Proceedings of the 2025 SICE Festival with Annual Conference
September 9-12, 2025, Chiang Mai, Thailand

Drone Path Planning through Deep Learning-based Real-time Image Recognition and Path Integral Control for Multiple Obstacles

Riku Nagano[1†] and Masaaki Nagahara[2]

[1]NAPIRE inc. [2]Hiroshima University, Japan
(E-mails: n.riku0325@gmail.com, nagahara@ieee.org)

Abstract: This paper introduces a drone obstacle avoidance method employing Model Predictive Path Integral (MPPI) control integrated with real-time deep learning-based image recognition. Specifically, we utilize the YOLOv8 model to perform real-time obstacle detection and integrate this capability with MPPI control for efficient obstacle avoidance in dynamic conditions. The proposed framework is implemented using Python and validated through experiments conducted with a DJI Tello drone. Experimental results confirm the effectiveness and robustness of the proposed method.

Keywords: Model Predictive Path Integral Control, Drone, Obstacle Avoidance, Deep Learning, YOLOv8, DJI Tello

1. INTRODUCTION

In recent years, autonomous drone flight technology has been applied to practical systems in various fields such as logistics, surveillance, and disaster response [1]. In particular, obstacle avoidance in complex environments remains a significant challenge in drone applications. Traditional obstacle avoidance methods typically rely on static obstacle detection using sensors such as LiDAR or ultrasonic devices, or require prior construction of environmental maps [2]. Nevertheless, these methods face difficulties adapting to dynamic environments and present issues concerning real-time processing and cost.

On the other hand, recent advances in deep learning have enabled real-time object detection directly from camera images. In particular, the YOLO (You Only Look Once) [3] family of models excels in high-speed and accurate object detection, which is particularly suitable for real-time applications on embedded devices. The latest YOLOv8 [4] model has achieved notable improvements in both compactness and accuracy and has been implemented on drone platforms.

In this study, we propose a drone control method enabling dynamic obstacle avoidance by combining real-time image recognition via deep learning with Model Predictive Path Integral (MPPI) control. MPPI is effective in addressing nonlinear and nonconvex control problems, and by utilizing stochastic sampling, it can achieve flexible real-time control [5, 6].

Additionally, the integration of MPPI with the parallel computational capabilities of advanced GPUs has recently advanced its applications to agile UAVs [7-9]. This research compiles methods and experimental results of using YOLOv8 as an object detection model to identify obstacles from camera images in real time, with MPPI control. To demonstrate the effectiveness of the proposed method, we show experimental results using the DJI Tello drone platform on which the proposed algorithm is implemented with Python.

† Riku Nagano is the presenter of this paper.

2. MODEL PREDICTIVE PATH INTEGRAL CONTROL (MPPI)

2.1. MPPI Overview

Model Predictive Path Integral (MPPI) control is a type of model predictive control utilizing stochastic sampling methods, particularly suitable for nonlinear systems and problems with nonconvex cost functions [5]. In MPPI, multiple samples are generated by introducing noise into future control input sequences, and the corresponding trajectories are evaluated according to their costs. Samples with lower costs receive higher weights, and the next control input is determined by computing a weighted average. The computation is simple and fast, and hence MPPI is also suitable for real-time control applications such as obstacle avoidance.

2.2. Mathematical Model of MPPI

Let $T \in \mathbb{N}$ be the predictive horizon length in the model predictive control. Define the control input sequence by $\mathbf{U} = \{\mathbf{u}_t\}_{t=0}^{T-1}$ and the system state sequence by $\{\mathbf{x}_t\}_{t=0}^{T}$, which satisfy the following system equation:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t), \quad t = 0, 1, \ldots, T-1. \tag{1}$$

For this system, the MPPI solves the following stochastic optimal control problem:

$$\mathbf{U}^* = \arg\min_{\mathbf{U}} \mathbb{E}_{\mathbf{V}\sim q}\left[\Phi(\mathbf{x}_T) + \sum_{t=0}^{T-1} L(\mathbf{x}_t, \mathbf{v}_t)\right], \tag{2}$$

where $\Phi(\mathbf{x}_T)$ is the terminal cost, $L(\mathbf{x}_t, \mathbf{v}_t)$ is the stage cost with control input sample $\mathbf{V} = \{\mathbf{v}_t\}_{t=0}^{T-1}$, and q is the probability distribution defined by

$$q = q(\mathbf{V}|\mathbf{U}, \Sigma) = \prod_{t=0}^{T-1} \mathcal{N}(\mathbf{v}_t|\mathbf{u}_t, \Sigma). \tag{3}$$

The algorithm to compute the optimal control input is given as follows:

1. Generate random control input samples $\{\delta\mathbf{u}_t^{(k)}\}$ (for $k = 1, \ldots, K$, where K is the number of samples).

2. Simulate system dynamics (1) for each sample to predict future states $\{\mathbf{x}_t^{(k)}\}$.
3. Calculate the cost for each sample and predicted state.
4. Compute weights based on these costs, then determine the optimal control input through a weighted average.

The details of this algorithm are discussed in the next section.

3. PROPOSED METHOD

3.1. System Configuration

The proposed approach analyzes images captured by the drone-mounted camera using YOLOv8 to estimate obstacle positions. These estimated positions are then utilized to select the optimal control inputs from multiple trajectories generated by MPPI, ensuring minimal cost while effectively avoiding obstacles.

3.2. Implementation of MPPI control algorithm

The control input to the drone is limited to the lateral acceleration $a_{y,t}$, and the MPPI control algorithm is implemented as follows:
1. Initialize the initial control input sequence $\{\mathbf{u}_t\}_{t=0}^{T-1}$ with zeros.
2. Generate samples of control inputs $\delta\mathbf{u}_t^{(k)}$ for each time step t, where $\delta\mathbf{u}_t^{(k)} \sim \mathcal{N}(0, \sigma^2)$.
3. For each sample, predict future states and compute the following cost function:

$$S^{(k)} = \Phi(\mathbf{x}_T^{(k)}) + \sum_{t=0}^{T-1} L(\mathbf{x}_t^{(k)}, \mathbf{u}_t^{(k)}). \tag{4}$$

4. Compute weights based on the costs:

$$w^{(k)} = \frac{\exp\left(-\frac{1}{\lambda}S^{(k)}\right)}{\sum_{i=1}^{K}\exp\left(-\frac{1}{\lambda}S^{(i)}\right)}. \tag{5}$$

5. Update the optimal control input by calculating the weighted average:

$$\mathbf{u}_t \leftarrow \mathbf{u}_t + \sum_{k=1}^{K} w^{(k)}\delta\mathbf{u}_t^{(k)}. \tag{6}$$

Note that when the prediction horizon T is short, the MPPI algorithm may predominantly generate trajectories favoring only one direction (either left or right) for obstacle avoidance. Consequently, the resulting averaged control input might be biased toward a single side, potentially reducing flexibility in trajectory planning.

3.3. Cost Function Design

The cost function comprises the following terms:
- **Position cost**: Squared distance between drone position and the target position

$$L_{\text{pos}}(\mathbf{x}_t) = c_{\text{pos}}\left((x_t - x_{\text{goal}})^2 + (y_t - y_{\text{goal}})^2\right). \tag{7}$$

- **Velocity cost**: Squared lateral velocity

$$L_{\text{vel}}(\mathbf{x}_t) = c_{\text{vel}} v_{y,t}^2. \tag{8}$$

- **Obstacle cost**: Applies a significant penalty when the distance to an obstacle falls below a specified threshold

$$L_{\text{obs}}(\mathbf{x}_t) = \begin{cases} c_{\text{obs}}\left(\frac{1}{d_t} - \frac{1}{d_{\text{safe}}}\right), & \text{if } d_t < d_{\text{safe}}, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

where d_t denotes the drone-obstacle distance, d_{safe} is the safety distance, and c_{obs} is the coefficient for obstacle cost.

The final cost function combines these terms as follows:

$$L(\mathbf{x}_t, \mathbf{u}_t) = L_{\text{pos}}(\mathbf{x}_t) + L_{\text{vel}}(\mathbf{x}_t) + L_{\text{obs}}(\mathbf{x}_t). \tag{10}$$

4. EXPERIMENTAL RESULTS

4.1. experimental setup

For the experiments, a DJI Tello drone [13] was utilized, controlled by an algorithm implemented in Python. The drone is equipped with a 720p camera capable of transmitting real-time video via Wi-Fi. The drone starts from an initial position $(x_0, y_0) = (0.0, 0.0)$ and moves forward towards the target position $(x_{\text{goal}}, y_{\text{goal}}) = (2.5, 0.0)$.

Obstacles, including persons, were identified from the video feed using YOLOv8. The object detection employed Ultralytics' implementation [4], with inference executed on a laptop PC.

Note that the drone itself does not directly execute Python code. Instead, the MPPI-based control input signals are generated by a host computer and transmitted to the DJI Tello drone via a Wi-Fi connection in real time.

As demonstrated by recent studies on MPPI [7, 9], this setup is confirmed to achieve sufficient real-time performance even on lightweight drones, such as the one used in this study.

4.2. control parameters

The control parameters used in the experiment are shown below.
- Time step: $\Delta t = 0.1$ [s]
- Number of samples: $K = 200$
- Horizon length: $T = 20$
- Noise standard deviation: $\sigma = 0.1$
- Lambda parameter: $\lambda = 1.0$
- Forward velocity: $v_{\text{forward}} = 0.2$ [m/s]
- Speed limit: $v_{\max} = 0.5$ [m/s]
- Acceleration limit: $a_{\max} = 1.0$ [m/s^2]
- Cost factor: $c_{\text{pos}} = 10.0$, $c_{\text{vel}} = 1.0$, $c_{\text{obs}} = 1000$
- safe distance: $d_{\text{safe}} = r_{\text{obs}} + 0.05$ [m]

subsectionexperimental video

Video of the experiment is available at the following link.
- Experimental video 1:
  - `https://youtu.be/QoqpZli2WtQ`
- Experimental video 2:
  - `https://youtu.be/oyacQAL9ohQ`

In the video, the drone can be seen navigating towards its target position while accurately identifying and avoiding obstacles an umbrella and a chair captured by its camera. The experimental results demonstrate that, even in

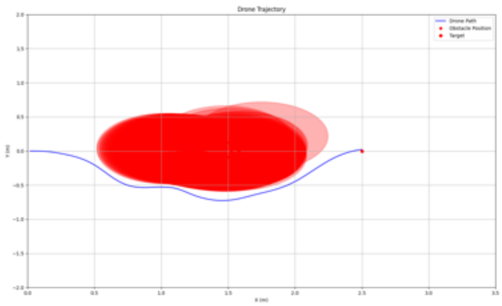

Fig. 1 plot results of experimental video 1

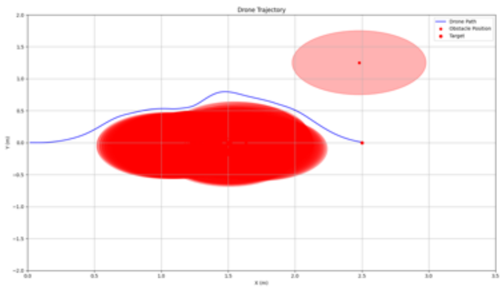

Fig. 2 plotting results of experimental video 2

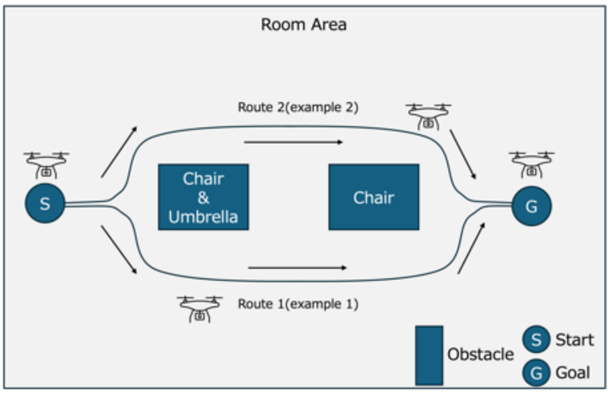

Fig. 3 Obstacle Area Map

the presence of obstacles, YOLOv8 successfully detects obstacle positions from the camera images, enabling the MPPI controller to promptly calculate control inputs for obstacle avoidance. Furthermore, as an extension, incorporating SE(3) theory [10] or differential flatness [12] could enhance agile attitude control and multi-obstacle handling.

4.3. results and discussion

The results illustrated in experimental videos 1 and 2, as well as their corresponding explanatory images, confirm that obstacle avoidance using the YOLOv8 combined with MPPI operates effectively and safely in real time. Thus, even when obstacles (an umbrella and a chair) are detected, the drone can adaptively perform evasive maneuvers while continuing to approach the target location. Although the YOLOv8 detection system operates at high speed, enabling potential real-time responses to dynamic obstacles, further experimental validation in

Fig. 4 Drone Route Pattern

dynamic environments is left for future work.

However, since the DJI Tello drone used in this experiment only features a forward-facing camera, obstacles that exit the camera frame may lead to incorrect assessments, resulting in collisions. As future improvements, it is crucial to implement mechanisms that either estimate and avoid obstacle positions after they exit the camera's view or integrate sensors capable of detecting obstacles in all directions.

5. CONCLUSION

In this study, we achieved obstacle avoidance control for drones by integrating real-time deep learning-based image recognition with Model Predictive Path Integral (MPPI) control. Utilizing a readily available DJI Tello drone, we implemented an algorithm in Python capable of performing real-time obstacle detection and avoidance.

Future research directions include further integration with information-theoretic approaches [8], exploring hybrid control methods combining MPPI and MPC [11], as well as expansions into SE(3)-based control theories [10], differential flatness methods for drones [12], and additional MPC techniques [11].

Key challenges moving forward involve enhancing the accuracy of obstacle detection, increasing robustness through sensor fusion with LiDAR or ultrasonic sensors, and managing more complex environments with multiple obstacles. Moreover, improving the efficiency of the control algorithm is essential to achieve higher operational frequencies.

6. ACKNOWLEDGMENTS

This work was supported in part by JSPS KAKENHI Grant Numbers 23K26130, 22KK0155, 22H00512, 24K21314, and JST as part of ASPIRE Grant Number JPMJAP2402. This research was also supported by piland, Inc. The authors would like to express their gratitude for this support.

REFERENCES

[1] Nonani: Sequel to "All about Drone Industrial Applications" —The Future of Evolving Autonomous Flight—, Ohmsha, 2020.

[2] Nonami: Latest technology trends and prospects in small unmanned aerial vehicles (drones), Measurement and Control, 59.7 pp. 437–443, 2020.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: You Only Look Once: Real-Time Object Detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[4] Ultralytics: YOLOv8, 2023 `github.com/ultralytics/ultralytics`

[5] G. Williams, A. Aldrich, and E. A. Theodorou: Model Predictive Path Integral Control: From Theory to Parallel Computation, *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017.

[6] X. Dai and M. Nagahara, Platooning control of drones with real-time deep learning object detection, *Advanced Robotics*, vol. 37, no. 3, pp. 220–225, 2022.

[7] M. Minarik, R. Penicka, V. Vonasek, and M. Saska: Model Predictive Path Integral Control for Agile Unmanned Aerial Vehicles, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[8] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou: Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving, *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603–1622, 2018.

[9] J. Higgins, N. Mohammad, and N. Bezzo: A Model Predictive Path Integral Method for Fast, Proactive, and Uncertainty-Aware UAV Planning in Cluttered Environments, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[10] T. Lee, M. Leok, and N. H. McClamroch: Nonlinear robust tracking control of a quadrotor UAV on SE(3), *Asian Journal of Control*, vol. 15, no. 2, pp. 391–408, 2013.

[11] H. Nguyen, M. Kamel, K. Alexis, and R. Siegwart: Model predictive control for micro aerial vehicles: A survey, *European Control Conference (ECC)*, pp. 1556–1563, 2021.

[12] D. Mellinger, N. Michael, and V. Kumar: Trajectory generation and control for precise aggressive maneuvers with quadrotors, *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[13] DJI: DJI Tello User Manual, *Technical Documentation*, DJI