4 |

## 5. Conclusion

This study presented an advanced trajectory tracking control methodology for nonholonomic mobile robots by integrating Type-2 fuzzy logic with MPC, addressing dynamic environments and system uncertainties. The Fuzzy Type-2 MPC (F2MPC) showed better results, achieving a position accuracy score of 0.0011, which is better than the Type-1 fuzzy MPC score of 0.0013 and the conventional MPC score of 0.0028. The F2MPC showed great energy efficiency, using 65% less control effort than MPC, and it handled disturbances well with very little change (0.0001) and a smaller orientation error (RMSE of 0.1454 compared to 0.2556 for MPC). The dynamic adaptability of the T-S fuzzy model enabled real-time adjustments, ensuring stability during complex trajectory transitions. Future work will focus on computational optimisation for real-time implementation, experimental validation in unstructured contexts, and hybrid machine learning approaches to improve flexibility. The F2MPC framework improves robotic control with precision, efficiency, and resilience for real-world applications.

**Author Contribution:** All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

Table 8 summarizes the metrics used to evaluate tracking performance. These metrics are computed for each posture error component ($ex, ey, e\theta$) in Section 4.

**Table 8.** Performance metrics for trajectory tracking evaluation

| Metric | Formula | Description |
|---|---|---|
| Integral Absolute Error (IAE) | $IAE = \int_0^T \lvert e(t) \rvert dt$ | The Integral Absolute Error (IAE) is a performance metric that quantifies the total accumulated magnitude of the tracking error over time. |
| Root Mean Square Error (RMSE) | $RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{n} e(k)^2}$ | Average error magnitude, sensitive to large deviations (e.g., outliers). |
| Integral Squared Error (ISE) | $ISE = \int_0^T e^2(t) dt$ | ISE measures the cumulative squared tracking error over time, emphasizing large deviations more than small ones. |
| Integral Time Squared Error (ITSE) | $ITSE = \int_0^T t.e^2(t) dt$ | The Integral Time Squared Error (ITSE) is a performance index used to evaluate the quality of control in dynamic systems, particularly in trajectory |

| | | |
|---|---|---|
| | | tracking tasks. |
| Integral Time Absolute Error (ITAE) | $ITAE = \int_0^T t.\|e(t)\|dt$ | Time-weighted performance metrics like the Integral Time Absolute Error (ITAE) assess control system tracking accuracy and transient behaviour. |
| Tracking Variance | $Var(e) = \frac{1}{n+1}\sum_{i=1}^{n}(e_i - \bar{e})^2$ | During system response's steady-state period, tracking variance quantifies the tracking error signal's fluctuation or dispersion. |
| ISA (Integrated Squared Actuation) | $ISA = \int_0^T u^2(t)dt$ | An Integrated Squared Actuation (ISA) performance metric measures a system's energy or control effort over time. |
| Settling Time (Ts) | $Ts = min\left\{t \geq 0 \left\| \frac{\|e(\tau)\|}{\|e_{ref}\|} \leq \delta \forall \tau \geq t \right.\right\}$ | Time for error to stay within ±2% of the reference. |
| Control Effort Metrics | $E = \sum_{k=1}^{N}(v(k)^2 + \omega(k)^2)$ | This shows how efficiently each method achieves tracking |

***Key:*** *$e(k)$: Traking error at time t (can be $e_x, e_y, or e_\theta$) , $e_{ref}$: Reference error (e.g., final desired error, often zero) , $v(k)$: Linear velocity (m/s) at step k and $\omega(k)$ : Angular velocity (rad/s) at step k.*

## Appendix B

Table 9 summarizes the key mathematical expressions, variables, and their physical meanings.

**Table 9.** Key variables and notations in the fuzzy predictive control framework

| Symbol | Description | Units | Physical Meaning |
|---|---|---|---|
| $e_x, e_y$ | Position tracking errors | m | Deviation in x- and y-axis from the reference trajectory. |
| $e_\theta$ | Orientation tracking error | rad | Angular deviation from the reference heading. |
| $u_r$ | Reference linear velocity | m/s | Desired forward/backward speed of the robot. |
| $w_r$ | Reference angular velocity | rad/s | Desired rotational speed of the robot. |
| $Q_f$ | State weighting matrix | Dimensionless | Penalizes tracking errors in the cost function (tunes accuracy). |
| $R_f$ | Control effort weighting matrix | Dimensionless | Penalizes actuator effort (tunes energy efficiency). |
| $N_p$ | Prediction horizon | steps | Number of future time steps optimized by MPC. |
| $A(k), B(k)$ | State-space matrices | Dimensionless | System dynamics adjusted by T-S fuzzy rules. |
| $\mu_i(k)$ | Membership value | Dimensionless ∈[0,1] | Degree of activation for the ithith fuzzy rule. |
| $J(u_p, k)$ | Cost function | Dimensionless | Quantifies trade-off between tracking performance and energy use. |
| $u_p$ | Control input vector | [m/s,rad/s] | Computed linear (v) and angular (ω) velocities. |
| $D_{ed}$ | Distance error | m | Euclidean distance between robot and reference path. |

## References

[1] A. Ollero, J. Ferruz, O. Sánchez, and G. Heredia, "Mobile robot path tracking and visual target tracking using fuzzy logic," *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*, vol. 61, pp. 51-72, 2001, https://doi.org/10.1007/978-3-7908-1835-2_3.

[2] I. Maurović, M. Baotić and I. Petrović, "Explicit Model Predictive Control for trajectory tracking with mobile robots," *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 712-717, 2011, https://doi.org/10.1109/AIM.2011.6027140.

[3] K. Kanjanawaniskul, "Motion control of a wheeled mobile robot using model predictive control: A survey," *KKU Research Journal*, vol. 17, no. 5, pp. 811-837, 2012, https://so01.tci-thaijo.org/index.php/APST/article/view/83293.

[4] D. N. M. Abadi and M. H. Khooban, "Design of optimal Mamdani-type fuzzy controller for nonholonomic wheeled mobile robots," *Journal of King Saud University – Engineering Sciences*, vol. 27, no. 1, pp. 92-100, 2013, https://doi.org/10.1016/j.jksues.2013.05.003.

[5] K. R. Sharma, D. Honc, and F. Dušek, "Model predictive control of trajectory tracking of differentially steered mobile robot," *Intelligent Data Analysis and Applications*, vol. 370, pp. 85-95, 2015, https://doi.org/10.1007/978-3-319-21206-7_8.

[6] K. R. Sharma, D. Honc, and F. Dušek, "Predictive control of differential drive mobile robot considering dynamics and kinematics," *30th European Conference on Modelling and Simulation*, pp. 354-360, 2016, http://dx.doi.org/10.7148/2016-0354.

[7] I. Škrjanc and G. Klančar, "A comparison of continuous and discrete tracking-error model-based predictive control for mobile robots," *Robotics and Autonomous Systems*, vol. 87, pp. 177-187, 2017, https://doi.org/10.1016/j.robot.2016.09.016.

[8] N. T. Thinh, N. T. Tuan and L. P. Hung, "Predictive Controller for Mobile Robot Based on Fuzzy Logic," *2016 3rd International Conference on Green Technology and Sustainable Development (GTSD)*, pp. 141-144, 2016, https://doi.org/10.1109/GTSD.2016.41.

[9] P. Kumar, B. Anoohya, and B. Padhi, "Model predictive static programming for optimal command tracking: A fast model predictive control paradigm," *Journal of Dynamic Systems, Measurement and Control*, vol. 141, no. 2, p. 021014, 2019, https://doi.org/10.1115/1.4041356.

[10] S. Peng and W. Shi, "Adaptive fuzzy integral terminal sliding mode control of a nonholonomic wheeled mobile robot," *Mathematical Problems in Engineering*, vol. 2017, no. 1, pp. 1-12, 2017, https://doi.org/10.1155/2017/3671846.

[11] M. Krid, F. Benamar, and R. Lenain, "A new explicit dynamic path tracking controller using generalized predictive control," *International Journal of Control, Automation and Systems*, vol. 15, pp. 303-314, 2017, https://doi.org/10.1007/s12555-015-0160-6.

[12] A. Alouache and Q. Wu, "Fuzzy logic PD controller for trajectory tracking of an autonomous differential drive mobile robot (i.e. Quanser Qbot)," *Industrial Robot*, vol. 45, no. 1, pp. 23-33, 2018, https://doi.org/10.1108/IR-07-2017-0128.

[13] B. Ladjal *et al*., "Hybrid models for direct normal irradiance forecasting: A case study of Ghardaia zone (Algeria)," *Natural Hazards*, vol. 120, no. 15, pp. 14703-14725, 2024, https://doi.org/10.1007/s11069-024-06837-1.

[14] S. Bouzoualegh, E. Guechi, and R. Kelaiaia, "Model predictive control of a differential-drive mobile robot," *Acta Universitatis Sapientiae Electrical and Mechanical Engineering*, vol. 10, no. 1, pp. 20-41, 2018, https://doi.org/10.2478/auseme-2018-0002.

[15] Y. Chen, H. Kong, Z. Li and F. Ke, "Trajectory-Tracking for a Mobile Robot Using Robust Predictive Control and Adaptive Control," *2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 30-35, 2018, https://doi.org/10.1109/ICARM.2018.8610695.

[16] B. Nail, B. Djaidir, I. E. Tibermacine, C. Napoli, N. Haidour, and R. Abdelaziz, "Gas turbine vibration monitoring based on real data and neuro-fuzzy system," *Diagnostyka*, vol. 25, no. 1, p. 2024108, 2024, https://doi.org/10.29354/diag/181190.

[17] S. Peng and W. Shi, "Adaptive Fuzzy Output Feedback Control of a Nonholonomic Wheeled Mobile Robot," *IEEE Access*, vol. 6, pp. 43414-43424, 2018, https://doi.org/10.1109/ACCESS.2018.2862163.

[18] C. Napoli, V. Ponzi, A. Puglisi, S. Russo, I. E. Tibermacine, "Exploiting robots as healthcare resources for epidemics management and support caregivers," *CEUR Workshop Proceedings*, vol. 3686, pp. 1-10, 2024, https://ceur-ws.org/Vol-3686/short1.pdf.

[19] X. Wu, P. Jin, T. Zou, Z. Qi, H. Xiao, and P. Lou, "Backstepping trajectory tracking based on fuzzy sliding mode control for differential mobile robots," *Journal of Intelligent & Robotic Systems*, vol. 96, pp. 109-121, 2019, https://doi.org/10.1007/s10846-019-00980-9.

[20] T. Q. Khai, Y. Ryoo, W. Gill, and D. Im, "Design of kinematic controller based on parameter tuning by fuzzy inference system for trajectory tracking of differential-drive mobile robot," *International Journal of Fuzzy Systems*, vol. 22, pp. 1972-1978, 2020, https://doi.org/10.1007/s40815-020-00842-9.

[21] M. Abdelwahab, V. Parque, A. M. R. Fath Elbab, A. A. Abouelsoud and S. Sugano, "Trajectory Tracking of Wheeled Mobile Robots Using Z-Number Based Fuzzy Logic," *IEEE Access*, vol. 8, pp. 18426-18441, 2020, https://doi.org/10.1109/ACCESS.2020.2968421.

[22] K.D. Sommer, P. Berner, and M. Mönnigmann, "Reducing the number of optimization problems in trajectory tracking model predictive control," *European Control Conference (ECC)*, pp. 614-619, 2021, https://doi.org/10.23919/ECC54610.2021.9654991.

[23] W. Benchouche, R. Mellah, and M.S. Bennouna, "Navigation of a differential drive mobile robot using nonlinear model predictive control," *The 2nd International Conference on Complex Systems and their Applications*, 2021, https://ceur-ws.org/Vol-2904/55.pdf.

[24] B. Moudoud, H. Aissaoui, and M. Diany, "Fuzzy adaptive sliding mode controller for electrically driven wheeled mobile robot for trajectory tracking task," *Journal of Control and Decision*, vol. 9, no. 1, pp. 71-79, 2021, https://doi.org/10.1080/23307706.2021.1912665.

[25] A. Noordin *et al.*, "A performance comparison of PID and fuzzy logic control methods for trajectory tracking of wheeled mobile robot," *International Journal of Nanoelectronics and Materials*, vol. 14, no. 1, pp. 387-394, 2021, http://dspace.unimap.edu.my:80/xmlui/handle/123456789/73365.

[26] B. Nail, A. Kouzou, A. Hafaifa, N. Hadroug, and V. Puig, "A robust fault diagnosis and forecasting approach based on Kalman filter and interval type-2 fuzzy logic for efficiency improvement of centrifugal gas compressor system," *Diagnostyka*, vol. 20, no. 2, pp. 57-75, 2019, https://doi.org/10.29354/diag/108613.

[27] B. Nail, M. A. Atoussi, S. Saadi, I. E. Tibermacine, and C. Napoli, "Real-Time Synchronisation of Multiple Fractional-Order Chaotic Systems: An Application Study in Secure Communication," *Fractal and Fractional*, vol. 8, no. 2, p. 104, 2024, https://doi.org/10.3390/fractalfract8020104.

[28] R. L. S. Sousa, M. D. do Nascimento Forte, F. G. Nogueira and B. C. Torrico, "Trajectory tracking control of a nonholonomic mobile robot with differential drive," *2016 IEEE Biennial Congress of Argentina (ARGENCON)*, pp. 1-6, 2016, https://doi.org/10.1109/ARGENCON.2016.7585356.

[29] M. L. Hadjili and K. Kara, "Modelling and control using Takagi-Sugeno fuzzy models," *2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC)*, pp. 1-6, 2011, https://doi.org/10.1109/SIECPC.2011.5876946.

[30] M. E. Hedroug, E. Bdirina and K. Guesmi, "Fuzzy Predictive Controller for Trajectory Tracking of Differential-Drive Mobile Robot," *2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM)*, pp. 1-6, 2023, https://doi.org/10.1109/IC2EM59347.2023.10419477.

[31] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robotics and Autonomous Systems*, vol. 55, no. 6, pp. 460-469, 2007, https://doi.org/10.1016/j.robot.2007.01.002.

[32] N. H. Hadi and K. K. Younus, "Path tracking and backstepping control for a wheeled mobile robot (WMR) in a slipping environment," *IOP Conference Series Materials Science and Engineering*, vol. 671, no. 1, p. 012005, 2020, https://doi.org/10.1088/1757-899X/671/1/012005.

[33] G. A. R. Ibraheem, A. T. Azar, I. K. Ibraheem, and A. J. Humaidi, "A novel design of a neural Network-Based fractional PID controller for mobile robots using hybridized fruit fly and particle swarm optimization," *Complexity*, vol. 2020, no. 1, pp. 1-18, 2020, https://doi.org/10.1155/2020/3067024.

[34] H. Xie, J. Zheng, R. Chai, and H. T. Nguyen, "Robust tracking control of a differential drive wheeled mobile robot using fast nonsingular terminal sliding mode," *Computers & Electrical Engineering*, vol. 96, p. 107488, 2021, https://doi.org/10.1016/j.compeleceng.2021.107488.

[35] K. Sehgal, S. Upadhyaya, Keshav, M. Verma and M. M. Rayguru, "Backstepping based Trajectory Tracking Control of a Class of Reconfigurable Mobile Robot," *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, pp. 752-756, 2021, https://doi.org/10.1109/ICAC3N53548.2021.9725399.

[36] G. Díaz-García, L. F. Giraldo and S. Jimenez-Leudo, "Dynamics of a Differential Wheeled Robot: Control and Trajectory Error Bound," *2021 IEEE 5th Colombian Conference on Automatic Control (CCAC)*, pp. 25-30, 2021, https://doi.org/10.1109/CCAC51819.2021.9633318.

[37] N. Hassan and A. Saleem, "Analysis of Trajectory Tracking Control Algorithms for Wheeled Mobile Robots," *2021 IEEE Industrial Electronics and Applications Conference (IEACon)*, pp. 236-241, 2021, https://doi.org/10.1109/IEACon51066.2021.9654675.

[38] W. Benchouche, R. Mellah, and M. S. Bennouna, "The impact of the dynamic model in feedback linearization trajectory tracking of a mobile robot," *Periodica Polytechnica Electrical Engineering and Computer Science*, vol. 65, no. 4, pp. 329-343, 2021, https://doi.org/10.3311/PPee.17127.

[39] N. Hacene, B. Mendil, M. Bechouat, and R. Sadouni, "Comparison between fuzzy and non-fuzzy Ordinary IF–Then Rule-Based control for the trajectory tracking of a differential drive robot," *International Journal of Fuzzy Systems*, vol. 24, no. 8, pp. 3666-3687, 2022, https://doi.org/10.1007/s40815-022-01365-1.

[40] F. Lawless *et al.*, "Trajectory Tracking for Mobile Robots with Human-in-the-Loop," *SoutheastCon 2022*, pp. 385-390, 2022, https://doi.org/10.1109/SoutheastCon48659.2022.9764114.

[41] J. Pan, K. Chen, Y. Li and Q. Gao, "A Path tracking controller design method for wheeled mobile robots: one dimensional case," *2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, pp. 1-6, 2022, https://doi.org/10.1109/CVCI56766.2022.9965137.

[42] H. Rodríguez-Cortés and M. Velasco-Villa, "A new geometric trajectory tracking controller for the unicycle mobile robot," *Systems & Control Letters*, vol. 168, p. 105360, 2022, https://doi.org/10.1016/j.sysconle.2022.105360.

[43] M. Siwek, J. Panasiuk, L. Baranowski, W. Kaczmarek, and S. Borys, "Trajectory Tracking Control of a Mobile Robot with the ROS System," *Problems of Mechatronics Armament Aviation Safety Engineering*, vol. 13, no. 4, pp. 67-84, 2022, https://doi.org/10.5604/01.3001.0016.1458.

[44] M. Auzan, R. M. Hujja, M. R. Fuadin, and D. Lelono, "Swarm Intelligence Autotune for differential Drive wheeled Mobile Robot," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 10, no. 3, pp. 644-654, 2022, https://doi.org/10.52549/ijeei.v10i3.3528.

[45] J. I. Aguilar-Pérez, M. Velasco-Villa, R. Castro-Linares, and J. González-Sierra, "Dynamic Modeling and Backstepping Control of a Wheeled Mobile Robot With Skidding and Slipping effects," *Memorias Del Congreso Nacional De Control Automático*, vol. 5, no. 1, pp. 250-255, 2022, https://doi.org/10.58571/CNCA.AMCA.2022.060.

[46] A. H. M, M. M. Fateh, and S. M. Ahmadi, "A finite-time adaptive Taylor series tracking control of electrically-driven wheeled mobile robots," *IET Control Theory and Applications*, vol. 16, no. 10, pp. 1042-1061, 2022, https://doi.org/10.1049/cth2.12284.

[47] N. Hassan and A. Saleem, "Neural Network-Based Adaptive Controller for Trajectory Tracking of Wheeled Mobile Robots," *IEEE Access*, vol. 10, pp. 13582-13597, 2022, https://doi.org/10.1109/ACCESS.2022.3146970.

[48] A. M. Abed *et al.*, "Trajectory tracking of differential drive mobile robots using fractional-order proportional-integral-derivative controller design tuned by an enhanced fruit fly optimization," *Measurement and Control*, vol. 55, no. 3-4, pp. 209-226, 2022, https://doi.org/10.1177/00202940221092134.

[49] L. Xu, J. Du, B. Song, and M. Cao, "A combined backstepping and fractional-order PID controller to trajectory tracking of mobile robots," *Systems Science & Control Engineering*, vol. 10, no. 1, pp. 134-141, 2022, https://doi.org/10.1080/21642583.2022.2047125.

[50] N. H. Thai, T. T. K. Ly, H. Thien, and L. Q. Dzung, "Trajectory Tracking control for Differential-Drive Mobile Robot by a Variable Parameter PID Controller," *International Journal of Mechanical Engineering and Robotics Research*, vol. 11, no. 8, pp. 614-621, 2022, https://doi.org/10.18178/ijmerr.11.8.614-621.

[51] A. Ferraro and V. Scordamaglia, "A set-based approach for detecting faults of a remotely controlled robotic vehicle during a trajectory tracking maneuver," *Control Engineering Practice*, vol. 139, p. 105655, 2023, https://doi.org/10.1016/j.conengprac.2023.105655.

[52] A.-M. D. Tran, T.-V. Vu, and Q.-D. Nguyen, "A study on general state model of differential drive wheeled mobile robots," *Journal of Advanced Engineering and Computation*, vol. 7, no. 3, p. 174, 2023, http://dx.doi.org/10.55579/jaec.202373.417.

[53] T.-V. Vu, A.-M. D. Tran, B.-H. Nguyen, and H.-V. V. Tran, "Development of decentralized speed controllers for a differential drive wheel mobile robot," *Journal of Advanced Engineering and Computation*, vol. 7, no. 2, p. 76, 2023, http://dx.doi.org/10.55579/jaec.202372.399.

[54] J. A. Báez-Hernández, M. Velasco-Villa and S. Mondié, "Non-Linear Prediction-Based Trajectory Tracking for Non-Holonomic Mobile Robots," *IEEE Access*, vol. 11, pp. 124265-124277, 2023, https://doi.org/10.1109/ACCESS.2023.3330145.

[55] S. YiGiT and A. SezgiN, "Trajectory Tracking via Backstepping Controller with PID or SMC for Mobile Robots," *Sakarya University Journal of Science*, vol. 27, no. 1, pp. 120-134, 2023, https://doi.org/10.16984/saufenbilder.1148158.

[56] A. A. Tilahun, T. W. Desta, A. O. Salau, and L. Negash, "Design of an Adaptive Fuzzy Sliding Mode Control with Neuro-Fuzzy system for control of a differential drive wheeled mobile robot," *Cogent Engineering*, vol. 10, no. 2, 2023, https://doi.org/10.1080/23311916.2023.2276517.

[57] K. P. M, K. M, N. K. K, A. K. Pinagapani and K. Ramkumar, "Trajectory Tracking Control of a Differential-Drive Mobile Robot," *2025 International Conference on Computational Innovations and Engineering Sustainability (ICCIES)*, pp. 1-7, 2025, https://doi.org/10.1109/ICCIES63851.2025.11032748.

[58] L. F. Pugliese, K. O. Santos, T. G. De Oliveira, and J. A. Monte-Mor, "A Framework for the Parameterization of Robust Stabilizing $H\infty$ Controllers Applied in Trajectory Tracking of Non-holonomic Robots," *Journal of Control Automation and Electrical Systems*, vol. 35, no. 2, pp. 301-313, 2024, https://doi.org/10.1007/s40313-024-01078-w.

[59] T. N. T. Cao, B. T. Pham, H. D. Tran, L. X. Gia, N. T. Nguyen, and V. N. Truong, "Non-singular terminal sliding mode control for trajectory-tracking of a differential drive robot," *E3S Web of Conferences*, vol. 496, p. 02005, 2024, https://doi.org/10.1051/e3sconf/202449602005.

[60] T. T. K. Ly, Ng. H. Thai, and L. T. Phong, "Design of neural network-PID controller for trajectory tracking of differential drive mobile robot," *Vietnam Journal of Science and Technology*, vol. 62, no. 2, pp. 374-386, 2024, https://doi.org/10.15625/2525-2518/18066.

[61] K. P. Kochumon, L. P. P. S and H. K. R, "Self-Tuning Backstepping and Sliding Mode Control for Robust Trajectory Tracking in Differential Drive Wheeled Mobile Robots," *2023 International Conference on Power, Instrumentation, Control and Computing (PICC)*, pp. 1-6, 2023, https://doi.org/10.1109/PICC57976.2023.10142870.

[62] P.-H. Huynh *et al.*, "Model Predictive Control for rotary Inverted pendulum: Simulation and experiment," *Journal of Fuzzy Systems and Control*, vol. 2, no. 3, pp. 215-222, 2024, https://doi.org/10.59247/jfsc.v2i3.263.

[63] P. H. Huynh, *et al.*, "A Study of Adaptive Model Predictive Control for Rotary Inverted Pendulum," *Journal of Fuzzy Systems and Control*, vol. 3, no 2, p. 98-103, 2025, https://doi.org/10.59247/jfsc.v3i2.302.