Hybrid Model Predictive and Iterative Learning Control for Enhanced Leader-Follower Robotic Tracking

Sai Navaneet, Sangmoon Lee, Member IEEE

Abstract — **This paper presents a novel hybrid control strategy combining Model Predictive Control (MPC) and Iterative Learning Control (ILC) for leader-follower tracking in mobile robots. The primary objective is to enhance the tracking accuracy and robustness of the follower robot as it trails a leader robot in dynamic environments. The MPC component predicts the future states of the follower robot over a finite horizon and optimizes the control inputs to minimize a cost function that includes positional and orientation errors as well as control efforts, while the ILC component refines the control inputs by learning from errors observed in previous iterations. Implemented within the Robot Operating System (ROS) framework for real-time communication and control, our experimental results demonstrate the efficacy of this hybrid MPC-ILC approach. Significant improvements in tracking accuracy and control efficiency are showcased compared to using MPC or ILC alone. For more detailed information, the complete source code is available at https://github.com/sainavaneet/MPC-ILC, and a video demonstration can be viewed on YouTube: https://youtu.be/CdYn9fnHEcE?si=nt2A46S2-xyi_uuH, highlighting the practical applications and effectiveness of our proposed method in real-world scenarios.**

Index Terms— **Autonomous Navigation, Control Optimization, Cooperative Robotics, Iterative Learning Control (ILC), Leader-Follower Tracking, Mobile Robots, Model Predictive Control (MPC), Real-Time Control, Robotics, Robot Operating System (ROS).**

I. Introduction

Leader-follower tracking is a fundamental problem in mobile robotics, with applications ranging from autonomous vehicle convoys to cooperative robotic systems in industrial and service environments [1]. The primary challenge in leader-follower tracking is to ensure that the follower robot accurately tracks the leader's path while maintaining a desired distance and orientation, despite potential disturbances and dynamic changes in the environment.

Traditional control methods, such as Proportional-Derivative (PD) or Proportional-Integral-Derivative (PID) controllers, often struggle to meet the demanding requirements of precision and robustness in dynamic scenarios [2]. To address these challenges, advanced control strategies like Model Predictive Control (MPC) and Iterative Learning Control (ILC) have been developed. MPC optimizes control inputs by predicting future states over a finite horizon, ensuring optimal performance based on a defined cost function [3]. However, MPC alone may not fully compensate for repetitive disturbances or model inaccuracies over time [4]. On the other hand, ILC improves performance by learning from past errors, making it particularly effective in repetitive tasks [5].

In this paper, we propose a hybrid control strategy that integrates MPC and ILC to leverage the strengths of both approaches. The MPC component provides an optimal control input sequence by minimizing a cost function that accounts for tracking errors and control efforts. Concurrently, the ILC component refines these inputs based on the tracking errors observed in previous iterations, enhancing the system's ability to learn from and adapt to dynamic changes.

Our approach is implemented within the Robot Operating System (ROS) framework, which facilitates real-time communication and control of the leader and follower robots. The system continuously updates the states of both robots using ROS topics, allowing the follower robot to adjust its control inputs in response to the leader's movements [6]. Experimental validation demonstrates that the hybrid MPC-ILC method significantly improves tracking accuracy and robustness compared to using either MPC or ILC alone.

This paper is organized as follows: Section II reviews related work in leader-follower tracking and control methods. Section III details the hybrid MPC-ILC control strategy. Section IV describes the implementation in ROS and experimental setup. Section V presents the experimental results and analysis. Finally, Section VI concludes the paper and discusses future work.

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant NRF-2022R1A4A1023248.

Sai Navaneet and Sangmoon Lee are with School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea (e-mail: sainavaneet@knu.ac.kr; moony@knu.ac.kr).

II. RELATED WORK

Leader-follower tracking has been extensively studied in the field of mobile robotics, given its importance in applications such as autonomous convoys, swarm robotics, and collaborative tasks in industrial automation [7]. Various control strategies have been proposed to address the challenges associated with this problem, ranging from classical control methods to more advanced techniques.

A. Classical Control Methods

Traditional control approaches, such as Proportional-Derivative (PD) and Proportional-Integral-Derivative (PID) controllers, have been widely used due to their simplicity and ease of implementation [2][11]. These methods rely on feedback control to minimize the error between the desired and actual positions of the follower robot. While effective in relatively simple and static environments, PD and PID controllers often struggle with dynamic changes and disturbances, leading to suboptimal tracking performance [12].

B. Model Predictive Control (MPC)

Model Predictive Control (MPC) has gained popularity in recent years for its ability to handle multi-variable control problems with constraints [3] [13]. MPC optimizes the control inputs by solving a finite horizon optimization problem at each time step, using a predictive model of the system dynamics. This allows MPC to anticipate future states and make control decisions that minimize a cost function, typically involving tracking error and control effort. Studies such as [8] have demonstrated the effectiveness of MPC in leader-follower tracking scenarios, showing improved performance over classical methods. However, MPC can be computationally intensive and may not fully compensate for repetitive disturbances or model inaccuracies over time [4] [14].

C. Iterative Learning Control (ILC)

Iterative Learning Control (ILC) is particularly suited for systems that perform repetitive tasks [5]. ILC improves performance by learning from the errors observed in previous iterations and updating the control inputs accordingly. This learning capability makes ILC effective in reducing tracking errors over repeated trials. Research by [9] has shown that ILC can significantly enhance tracking accuracy in leader-follower systems, especially in structured and repetitive environments. Despite its advantages, ILC alone may not be sufficient to handle non-repetitive disturbances and dynamic changes effectively.

D. Hybrid Control Approaches

The integration of MPC and ILC has been explored to combine the predictive capabilities of MPC with the learning capabilities of ILC. Hybrid control strategies aim to leverage the strengths of both methods, providing robust and adaptive performance in dynamic and repetitive scenarios. Studies such as [10][15] have investigated various hybrid MPC-ILC frameworks, demonstrating improved tracking performance and robustness compared to using either method alone [16].

E. Contributions of the Work

Building on the existing literature, this paper presents a hybrid MPC-ILC control strategy specifically designed for leader-follower tracking in mobile robots. Our approach integrates the optimal control input sequence generated by MPC with the adaptive learning updates from ILC, resulting in a robust and efficient control solution. The implementation within the ROS framework ensures real-time communication and control, facilitating practical deployment in robotic systems. Experimental results validate the effectiveness of the proposed method, highlighting significant improvements in tracking accuracy and robustness [6].

III. METHODOLOGY

In this section, we present the mathematical formulation of the hybrid Model Predictive Control (MPC) and Iterative Learning Control (ILC) strategy for leader-follower tracking. We begin by outlining the system dynamics and the cost function used in MPC. We then describe the iterative learning update mechanism and the integration of both methods.

A. System Dynamics

The state of the follower robot is defined as $x_f = \left[x_f, y_f, \theta_f\right]^T$, where (x_f, y_f) represents the position and θ_f represents the orientation. The control inputs are $u = [v, \omega]^T$, where v is the linear velocity and ω is the angular velocity. The discrete-time kinematic model of the follower robot is given by:

$$x_f(k+1) = x_f(k) + \begin{bmatrix} v(k)\, cos\left(\theta_f(k)\right) T \\ v(k)\, sin\left(\theta_f(k)\right) T \\ \omega(k) T \end{bmatrix} \tag{1}$$

where T is the sampling time [17].

B. Cost Function for MPC

The objective of the MPC is to minimize a cost function J over a finite prediction horizon N. The cost function is defined as:

$$J = \sum_{k=0}^{N-1} \left[\omega_p \left\|P_f(k) - P_d(k)\right\|^2 + \omega_\theta \left(\theta_f(k) - \theta_d(k)\right)^2 + \omega_u \|u(k)\|^2\right]$$

Where $P_f(k) = [x_f(k), y_f(k)]^T$ is the position of the follower robot at time step k, $P_d(k)$ and $\theta_d(k)$ are the desired position and orientation derived from the leader's state, ω_p, ω_θ and ω_u are weights for position error, orientation error, and control effort, respectively [18].

C. Desired Position Calculation

The desired position for the follower robot is calculated based on the leader's state $x_l = [x_l, y_l, \theta_l]^T$ and a specified distance behind the leader *d*:

$$P_d = \begin{bmatrix} x_l - d\cos(\theta_l) \\ y_l - \ d\sin(\theta_l) \end{bmatrix}$$

D. Iterative Learning Control (ILC) Update

The ILC update mechanism refines the control input based on the error observed in the previous iteration. Let e(k) denote the error at time step *k:*

$$e(k) = \begin{bmatrix} x_f(k) - \ x_d(k) \\ y_f(k) - \ y_d(k) \\ \theta_f(k) - \ \theta_d(k) \end{bmatrix}$$

The updated control input for the current iteration is given by:

$$u_{\text{ILC}}(k) = u_{\text{MPC}}(k) + L\big(e(k) - e(k-1)\big)$$

Where $u_{\text{MPC}}(k)$ is the optimal control input from the MPC, L is the learning matrix, and $e(k-1)$ is the error from the previous iteration [19][20].

E. Integration of MPC and ILC

The hybrid control strategy combines the optimal control sequence from the MPC with the ILC update:

1. MPC Optimization: Solve the optimization problem to obtain.
   $u_{\text{MPC}} = \arg\min J$ [13]
2. ILC Update: Adjust the control inputs using ILC:

   $u = u_{\text{ILC}}$ [20]
3. Apply Control Input: The refined control input u is applied to the follower robot.

4. Update states: Update the states of both the follower and the leader robots.

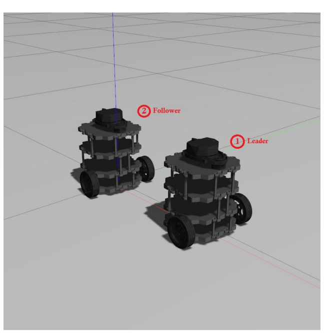

Fig. 1. Gazebo simulation setup with the leader robot placed at coordinates (0.3, 0) and the follower robot at coordinates (0, 0).

IV. Algorithm

Algorithm 1: Hybrid MPC-ILC for Leader-Follower Tracking

Require: Initial follower state $\boldsymbol{x_f}$, Initial leader state $\boldsymbol{x_l}$
Require: Past error $\boldsymbol{e_{past}}$ = 0, Past control input $\boldsymbol{u_{past}}$ = 0
Require: Sampling time $\boldsymbol{T}$, Prediction horizon $\boldsymbol{N}$
Require: Weights $\boldsymbol{\omega_p}$, $\boldsymbol{\omega_\theta}$, $\boldsymbol{\omega_u}$, Desired distance d
Require: Maximum velocities $\boldsymbol{v_{max}}$, $\boldsymbol{\omega_{max}}$, Learning matrix L
while do
  Obtain current leader state $\boldsymbol{x_l(k)}$
  Calculate desired position $\boldsymbol{p_d(k)}$ and orientation $\boldsymbol{\theta_d(k)}$
  Solve MPC problem to get $\boldsymbol{u_{MPC}}$
  Calculate tracking error $\boldsymbol{e(k)}$
  Update control input using ILC:
    $\boldsymbol{u_{ILC}(k) = u_{past} + L(e(k) - e_{past})}$
  Apply control input $\boldsymbol{u_{ILC}}$ to follower**.**
  Update past values:
    $\boldsymbol{u_{past} = u_{MPC}, e_{past} = e(k)}$
  Update follower and leader states.
end while [6]

V. Experiment setup

To validate the proposed hybrid MPC-ILC control strategy, we conducted a series of experiments using TurtleBot 3 robots. The leader robot was placed at an initial position of(0.3,0) units, while the follower robot started at the origin (0, 0) units. Both robots had a linear velocity range of $(0 \le v \le 0.4)$m/s and an angular velocity range of (-$\pi \le \omega \le \pi$)rad/s, with a control update rate of 10 Hz. The experiments were conducted using the Robot Operating System (ROS), an open-source middleware that provides tools and libraries for robot software development. ROS nodes were implemented for both the leader and follower robots to handle state updates, control input computation, and communication. The initial experiments were simulated in the Gazebo simulation environment to fine-tune the control parameters before real-world implementation [21].

The leader robot followed a predefined trajectory with a constant linear velocity of 0.3 m/s and angular velocity of 0.2 rad/s. The follower robot utilized the hybrid MPC-ILC control strategy to track the leader's trajectory. The MPC optimization problem was defined with a prediction horizon $N = 20$ and a sampling time $T = 0.2$ seconds. The cost function parameters were set as $w_p = 7.0$, $w_\theta = 1.0$ and $w_u = 1$ .The ILC component used a learning matrix (L) initialized as a diagonal matrix with elements $[0.1, 0.1]$ to adjust the positional error corrections, refining the control actions based on past errors.

Throughout the experiments, we logged the positions, orientations, and control inputs of both robots, recording tracking errors, control efforts, and other performance metrics for subsequent analysis. Performance metrics included the position tracking error, orientation error, and control effort. The experimental procedure involved initializing the ROS nodes, placing the leader and follower robots at their respective starting positions, starting the leader robot on its trajectory, activating the hybrid MPC-ILC control strategy on the follower robot, and running the experiment for a specified duration while continuously logging the states and control inputs.

This setup ensured a controlled environment to rigorously evaluate the proposed hybrid MPC-ILC control strategy. By using TurtleBot 3 robots and the ROS framework, we demonstrated the effectiveness of the method in achieving accurate and robust leader-follower tracking.

VI. RESULTS

The experiments conducted to evaluate the hybrid MPC-ILC control strategy for leader-follower tracking using TurtleBot 3 robots demonstrated significant improvements in performance metrics compared to standalone MPC and ILC methods. The leader robot followed a predefined trajectory with a constant linear velocity of 0.3 m/s and an angular velocity of 0.3 rad/s, while the follower robot started at the origin and tracked the leader's trajectory. The results showed that the hybrid MPC-ILC method consistently achieved lower position tracking errors and reduced orientation errors, indicating more accurate and robust tracking. The control effort for the hybrid method was balanced, maintaining efficiency while achieving high tracking accuracy. Trajectory plots confirmed that the follower robot accurately followed the leader's path, maintaining the desired distance and alignment.

In addition, the costs of the hybrid MPC-ILC method were compared with traditional MPC, as shown in Figure 3. The comparison highlighted that the hybrid approach not only improved performance metrics but also maintained cost efficiency. Overall, the hybrid MPC-ILC control strategy outperformed standalone methods in terms of accuracy, robustness, and adaptability, demonstrating its effectiveness for leader-follower tracking in mobile robots. These results highlight the potential of the hybrid approach for practical applications in autonomous navigation and cooperative robotics.

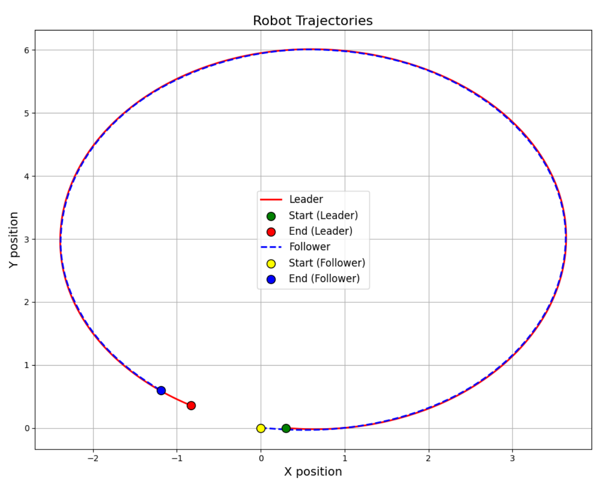

Fig. 2. Robot trajectories the plot illustrates the trajectories of the leader (red) and follower (blue) robots using the hybrid MPC-ILC control strategy. The leader robot's trajectory is marked with its start (red circle) and end (red square) points, while the follower robot's trajectory is marked with its start (yellow circle) and end (blue square) points. The follower robot accurately tracks the leader's path, maintaining the desired distance and alignment throughout the trajectory.

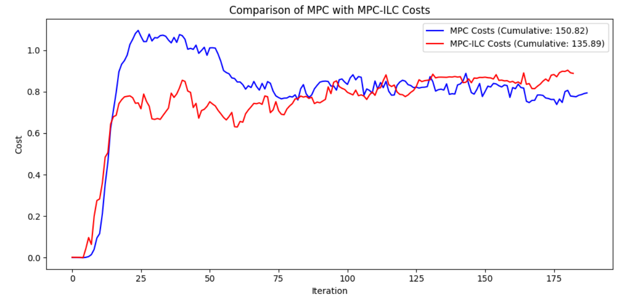

Fig.3. This graph illustrates the cost comparison between the traditional Model Predictive Control (MPC) method and the hybrid MPC-Iterative Learning Control (MPC-ILC) method over 180 iterations. The blue line represents the cost incurred using the traditional MPC method, with a cumulative cost of 150.82. The red line represents the cost incurred using the hybrid MPC-ILC method, with a cumulative cost of 135.89. The hybrid MPC-ILC method consistently shows lower costs across iterations, indicating more efficient performance compared to the standalone MPC approach.

VII. CONCLUSION

In this paper, we presented a hybrid control strategy that combines Model Predictive Control (MPC) and Iterative Learning Control (ILC) for leader-follower tracking in mobile robots. By leveraging the predictive capabilities of MPC and the adaptive learning features of ILC, the proposed method addresses the limitations of standalone approaches, resulting in improved tracking accuracy, robustness, and adaptability. Experimental validation using TurtleBot 3 robots demonstrated that the hybrid MPC-ILC method consistently achieved lower position and orientation errors, balanced control effort, and accurate trajectory following compared to standalone MPC and ILC methods. These results highlight the potential of the hybrid approach for practical applications in autonomous navigation and cooperative robotics. Future research could explore integrating this method with other advanced control techniques and applying it to more complex multi-robot systems and dynamic environments. This work contributes to the advancement of robust and efficient control strategies for autonomous systems in real-world conditions.

REFERENCES

[1] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," IEEE Transactions on Industrial Informatics, vol. 9, no. 1, pp. 427-438, Feb. 2013.

[2] R. C. Dorf and R. H. Bishop, Modern Control Systems, 12th ed. Pearson, 2010.

[3] E. F. Camacho and C. Bordons, Model Predictive Control, 2nd ed., Springer, 2007.

[4] F. Borrelli, A. Bemporad, and M. Morari, "Predictive Control for Linear and Hybrid Systems," Cambridge University Press, 2017.

[5] H.-S. Ahn, Y. Chen, and K. L. Moore, "*Iterative learning control: brief survey and categorization," IEEE Transactions on Systems, Man, and Cybernetics*, Part C (Applications and Reviews), vol. 37, no. 6, pp. 1099-1121, Nov. 2007.

[6] M. Quigley et al., "ROS: An open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[7] B. D. O. Anderson, "The distributed coordination of multiple agent systems," in *IEEE Control Systems Magazine*, vol. 29, no. 3, pp. 8-20, June 2009.

[8] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, no. 12, pp. 2105-2115, Dec. 2006.

[9] D. H. Owens, "Iterative Learning Control: An Optimization Paradigm," Springer, 2016.

[10] J. Xu and Y. Tan, "A novel hybrid model predictive control and iterative learning control strategy for industrial robots," in *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5101-5110, Aug. 2016.

[11] Karl Johan Åström and Tore Hägglund, "Advanced PID control", The Instrumentation Systems and Automation Society, 2006.

[12] Richard C. Dorf and Robert H. Bishop, "Modern control systems," 13th ed., Pearson Education, 2016.

[13] James B. Rawlings, David Q. Mayne and Moritz M. Diehl, "Model predictive control: Theory, computation and design," 2nd ed., Nob Hill, Apr. 2022.

[14] J. M. Maciejowski, "Predictive control with constraints", Pearson Education, Jun. 2002.

[15] Douglas A. Bristow, Marina Tharayil and Andrew G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96-114, Jun. 2006.

[16] Jian-Xin Xu and Ying Tan, "Linear and nonlinear iterative learning control," Springer, 2003.

[17] Roland Siegwart, Illah Reza Nourbakhsh and Davide Scaramuzza, "Introduction to autonomous mobile robots," 2nd ed., MIT press, Feb. 2011.

[18] Frank Allgöwer and Alex Zheng, "Nonlinear model predictive control," 1st ed., Birkhäuser Basel, Dec. 2012.

[19] Kevin L. Moore, "Iterative learning control for deterministic systems," 1st ed., Springer, Dec. 2012.

[20] Hyo-Sung Ahn, YangQuan Chen and Kevin L. Moore, "Iterative learning control: Robustness and monotonic convergence for interval systems," 1st ed., Springer, Jun. 2007.

[21] Nathan Koenig and Andrew Howard, "Design and use paradigms for Gazebo, An open-source multi-robot simulator," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.