

Model Predictive Path Integral Docking of Fully Actuated Surface Vessel

1st Akash Vijayakumar

Department of Ocean Engineering
Indian Institute of Technology Madras
Chennai, India
na21b003@smail.iitm.ac.in

2nd Atmanand M A

Department of Ocean Engineering
Indian Institute of Technology Madras
Chennai, India
atma@ee.iitm.ac.in

3rd Abhilash Somayajula

Department of Ocean Engineering
Indian Institute of Technology Madras
Chennai, India
abhilash@iitm.ac.in

Abstract—Autonomous docking remains one of the most challenging maneuvers in marine robotics, requiring precise control and robust perception in confined spaces. This paper presents a novel approach integrating Model Predictive Path Integral (MPPI) control with real-time LiDAR-based dock detection for autonomous surface vessel docking. Our framework uniquely combines probabilistic trajectory optimization with a multi-objective cost function that simultaneously considers docking precision, safety constraints, and motion efficiency. The MPPI controller generates optimal trajectories by intelligently sampling control sequences and evaluating their costs based on dynamic clearance requirements, orientation alignment, and target position objectives. We introduce an adaptive dock detection pipeline that processes LiDAR point clouds to extract critical geometric features, enabling real-time updates of docking parameters. The proposed method is extensively validated in a physics-based simulation environment that incorporates realistic sensor noise, vessel dynamics, and environmental constraints. Results demonstrate successful docking from various initial positions while maintaining safe clearances and smooth motion characteristics.

Index Terms—MPPI, Docking, ASV.

I. INTRODUCTION

Autonomous docking of marine vessels represents one of the most challenging maneuvers in maritime autonomy, requiring precise control while accounting for complex vessel dynamics, environmental conditions, and safety constraints. This capability is particularly crucial for unmanned surface vessels (USVs), where reliable autonomous docking enables extended operations and reduced human intervention. The challenge lies not only in achieving the final docked position but also in executing a safe and efficient approach trajectory while maintaining proper clearance from dock structures.

Traditional control approaches for autonomous docking often rely on predetermined trajectories or reactive behaviors, which may not optimally handle the full complexity of the docking maneuver. Model Predictive Control (MPC) offers a more sophisticated approach by continuously optimizing future trajectories, but classical MPC implementations can struggle with the nonlinear dynamics and uncertainty inherent in marine environments. This motivates the exploration of sampling-based MPC methods, particularly the Model Predictive Path Integral (MPPI) control framework [1], which can naturally handle nonlinear dynamics and non-convex constraints through probabilistic sampling.

In this paper, we present a comprehensive MPPI-based control framework specifically designed for autonomous vessel docking. Our approach incorporates several key innovations:

- A multi-objective cost function that balances docking precision, safety constraints, and motion efficiency
- Real-time processing of LiDAR data for dock detection and relative positioning for cost computation.
- Computation of costs without access to prior map information of the environment.

The control framework is extensively validated in a physics-based simulation environment that captures essential aspects of marine vessel dynamics and sensor characteristics. Our implementation considers a fully actuated surface vessel equipped with omnidirectional thrust capability, similar to modern autonomous water taxis and small marine research platforms. The simulation incorporates realistic sensor noise and precise collision detection to evaluate the robustness of the control strategy.

Results demonstrate the effectiveness of our approach in achieving precise docking maneuvers while maintaining safe clearances and smooth motion characteristics. The controller successfully handles various initial positions and orientations, demonstrating its potential for practical deployment in autonomous marine systems. Quantitative analysis of docking accuracy, path efficiency, and motion smoothness provides insights into the controller's performance characteristics.

II. REFERENCES AND RELATED WORK

The task of autonomous vessel docking has been approached through various methodologies in recent literature. Here, we discuss key related works and highlight how our approach differs.

Homburger et al. [2] applied MPPI control to a fully-actuated surface vessel, achieving precise docking maneuvers through a cost function that accounts for docking position, orientation, and actuator dynamics. While similar in using MPPI, our work extends this by incorporating lidar scan instead of a prior map of environment and cost formulation for practical docking scenarios.

Streichenberg et al. [3] proposed a decentralized MPPI framework for multi-agent path planning in urban canals, focusing on interaction-aware motion planning. While they

demonstrated success in navigation through narrow canals, their work did not specifically address the docking problem that we tackle.

Patel et al. [4] presented an MPPI framework for USV control under wave disturbances using Froude-Krylov theory. While they showed robust control under environmental disturbances, their work focused on trajectory tracking rather than the specific challenges of docking maneuvers that we address.

The key contribution of our work lies in bridging the gap between theoretical MPPI formulations and practical autonomous docking implementations by incorporating real-world considerations while maintaining computational feasibility. Future work could address the limitations through adaptive cost weighting schemes and improved environmental disturbance modeling.

III. MODEL PREDICTIVE PATH INTEGRAL (MPPI) ALGORITHM

Model Predictive Path Integral (MPPI) control is a sampling-based framework for solving Stochastic Optimal Control (SOC) problems in discrete-time dynamical systems. It is particularly effective in scenarios with complex, non-linear dynamics and non-convex cost functions, leveraging forward simulations to compute optimal control sequences.

A. Algorithm Overview

The MPPI algorithm solves SOC problems for systems governed by:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t + \mathbf{n}_t), \quad (1)$$

where $\mathbf{x}_t$ is the state vector at time t , $\mathbf{u}_t$ represents the control input, $\mathbf{n}_t \sim \mathcal{N}(0, \Sigma)$ denotes Gaussian noise, and f is the system's transition function.

At each iteration, K control sequences $\{\mathbf{U}_k\}_{k=1}^K$ are sampled from a Gaussian distribution $\mathcal{N}(\mathbf{u}_t, \nu\Sigma)$ and used to roll out state trajectories over a horizon T :

$$\mathbf{X}_k = \{\mathbf{x}_t, f(\mathbf{x}_t, \mathbf{u}_{k,t}), \dots, f(\mathbf{x}_{t+T-1}, \mathbf{u}_{k,T-1})\}. \quad (2)$$

For each trajectory, the associated cost S_k is computed:

$$S_k = \sum_{t=0}^{T-1} c(\mathbf{x}_t, \mathbf{u}_t) + c_T(\mathbf{x}_T), \quad (3)$$

where $c(\cdot)$ is the stage cost, and $c_T(\cdot)$ is the terminal cost.

The weights for importance sampling are calculated using:

$$w_k = \frac{\exp\left(-\frac{1}{\lambda}(S_k - S_{\min})\right)}{\sum_{j=1}^K \exp\left(-\frac{1}{\lambda}(S_j - S_{\min})\right)}, \quad (4)$$

where $S_{\min}$ is the minimum cost among all sampled trajectories, and λ is a temperature parameter.

The optimal control sequence is computed as:

$$\mathbf{U}^* = \sum_{k=1}^K w_k \mathbf{U}_k, \quad (5)$$

and the first control input $\mathbf{u}_0^*$ from $\mathbf{U}^*$ is applied to the system. This process is repeated at each time step.

B. Advantages of MPPI

Unlike optimization-based approaches that require linearization of dynamics or convex approximations of cost functions, MPPI directly incorporates non-linear and non-convex aspects. Its reliance on parallelizable sampling enables efficient computation, making it suitable for real-time applications. Additionally, MPPI does not require explicit gradient computations, further enhancing its robustness in handling discontinuous or complex cost landscapes.

IV. SIMULATION ENVIRONMENT

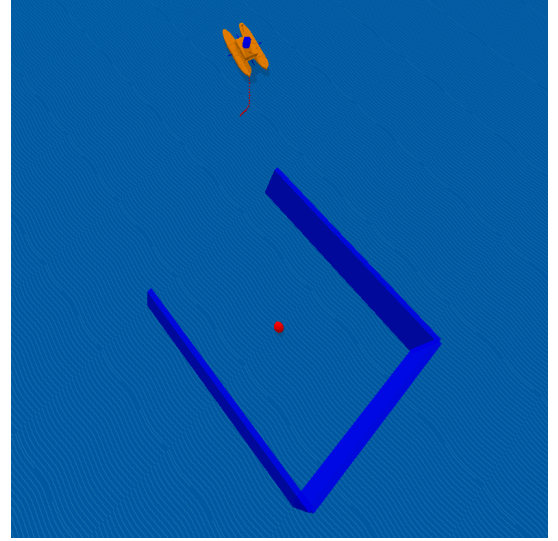


Fig. 1. Simulation Environment

The docking simulations were conducted using a PyBullet-based simulation environment, which provides physics-based modeling and visualization capabilities. The environment incorporates several key features designed to replicate realistic vessel docking scenarios.

A. Vessel Dynamics

The surface vessel is modeled as a fully actuated rigid body operating in 3 degrees of freedom (surge, sway, and yaw). The state vector is defined as:

$$\mathbf{q}_i = [\eta_i^T \ \nu_i^T]^T = [x_i \ y_i \ \psi_i \ u_i \ v_i \ r_i]^T \quad (6)$$

where the state vector is a combination of the 2D pose

$$\eta = [x \ y \ \psi]^T \quad (7)$$

and the velocity vector

$$\nu = [u \ v \ r]^T \quad (8)$$

in body-fixed coordinates.

The dynamic model of the vessel used in running simulations, according to [5], is given by:

$$M\dot{\nu} + C_{RB}(\nu)\nu + N\nu = \tau_c + \tau_d \quad (9)$$

$$\dot{\eta} = J(\psi)\nu \quad (10)$$

The mass matrix M , the Coriolis matrix $C_{RB}(\nu)$, and the damping matrix N include the system parameters. The input vector is denoted by τ_c , and the disturbance vector is denoted by τ_d . The disturbance vector τ_d is set to zero for simplicity for this study. The kinematic equation (10) describes the transformation of the body-fixed velocity into local coordinates as a function of the rotation matrix $J(\psi)$. The action input vector has 4 thrust values which helps the vessel to move in all 3 degrees of freedom.

B. Physical Environment

The simulation environment consists of a water surface rendered with realistic textures and a docking bay structure. The docking bay is modeled with precise physical dimensions:

- Dock width: 4 meters
- Wall thickness: 0.1 meters
- Safety clearance zone: 3.5 meters from center

The dock structure includes three primary walls forming a U-shaped berthing area, with wall positions precisely defined relative to the dock center position (10.0, -5.0, 0.0) in the global coordinate frame.

C. Sensor Simulation

The simulation includes realistic sensor modeling of 2D LiDAR System. A 360-degree LiDAR sensor is simulated with the following specifications:

- Angular resolution: 0.1 degrees (3600 rays)
- Maximum range: 50 meters
- Update rate: 5 Hz
- Gaussian noise with $\sigma = 0.1$ meters

D. Collision Detection

The environment implements a collision detection system that monitors:

- Contact points between vessel corners and dock walls
- Dynamic safety zones with critical (0.25m) and warning (0.5m) thresholds
- Real-time collision state feedback for termination conditions

This simulation environment provides a realistic testbed for evaluating the MPPI controller's performance in autonomous vessel docking scenarios, incorporating both the physical complexities of marine operations and practical sensing considerations.

V. LIDAR SCAN PROCESSING

The LiDAR processing pipeline is designed to extract critical geometric features of the dock environment, including the entry point, dock center, orientation, and clearances. These features are subsequently used to guide the vessel's docking maneuvers.

A. Point Cloud Generation

LiDAR scans provide a set of range and angle measurements. The range data is filtered to exclude points beyond the maximum valid distance, $d_{\max}$, and the corresponding (x, y) coordinates are computed as:

$$x = r \cos(\theta), \quad y = r \sin(\theta), \quad (11)$$

where r and θ are the range and angle of each LiDAR beam, respectively. The resulting point cloud, $\mathbf{P} = \{(x_i, y_i)\}$, represents the spatial distribution of objects around the vessel.

B. Clustering and Dock Identification

The docking station points are extracted using the DBSCAN clustering algorithm [6]. Clusters with sufficient density are retained, and the dock points $\mathbf{P}_{\text{dock}}$ are identified. The centroid of $\mathbf{P}_{\text{dock}}$ is calculated to approximate the dock center:

$$\mathbf{C}_{\text{dock}} = \frac{1}{N} \sum_{i=1}^N \mathbf{p}_i, \quad \mathbf{p}_i \in \mathbf{P}_{\text{dock}}, \quad (12)$$

where N is the number of points in $\mathbf{P}_{\text{dock}}$.

C. Wall Segmentation and Orientation Estimation

To identify individual dock walls, the Gaussian Mixture Model (GMM) [7] is applied to segment $\mathbf{P}_{\text{dock}}$ into M clusters. For each cluster, a robust line is fitted using RANSAC [8], yielding parameters for the wall as shown in Figure 2:

$$y = mx + c, \quad (13)$$

where m is the slope and c is the intercept.

Parallel walls are identified by comparing the slopes of the fitted lines. The dock orientation is determined as the average angle of these parallel walls:

$$\theta_{\text{dock}} = \frac{\arctan(m_1) + \arctan(m_2)}{2}. \quad (14)$$

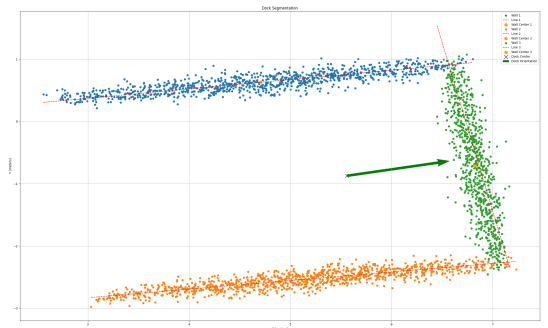


Fig. 2. Dock Point Cloud Segmentation

D. Clearance Calculation

The clearance from the vessel to each wall is computed as the minimum distance from the vessel's corners to the wall:

$$d_{\text{clearance}} = \min \left(\frac{|mx_0 - y_0 + c|}{\sqrt{m^2 + 1}} \right), \quad (15)$$

where (x_0, y_0) are the coordinates of the vessel's corner points.

E. Entry Point Estimation

The dock entry point is computed as a point offset from the dock center along the dock orientation towards the open dock face:

$$\mathbf{P}_{\text{entry}} = \mathbf{C}_{\text{dock}} + d_{\text{entry}} \begin{bmatrix} \cos(\theta_{\text{dock}} + \pi) \\ \sin(\theta_{\text{dock}} + \pi) \end{bmatrix}, \quad (16)$$

where d_{entry} is the desired offset distance.

F. Transformation to World Frame

All calculations are performed in the vessel's local frame. To transform points into the world frame, the rotation and translation induced by the vessel's pose (x, y, θ) are applied:

$$\mathbf{P}_{\text{world}} = \mathbf{R}\mathbf{P}_{\text{local}} + \mathbf{t}, \quad (17)$$

where $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ is the translation vector:

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} x \\ y \end{bmatrix}. \quad (18)$$

VI. COST FORMULATION

The cost formulation is a central aspect of the Model Predictive Path Integral (MPPI) controller, ensuring optimal decision-making by penalizing undesirable states and actions while encouraging desired behaviors. Below, we elaborate on the primary components of the cost function used.

A. Dock Goal Cost

The dock goal cost encourages the vessel to move toward the docking position:

$$C_{\text{dock_goal}} = w_{\text{dock_goal}} \cdot \|\mathbf{p} - \mathbf{p}_{\text{dock}}\|_2, \quad (19)$$

where $\mathbf{p}$ is the current position of the vessel, $\mathbf{p}_{\text{dock}}$ is the dock center position, and $w_{\text{dock_goal}}$ is the weight for dock alignment.

B. Velocity Cost

The velocity cost penalizes deviations from desired motion characteristics in the body frame. The velocity cost formulation is similar to Streichenberg et al [3] :

$$C_{\text{velocity}} = w_{\text{back}} \cdot \text{ReLU}(-v_x) + w_{\text{lat}} \cdot v_y^2 + w_{\text{rot}} \cdot \omega^2 + w_{\text{max_speed}} \cdot (\|\mathbf{v}\| - v_{\text{max}})^2, \quad (20)$$

where v_x and v_y are forward and lateral velocities, ω is the angular velocity, and v_{max} is the maximum allowed speed of 0.3 m/s.

C. Dock Heading Cost

This cost ensures proper vessel orientation toward the dock when sufficiently far:

$$C_{\text{dock_heading}} = \begin{cases} w_{\text{dock_heading}} \cdot \theta_{\text{err}}^2 & \text{if } d > d_{\text{th}} \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

where θ_{err} is the angle between the vessel's heading and the vector to the dock center, d is the distance to dock, and d_{th} is the distance threshold.

D. Dock Orientation Cost

The dock orientation cost enforces proper alignment when near the dock:

$$C_{\text{goal_ori}} = \begin{cases} w_{\text{goal_ori}} \cdot (\theta - \theta_{\text{dock}})^2 & \text{if } d < d_{\text{th}} \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

where θ is the vessel's heading and θ_{dock} is the dock's orientation.

E. Dock Clearance Cost

To ensure safe operation near the dock walls, a piecewise clearance cost is implemented:

$$C_{\text{dock_clear}} = w_{\text{dock_clear}} \cdot \begin{cases} 10 & \text{if } d_{\text{min}} < d_{\text{crit}} \\ 5 & \text{if } d_{\text{crit}} \leq d_{\text{min}} < d_{\text{warn}} \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

where d_{min} is the minimum distance from vessel corners to dock walls, d_{crit} is the critical distance threshold of 0.25m, and d_{warn} is the warning distance threshold of 0.5m.

F. Dock Entrance Cost

The dock entrance cost guides the vessel through a safe approach trajectory:

$$C_{\text{dock_entrance}} = \begin{cases} w_{\text{dock_entrance}} \cdot \|\mathbf{p} - \mathbf{p}_{\text{ent}}\|_2 & \text{if } \|\mathbf{p}_{\text{curr}} - \mathbf{p}_{\text{ent}}\|_2 > 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

where $\mathbf{p}_{\text{ent}}$ is the entrance point position and $\mathbf{p}_{\text{curr}}$ is the current vessel position.

G. Total Cost

The total cost is computed as the sum of the individual components:

$$C_{\text{total}} = C_{\text{dock_goal}} + C_{\text{velocity}} + C_{\text{dock_heading}} + C_{\text{goal_ori}} + C_{\text{dock_clear}} + C_{\text{dock_entrance}}. \quad (25)$$

This formulation balances various objectives, ensuring the vessel achieves precise docking while maintaining safety and efficiency throughout the maneuver.

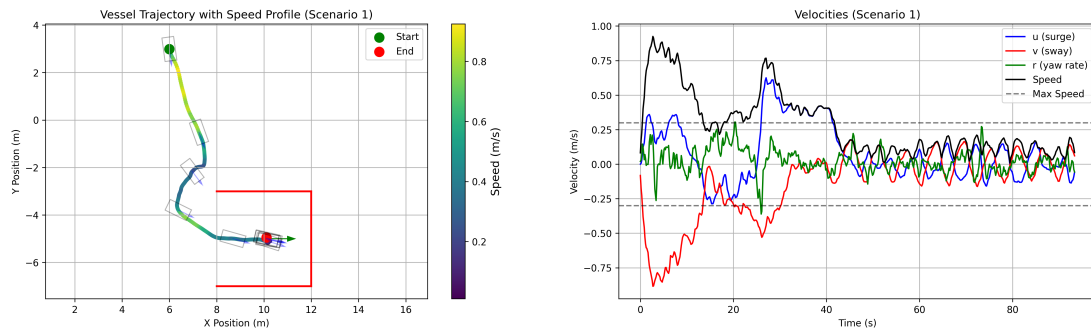


Fig. 3. Scenario 1

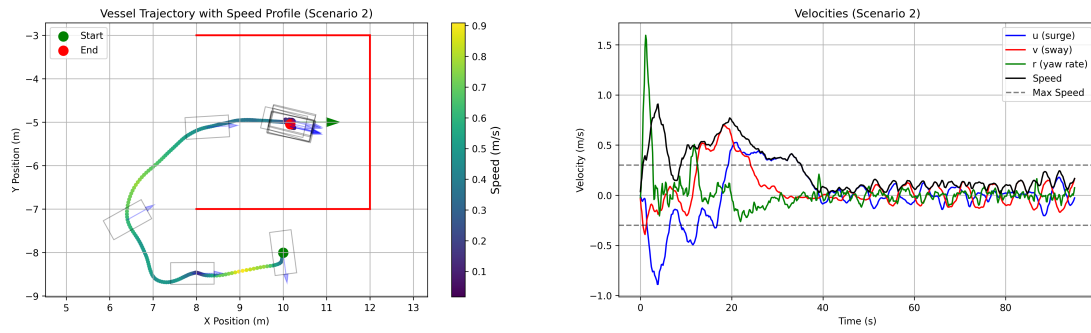


Fig. 4. Scenario 2

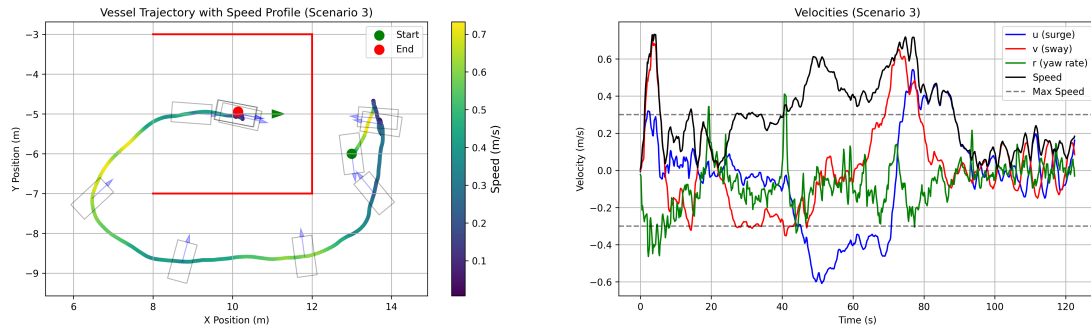


Fig. 5. Scenario 3

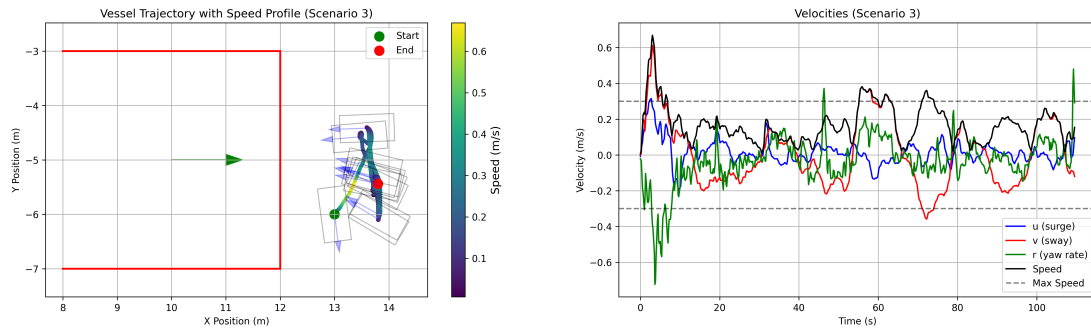


Fig. 6. Scenario 3 (Failed)

TABLE I
COST PARAMETERS USED IN THE OBJECTIVE FUNCTION

Parameter	Symbol	Value
Dock Goal Weight	$w_{\text{dock_goal}}$	1.0
Back Velocity Weight	w_{back}	0.08
Rotational Velocity Weight	w_{rot}	10.0
Lateral Velocity Weight	w_{lat}	1.0
Maximum Speed Weight	$w_{\text{max_speed}}$	5.0
Within Goal Orientation Weight	$w_{\text{goal_ori}}$	1.0
Dock Heading to Goal Weight	$w_{\text{dock_heading}}$	3.0
Dock Clearance Weight	$w_{\text{dock_clear}}$	2.0
Dock Entrance Weight	$w_{\text{dock_entrance}}$	3.0
Maximum Speed	v_{max}	0.3 m/s
Critical Clearance Threshold	d_{crit}	0.25 m
Warning Clearance Threshold	d_{warn}	0.5 m
Distance Threshold	d_{th}	0.5 m

VII. RESULTS AND DISCUSSION

The proposed MPPI-based docking controller was evaluated through extensive simulation studies under varying initial conditions, with results categorized into three distinct scenarios. In Scenario 1 (Fig. 3), the vessel begins from a position directly in front of the dock, consistently achieving successful docking maneuvers due to optimal LiDAR visibility of the dock structure. Scenario 2 (Fig. 4) demonstrates the controller's capability to handle lateral approaches, successfully guiding the vessel from a side position while maintaining safe clearances. Scenario 3 (Fig. 5) represents the most challenging configuration, with the vessel approaching from behind the docking station, where success rates become variable due to limited dock visibility. Failed attempts (Fig. 6) primarily occur when approaching from the dock's rear, attributed to poor dock center detection and the stochastic nature of MPPI decision-making under uncertainty. The controller exhibits high reliability in frontal and lateral approaches but shows sensitivity to initialization conditions in scenarios with limited visibility. Notable limitations include small residual velocities persisting after reaching the docking position and unstable behavior observed when the maximum speed weight exceeds 50, occasionally causing the vessel to undock after reaching the target position. These findings highlight the need for refined cost function formulation, particularly in terminal state handling, and suggest potential improvements through enhanced dock detection algorithms or additional sensing modalities for scenarios with limited visibility. Despite these limitations, the controller demonstrates robust performance across most common docking scenarios, effectively balancing safety constraints with docking precision.

VIII. CONCLUSION

This work has demonstrated the effectiveness of Model Predictive Path Integral (MPPI) control for autonomous vessel docking through extensive simulation studies, offering a comprehensive framework that balances docking precision with safety constraints. While our simulation results show promising performance in controlled scenarios, significant challenges remain in transitioning to real-world implementation. These

include robust localization in noisy GPS environments, reliable dock detection and segmentation under varying lighting and weather conditions, and accurate estimation of vessel dynamics in the presence of environmental disturbances such as waves, wind, and currents. Our current implementation has identified several areas for improvement, including the need to eliminate residual velocities at the docking position and enhance reliability when approaching from positions with limited dock visibility. Future work should focus on addressing these limitations through adaptive cost function tuning, integration of multiple sensing modalities (LiDAR, vision, radar), and robust point cloud processing algorithms that can handle noisy and incomplete sensor data. Looking to the future, significant advancements can be made by making the weights of the cost function adaptive to dynamic environmental conditions, mission objectives, and vessel states. Furthermore, replacing hand-tuned cost functions with reward functions learned from expert demonstrations via inverse reinforcement learning (IRL) presents an exciting opportunity. By capturing the implicit objectives of human operators, IRL could enable MPPI to generalize across diverse scenarios and reduce the dependency on manual parameter tuning.

REFERENCES

- [1] 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. [Online]. Available: <https://doi.org/10.2514/1.G001921>
- [2] H. Homburger, S. Wirtensohn, and J. Reuter, "Docking control of a fully-actuated autonomous vessel using model predictive path integral control," in *Proceedings of the 20th European Control Conference (ECC)*, 12-15 July 2022, London, United Kingdom, 2022, pp. 755 – 760.
- [3] L. Streichenberg, E. Trevisan, J. J. Chung, R. Siegwart, and J. Alonso-Mora, "Multi-agent path integral control for interaction-aware motion planning in urban canals," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2023, p. 1379–1385. [Online]. Available: <http://dx.doi.org/10.1109/ICRA48891.2023.10161511>
- [4] A. M. Patel, M. J. Bays, E. N. Evans, J. R. Eastridge, and E. A. Theodorou, "Model-predictive path-integral control of an unmanned surface vessel with wave disturbance," in *OCEANS 2023 - MTS/IEEE U.S. Gulf Coast*, 2023, pp. 1–7.
- [5] T. Fossen, *Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles*, 12 2002, vol. 28.
- [6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*. AAAI Press, 1996, pp. 226–231.
- [7] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
- [8] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.