2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Maneuverability Ellipsoid for Analyzing Sampling Space of Model Predictive Path-Integral Control for 4WIDS Robot Navigation

Ryo Ueda and Genya Ishigami

***Abstract*— Four-wheel independent drive and steering (4WIDS) robot possesses high maneuverability, with exploiting its stemming 8-DoF (Degrees of Freedom) control inputs, for navigation itself in complex, obstacle-rich environments. In such scenarios, Model Predictive Path Integral (MPPI) control, a sampling-based approach to model predictive control, has emerged as a powerful technique. While MPPI can effectively handle nonlinear dynamics and non-differentiable cost functions, it confronts the fundamental challenge of *curse of dimensionality*, where the number of required samples grows exponentially with the dimension of the control space. Although dimensionality reduction of the control space is a known countermeasure for suppressing the computational burden of MPPI, a systematic analysis for rationally designing the reduced space remains as an open issue. This research therefore addresses this issue by first proposing a novel metric, *Maneuverability Ellipsoid* of the 4WIDS, to quantify the robot's maneuvering capability with regard to the multiple DoFs of the robot control inputs. Based on this ellipsoid, we numerically analyze a sampling method for selecting variables that contribute higher maneuverability in the MPPI framework. The robot maneuverability index is also proposed that is quantified by the size and shape of the manipulability ellipsoid. Through simulations, we demonstrate that this index significantly correlates with the success rate of robot navigation.**

I. INTRODUCTION

A four-wheel independent steering and driving (4WIDS) robot is equipped with individual motors for both driving and steering on all four wheels. This 8-degree-of-freedom (DOF) configuration enables omnidirectional movement. Unlike differential-drive robots, which can perform point turns but cannot move diagonally in a straight line, 4WIDS robots can execute both maneuvers [1]. Consequently, robots with four-wheel independent drive and steering are widely deployed as logistics, scientific survey, and specialized agricultural vehicles [2]. While other omnidirectional robots using omni-wheels or Mecanum wheels exist, they are often subject to less controllability due to its mechanical constraints (i.e. sliding rollers on the wheel surface). The 4WIDS robot overcomes these limitations, making it well-suited for applications in obstacle-rich, cluttered environments and require the transport of heavy goods [3]. However, despite its high maneuverability, the complexity of its 8-DOF system presents a significant control challenge [4], [5].

For the autonomous navigation of such robots, Model Predictive Control (MPC) has been a promising approach [6]. MPC can solve local path planning and trajectory generation simultaneously. Unlike the Dynamic Window Approach (DWA), MPC can handle time-varying control inputs, and compared to PID control, it more readily incorporates the robot's physical constraints [7]. Nevertheless, standard MPC relies on gradient-based optimization, which cannot handle non-differentiable cost functions and often converges to local optima in nonlinear problems [8], [9]. To address these issues, a sampling-based MPC method, Model Predictive Path Integral (MPPI) control, was developed and has been successfully applied not only to wheeled robots but also to a wide range of other systems, including robotic arms [10]. When applying MPPI to a 4WIDS robot, the selection of sampling variables in the control space is a critical issue. Although MPPI can handle nonlinear models and non-differentiable costs like collision costs, it suffers from the *curse of dimensionality*; the required number of samples increases exponentially with the dimension of the sampling space, leading to poor computational efficiency [12]. Furthermore, sampling all control variables may result in sampling states that are physically impossible for the robot to achieve due to its movement constraints, which lowers the navigation success rate. Therefore, instead of sampling all eight variables corresponding to the 8 DOFs, it is necessary to sample in a reduced-dimensional space. However, no prior research has provided a systematic analysis for selecting this sampling space. Therefore, the objective of our research is to mathematically quantify the control space for MPPI applied to 4WIDS robots. The main contributions of this study are as follows:

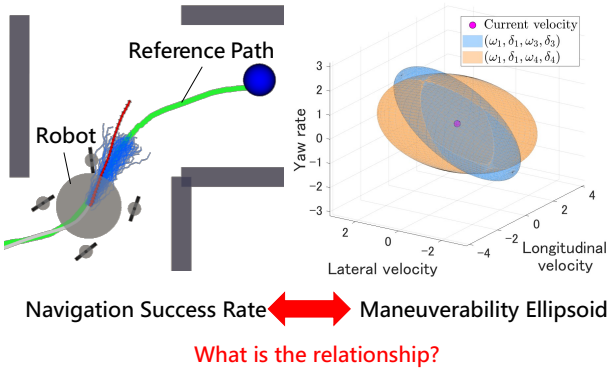


Fig. 1: The left figure shows a 4WIDS robot following and navigating a reference path. The right figure shows maneuverability ellipsoids of a wheeled robot based on the MPPI sampling space. This study examines the correlation between the navigation success rate and the size and shape of the maneuverability ellipsoid.

*This work was not supported by any organization

R. Ueda and G. Ishigami are with the Space Robotics Group, Department of Mechanical Engineering, Keio University, Yokohama 223-8522, Japan ryo14398@keio.jp, ishigami@mech.keio.ac.jp

1) We enumerated the possible combinations of sampling variables for MPPI based on the kinematic models of the 4WIDS robot.
2) Through simulation experiments like Fig. 1, we identified the combination of sampling variables and kinematic models that achieves the highest success rate in navigation tasks.
3) We introduce the concept of a **Maneuverability Ellipsoid** to in Fig. 1 quantify the robot's maneuvering capability, providing a theoretical explanation for why the identified sampling space produces a high navigation success rate.

This work establishes a theoretical basis for selecting a control strategy for 4WIDS robots, a process that has previously been based only on qualitative assessment.

II. RELATED WORK

A. Conventional Control Methods for 4WIDS Robots

Conventional control methods for 4WIDS robots have typically involved adding constraints to reduce the degrees of freedom (DOF) [12]. A representative approach is to simplify the four-wheel model into a two-wheel model, such as a bicycle model, thereby reducing the number of control variables. While this method is simple and computationally efficient, it has the drawback of not fully leveraging the high kinematic performance inherent to 4WIDS robots.

To address this, Model Predictive Control (MPC) is widely adopted due to its ability to flexibly handle diverse constraints and objectives. MPC is a method that predicts future states using a robot model and calculates the optimal control input that minimizes a cost function. Although the Dynamic Window Approach (DWA) is a similar method, it assumes that control inputs are constant over time, making MPC the more common choice in complex environments. However, standard MPC, which relies on gradient-based methods, struggles with complex-shaped obstacles and non-differentiable cost functions. Furthermore, its flexibility comes at the cost of high computational expense. While strategies such as linear approximation of the robot model and the use of fast, gradient-based optimization solvers have been employed to mitigate this cost, solving optimization problems involving non-convex and non-differentiable cost functions remains a significant challenge [13], [14].

Therefore, this research introduces MPPI (Model Predictive Path Integral control), a sampling-based MPC capable of addressing these issues, to achieve high-success-rate navigation even in narrow environments.

B. Application of MPPI to Redundant Systems

Sampling-based control methods have been applied across a wide range of robotics fields, and have been shown to be capable of controlling complex systems such as humanoid and quadruped robots [15]. However, in the case of 4WIDS robots, the sampling method must be carefully designed, as smooth steering motion is required.

MPPI facilitates the generation of smooth motion trajectories by incorporating a term for control input variation into its cost function. It is also applicable to highly nonlinear kinematic models, such as those of 4WIDS vehicles, which involve large changes in steering angles. Furthermore, MPPI offers the advantage of accommodating non-differentiable cost functions, which is effective in complex, obstacle-rich environments where such vehicles are often utilized.

Although MPPI has been applied to various systems, a systematic, mathematical quantification for selecting the sampling variables has not yet been established. For example, a previous study by Aoki et al. demonstrated the effectiveness of applying MPPI to a 4WIDS robot and switching the control space based on the environment [16]. This study suggests that sampling based on control variables such as wheel rotation speed and steering angle enhances success rates of robotic navigation than sampling based on the robot's velocity and angular velocity (V_x, V_y, ω_z). However, this study does not discuss the optimality of the selected sampling space. Specifically, it does not clarify the theoretical basis for selecting precisely 4 out of the 8 control variables possessed by a 4WIDS robot, nor does it reveal how the navigation success rate changes when the sampling space is constructed using 4 or fewer control variables. Therefore, this study aims to provide new insights into this issue by identifying the optimal control space for MPPI based on theoretical analysis and simulation verification.

III. NAVIGATION ARCHITECTURE

A. Overview

This research addresses the task of robot navigation in an environment with obstacles, where the robot must reach a predetermined destination while avoiding collisions with the obstacles. We assume that a global reference path from the robot's current position to the destination is pre-generated by a global path planner. This study focuses on a local path planning designed to follow this reference path while ensuring safe vehicle motion. The proposed navigation system consists of the following components:

1) Global Path Planning: a global path from the vehicle's current location to the target destination is generated using Dijkstra's algorithm.
2) Local Path Planning: we exploit Model Predictive Path Integral (MPPI) control for the local path planner. Searching the entire control space of an 8-Degrees-of-Freedom (8-DoF) vehicle model in the MPPI is inefficient due to the increased computational cost associated with the *curse of dimensionality*. Our approach involves reducing the control space to three or four dimensions and sampling solutions within this lower-dimensional space. This reduction approach has been demonstrated in the research by Aoki et al [16].
3) Control Input Conversion: the control inputs calculated by MPPI are converted into 8-DoF vehicle commands and transmitted to the robot actuators as command values.

In this study, the 4WIDS robot is considered as a kinematics-based model, assuming that the vehicle dynamics

are not considered and that no slip occurs between the wheels and the ground. This is because a rigorous treatment of a dynamics model would require the precise calculation of forces acting on each of the four wheels. This, in turn, necessitates the real-time measurement of parameters such as tire slip angles, slip ratios, and other tire-specific physical properties, which is extremely difficult to achieve in real-world environments.

B. Theoretical Background of MPPI

MPPI is a sampling-based, model predictive optimal control algorithm for computing an optimal sequence of control inputs over a near-future time horizon [17]. The algorithm operates by sampling a multitude of candidate control input sequences from a normal distribution centered around the optimal sequence obtained at the previous time step. It then evaluates each candidate sequence, assigning high weights to favorable sequences that minimize a cost function and low weights to others. Finally, the optimal control input sequence for the current time step is determined by calculating the weighted average of these sequences.

A key advantage of MPPI is its versatility, as it is applicable even when the system model or cost function is nonlinear, non-convex, or non-differentiable. Theoretically, the algorithm is guaranteed to be equivalent to a process that minimizes the Kullback-Leibler (KL) divergence between the probability distribution used for sampling and the true optimal control distribution that minimizes the total cost.

While many previous studies have been limited to specific sampling spaces, this research aims to evaluate the performance of MPPI under different sampling spaces and conduct a comparative analysis of their effectiveness.

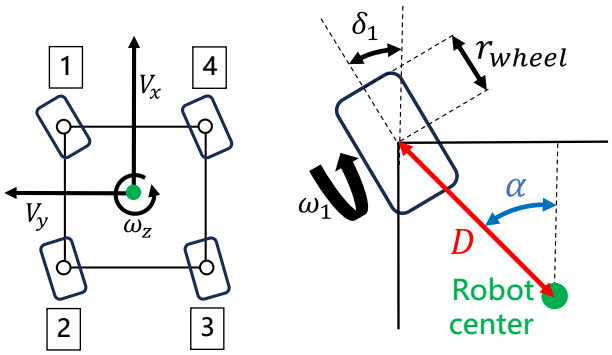
IV. 4WIDS Robot Model

To discuss the sampling space in MPPI, it is first necessary to address the kinematic model of a target vehicle and, to define the theoretically permissible sampling space based on that model.

A. 4WIDS Robot Kinematics

A 4WIDS robot has a total of 8 degrees of freedom because it can independently control the angular velocity and steering angle for each of its four wheels, as shown in Fig. 2a. As shown in Fig. 2b, we consider a robot model where D is the distance from the center of the body to each wheel, and α is the angle of each wheel's position relative to the body's reference axis. Given the angular velocity of i-th wheel is denoted by ω_i, its steering angle by δ_i, and the wheel radius by r_{wheel}, the following relationship holds:

$$r_{\text{wheel}} \begin{bmatrix} \omega_1 \cos\delta_1 \\ \omega_2 \cos\delta_2 \\ \omega_3 \cos\delta_3 \\ \omega_4 \cos\delta_4 \\ \omega_1 \sin\delta_1 \\ \omega_2 \sin\delta_2 \\ \omega_3 \sin\delta_3 \\ \omega_4 \sin\delta_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -D\sin\alpha \\ 1 & 0 & -D\sin\alpha \\ 1 & 0 & D\sin\alpha \\ 1 & 0 & D\sin\alpha \\ 0 & 1 & D\cos\alpha \\ 0 & 1 & -D\cos\alpha \\ 0 & 1 & -D\cos\alpha \\ 0 & 1 & D\cos\alpha \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix} \quad (1)$$



(a) Robot model (b) Wheel-1 model

Fig. 2: 4WIDS robot

TABLE I: Control space dimensions and variable combinations

Dimension	Combinations of control variables
3D	$(\omega_i, \delta_i, \delta_j)$, $(\omega_i, \delta_i, \omega_j)$
4D	$(\omega_i, \delta_i, \omega_j, \delta_j)$

The subscript i indicates the number of wheels denoted in Fig. 2a. The robot's state of motion is described by three variables: V_x, V_y, and ω_z. However, since there are eight control variables, this system possesses redundancy. Theoretically, by determining at least three of these eight control variables, the robot's velocity can be uniquely determined due to its kinematic constraints.

However, the combination of these three variables cannot be chosen arbitrarily. To uniquely determine the robot velocity, the selected set of variables must include both the angular velocity and the steering angle for at least one wheel. For example, if only three steering angles δ_1, δ_2, and δ_4 are chosen as inputs, only the direction of the robot's velocity vector can be determined, and the magnitude of the velocity cannot be uniquely quantified.

This constraint also applies when four or more variables are used as control inputs; the combination must always include both the angular velocity and the steering angle of at least one wheel.

B. Combinations of Control Variables

Based on the kinematics of the 4WIDS robot described previously, the possible combinations of control variables are shown in Table I. This study aims to achieve obstacle avoidance using the lowest possible dimensionality; therefore, we focus our investigation on 3-dimensional and 4-dimensional control spaces. This decision is supported by prior research from Aoki et al., in which MPPI using a 4-dimensional control space demonstrated a high success rate, suggesting it is sufficient for practical applications [16]. In the 4D control space, MPPI can be performed using combinations such as $(\omega_i, \delta_i, \delta_j, \delta_k)$ and $(\omega_i, \delta_i, \omega_j, \omega_k)$. However, to focus on the navigation success rate of MPPI and its numerical analysis in this study, only $(\omega_i, \delta_i, \omega_j, \delta_j)$ equivalent to the sampling

space used in Aoki et al.'s research was adopted in the 4D control space.

Furthermore, since the objective of this study is to select the optimal type of control space, combinations that can be considered equivalent due to their geometric symmetry such as (ω_1, δ_1, δ_3) and (ω_4, δ_4, δ_2) are treated as a single group.

C. Kinematic Constraints

Considering the case where the combination $(\omega_i, \delta_i, \omega_j)$ is used for the control variables, the pair of i and j is constrained due to the robot kinematics. For example, a tuple $(\omega_1, \delta_1, \omega_2)$ is associated with the following constraint given by the first and second lines of (1).

$$\delta_2 = \arccos\left(\frac{\omega_1 \cdot \cos\delta_1}{\omega_2}\right) \tag{2}$$

This relational expression indicates the existence of a purely mathematical constraint, separate from those arising from physical performance limits such as motor output. Specifically, for the steering angle δ to have a real solution, the argument of the arccosine (arccos) function must be within its domain of $[-1, 1]$.

Therefore, the term composed of the three aforementioned variables (ω_1, δ_1, ω_2) must satisfy the following condition:

$$-1 \leq \frac{\omega_1 \cdot \cos\delta_1}{\omega_2} \leq 1 \tag{3}$$

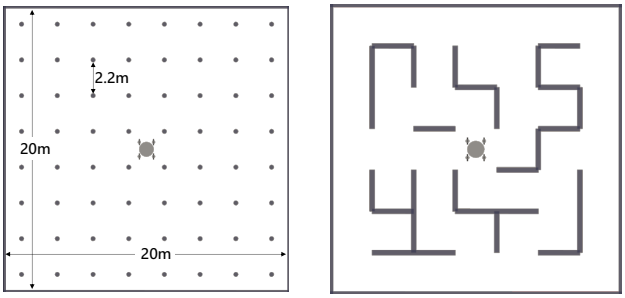
In this study, we address this issue by explicitly incorporating this mathematical constraint into the sampling process, thereby eliminating kinematically infeasible combinations and achieving a consistent sampling method.

V. SIMULATION EXPERIMENTS

A. Environment Setup

To evaluate the performance of the proposed navigation system, we constructed simulation environments using Gazebo. We prepared two types of environments, Cylinder Garden and Maze, as shown in Fig. 3. In these settings, the robot was tasked with navigating 100 paths while avoiding densely placed obstacles. These 100 paths were set to traverse routes among the 100 goal points defined in the study by Aoki et al [16]. The navigation architecture is shown in Fig. 4. First, a global planner generates a global reference path based on a given goal destination using Dijkstra's algorithm. Then, to follow this reference path, a local planner determines the optimal local path and control inputs at each time step using MPPI control. Furthermore, to match the robot model from their research, we set the parameters $D = \sqrt{2}/2$ meters and $\alpha = \pi/4$, and arranged the wheels symmetrically.

The evaluation system was developed using ROS and C++. Robot dynamic behavior was also calculated by Gazebo simulation. The computations were performed on a laptop PC equipped with an Intel(R) Core(TM) i7-1260P processor and 16GB of RAM. CPU multi-threading via OpenMP was utilized to accelerate computations. It should be noted that the CPU performance in this study is lower than that in Aoki et al.'s research, leading to a tendency for longer MPPI computation times.



(a) Cylinder Garden (b) Maze

Fig. 3: Simulation environment

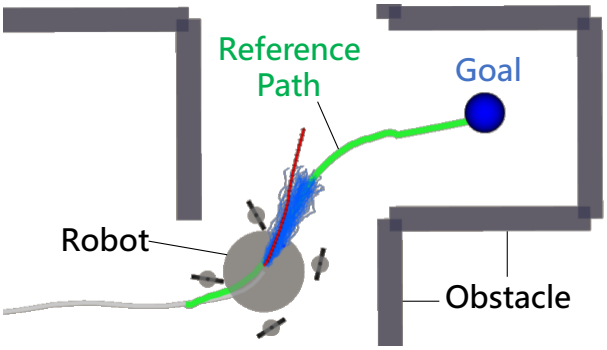


Fig. 4: The robot follows a reference path that is computed using Dijkstra's algorithm based on the goal destination. The control input is determined by MPPI control.

TABLE II: Parameter Set

(a) MPPI Params

Param	Value	Unit
K	3000	sample
T	30	step
Δt	0.033	s
α	0.1	-
λ	100	-
γ	2.5	-

(b) Controller params

Param	Value	Unit
Target Velocity v_{des}	2.00	m/s
Max. velocity $v_{\max}$	2.00	m/s
Max. yaw rate $\omega_{\max}$	1.58	rad/s
Max. steering angle	1.58	rad
Σ_ω	5.00	rad/s
Σ_δ	0.78	rad

B. Cost Function

The cost function used in the MPPI controller consists of a stage cost, $c(\mathbf{x}, \mathbf{u})$, and a terminal cost, $\phi(\mathbf{x})$. Based on the research by Aoki et al., the stage cost c was defined as a weighted sum of the following terms [16]:

$$\begin{aligned} c(\mathbf{x}, \mathbf{u}) = 40c_{\text{dist}}(p_x, p_y) + 30c_{\text{angle}}(\theta) + 10c_{\text{speed}}(v) \\ + 50c_{\text{collision}}(p_x, p_y) + c_{\text{cmd}}(\mathbf{u}), \end{aligned} \tag{4}$$

Here, $\mathbf{x} = (p_x, p_y, \theta)$ represents the vehicle's state (position and orientation), and $\mathbf{u}$ represents the control input. The definitions and formulas for the cost function are identical to those used in the prior study.

The terminal cost ϕ was defined as the squared error from the target position and expressed by the following equation:

$$\phi(\mathbf{x}) = 10\phi_{\text{goal}}(p_x, p_y), \tag{5}$$

TABLE III: Evaluation results of 100 navigation episodes

Blue value is the best score, and red value is the worst score of all controllers.

	Cylinder Garden					Maze				
Control Space	Cost [-] ↓	Calculation Time [ms] ↓	Steer. Rate [rad/s] ↓	Wheel Acc. [m/s^2] ↓	Success [%] ↑	Cost [-] ↓	Calculation Time [ms] ↓	Steer. Rate [rad/s] ↓	Wheel Acc. [m/s^2] ↓	Success [%] ↑
$[\omega_1, \delta_1, \delta_2]$	1.27×10^5	55.4	4.84	23.88	21	1.56×10^5	56.0	4.76	25.26	29
$[\omega_1, \delta_1, \delta_3]$	1.09×10^7	51.0	3.78	27.14	39	6184.0	48.7	3.36	23.56	26
$[\omega_1, \delta_1, \delta_4]$	-	-	-	-	0	-	-	-	-	0
$[\omega_2, \delta_2, \delta_1]$	9.65×10^5	53.0	5.54	38.58	31	1.38×10^5	53.0	5.76	30.96	21
$[\omega_2, \delta_2, \delta_3]$	-	-	-	-	0	-	-	-	-	0
$[\omega_2, \delta_2, \delta_4]$	1153.5	57.7	0.60	1.60	4	766.6	62.5	0.74	1.66	1
$[\omega_1, \delta_1, \omega_2]$	1453.3	72.6	1.88	11.44	51	1334.3	55.4	1.56	9.90	37
$[\omega_1, \delta_1, \omega_3]$	1714.4	69.3	1.84	7.60	75	1592.6	60.8	1.58	6.54	78
$[\omega_1, \delta_1, \omega_4]$	1636.0	58.5	1.94	8.20	50	1442.3	50.3	1.50	5.98	41
$[\omega_2, \delta_2, \omega_1]$	1692.4	64.4	2.08	11.44	31	1596.2	54.9	1.88	10.34	26
$[\omega_2, \delta_2, \omega_3]$	1543.7	58.9	1.80	8.98	83	1509.0	52.3	1.64	7.84	82
$[\omega_2, \delta_2, \omega_4]$	2497.1	59.7	2.52	8.22	16	2341.9	50.0	2.50	7.96	12
$[\omega_1, \omega_2, \delta_1, \delta_2]$	1571.3	52.3	2.02	11.74	56	1690.4	55.0	2.06	11.96	42
$[\omega_1, \omega_3, \delta_1, \delta_3]$	1602.2	51.8	1.46	7.82	78	1494.3	50.4	1.72	10.68	86
$[\omega_1, \omega_4, \delta_1, \delta_4]$	1607.8	56.3	1.48	7.46	48	1548.7	41.9	1.40	6.78	44
$[\omega_2, \omega_3, \delta_2, \delta_3]$	1606.9	59.1	1.74	9.10	86	1586.7	46.7	1.38	7.46	81

C. Parameter Setup

In this study, we conducted simulations using the combinations of sampling variables described in Table I. The parameters for MPPI are listed in Table IIa, which were determined based on the previous studies and empirical tuning. To ensure a fair comparison, these parameters were kept constant across all simulations for every sampling space.

The standard deviation for the sampling noise was systematically set as detailed in Table IIb, following the principle that the standard deviation is half of the maximum control value. This approach is intended to facilitate a broad exploration of the control space by leveraging the property that approximately 95% of a normal distribution falls within two standard deviations of the mean. The magnitude of the standard deviation for each variable type—either angular velocity or steering angle—was held constant regardless of the specific sampling combination. For example, for a sampling space of $(\omega_1, \delta_1, \delta_2)$, the standard deviations were set to $(\Sigma_\omega, \Sigma_\delta, \Sigma_\delta)$, while for a space of $(\omega_1, \delta_1, \omega_2)$, they were set to $(\Sigma_\omega, \Sigma_\delta, \Sigma_\omega)$.

D. Evaluation Metrics

The following metrics are adopted to evaluate the performance of the simulations. The success rate is considered the primary metric. These indicators are also selected based on the study by Aoki et al [16].

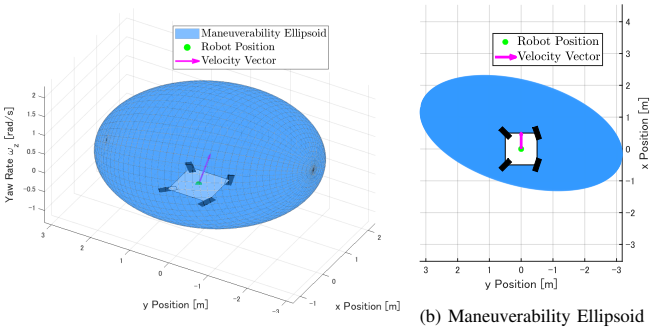
- Cost: this refers to the sum of the stage costs and the terminal cost along the optimal trajectory output by the MPPI controller. This metric indicates how effectively the controller minimized its cost function and approached optimal behavior.
- Control Input Variation: this is quantified by two sub-metrics: Steering rate, which represents the absolute value of the steering angular velocity, and Wheel acceleration, which represents the absolute value of the wheel's linear acceleration. For both, the average value across all four wheels is used in the evaluation.
- Success rate: the success rate is the percentage of trials in which the vehicle successfully reached its target destination in 100 navigation episodes. The main causes of failure are collisions with obstacles or falling into local minima in the environment.

E. Results

The evaluation results of the simulation are presented in Table III. Focusing on the success rate, which is the most critical metric, the control space combinations of $[\omega_1, \delta_1, \omega_3]$, $[\omega_2, \delta_2, \omega_3]$, $[\omega_1, \delta_1, \omega_3, \delta_3]$, and $[\omega_2, \delta_2, \omega_3, \delta_3]$ achieved a success rate of approximately 80% in both environments, surpassing the other combinations. It should be noted that the control space that yielded the highest success rate differed between the two environments. Although this suggests the potential effectiveness of adaptively switching the MPPI's control space according to the environment, a detailed investigation for this point is beyond the scope of this paper. These results demonstrate that even with identical cost functions and algorithm parameters, modifying the control space for exploration can significantly alter the navigation success rate.

VI. DISCUSSION

The most significant difference in navigation success rates among the control space combinations was observed in situations where the robot is surrounded by obstacles and required to perform a turn-in-place maneuver. In this section, we define a mobility performance metric based on the robot's kinematics and discuss the correlation between this metric and the navigation success rate.



(a) Maneuverability ellipsoid drawn in 3D

(b) Maneuverability Ellipsoid on xy plate

Fig. 5: The left figure shows the manipulability ellipsoid drawn in the three-dimensional coordinate system $[x, y, \theta]$ when the robot's current velocity is $(\hat{V}_x, \hat{V}_y, \hat{\omega}_z) = (0.5, 0, 0.5)$. The right figure shows its projection onto the xy-plane.

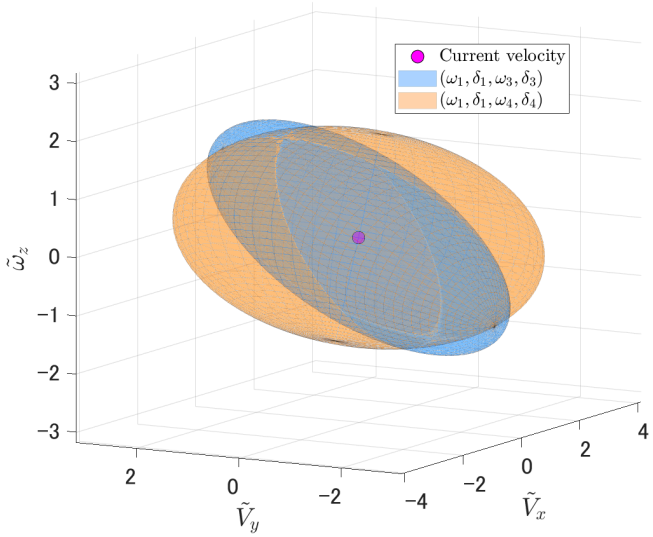


Fig. 6: This figure assumes a state where the robot's current velocity is $(\hat{V}_x, \hat{V}_y, \hat{\omega}_z) = (0.5, 0, 0.5)$. Under this condition, we compare the non-dimensionalized manipulability ellipsoids for two control spaces: $(\omega_1, \delta_1, \omega_3, \delta_3)$ and $(\omega_1, \delta_1, \omega_4, \delta_4)$. To quantify the maneuvering performance, we use the volume of the ellipsoid and the ratio of its major to minor axes.

A. *Maneuverability Ellipsoid of a Wheeled Robot*

Analogous to the manipulability ellipsoid for robotic arms, we define a **maneuverability ellipsoid** for wheeled robots. The manipulability ellipsoid of a robotic arm indicates the ease of moving the end-effector position, whereas the maneuverability ellipsoid of a wheeled robot indicates the ease of changing its current velocity.

1) Theory of Maneuverability: The robot's velocity vector $\mathbf{V}$, can be described as a function of the control input vector $\mathbf{u}$. The relationship of their infinitesimal changes is expressed using a Jacobian matrix, J, as follows:

$$\dot{\mathbf{V}} = J\dot{\mathbf{u}} \tag{6}$$

In contrast to the mapping from joint velocity to end-effector velocity in robotic arms, this study considers the mapping from time derivative of control input, $\dot{\mathbf{u}}$, to time derivative of the robot body velocity, $\dot{\mathbf{V}}$. Given the control input vector $\mathbf{u} = (\omega_1, \delta_1, \omega_3, \delta_3)$, the Jacobian matrix $\mathbf{J}$ can be given by the following equation. At a specific operating point, the control input is denoted as $\hat{\mathbf{u}} = (\hat{\omega}_1, \hat{\delta}_1, \hat{\omega}_3, \hat{\delta}_3)$.

$$J = \begin{bmatrix} \frac{\cos\hat{\delta}_1}{2} & \frac{-\hat{\omega}_1 \sin\hat{\delta}_1}{2} & \frac{\cos\hat{\delta}_3}{2} & \frac{-\hat{\omega}_3 \sin\hat{\delta}_3}{2} \\ \frac{\sin\hat{\delta}_1}{2} & \frac{\hat{\omega}_1 \cos\hat{\delta}_1}{2} & \frac{\sin\hat{\delta}_3}{2} & \frac{\hat{\omega}_3 \cos\hat{\delta}_3}{2} \\ \frac{\cos\hat{\delta}_1}{2D\sin\alpha} & \frac{-\hat{\omega}_1 \sin\hat{\delta}_1}{2D\sin\alpha} & \frac{-\cos\hat{\delta}_3}{2D\sin\alpha} & \frac{\hat{\omega}_3 \sin\hat{\delta}_3}{2D\sin\alpha} \end{bmatrix} \tag{7}$$

This Jacobian serves as a linear algebraic map indicating how large the velocity vector $\mathbf{V}$ changes in response to an infinitesimal change $d\mathbf{u}$.

Based on the analogy with the manipulability ellipsoid of a robotic arm, we define the maneuverability ellipsoid for a wheeled robot as the mapping of $\dot{\mathbf{V}}$ for a given $\dot{\mathbf{u}}$ that satisfies the constraint $\|\dot{\mathbf{u}}\| \leq 1$. However, since the control inputs—angular velocity ω and steering angle δ—have different physical dimensions and scales, normalization is necessary. At this point, it is common practice to normalize the rotational speed $\dot{\omega}$ by the maximum angular acceleration $\dot{\omega}_{\max}$, and the steering angle $\dot{\delta}$ by the maximum steering angular velocity $\dot{\delta}_{\max}$. In other words, summarizing these, when the control input vector is $\mathbf{u} = (\omega_1, \delta_1, \omega_3, \delta_3)$, the normalized constraints are given by the following.

$$\|\Sigma\dot{\mathbf{u}}\| \leq 1 \quad \text{where} \quad \Sigma = \mathrm{diag}(\dot{\omega}_{\max}, \dot{\delta}_{\max}, \dot{\omega}_{\max}, \dot{\delta}_{\max}) \tag{8}$$

In this study, the control input is normalized using the covariance matrix Σ obtained from the MPPI algorithm. That is, the covariance matrix is defined by setting $(\dot{\omega}_{\max}, \dot{\delta}_{\max}) = (\Sigma_\omega, \Sigma_\delta)$. The set of $\dot{\mathbf{V}}$ under this constraint forms an ellipsoid that visually represents the robot's ease of velocity change or maneuvering capability. This allows us to determine in which directions and to what extent the robot can easily change its velocity from its current state. For example, Fig. 5a illustrates a maneuverability ellipsoid when the control input is $(\omega_1, \delta_1, \omega_3, \delta_3)$ and the robot's velocity is $(V_x, V_y, \omega_z) = (0.5, 0, 0.5)$. The orientation of this ellipsoid's axes corresponds to the singular vectors of the (normalized) Jacobian matrix, and the length of each axis corresponds to the respective singular value. The tip of the current velocity vector is the center of the maneuverability ellipsoid. The direction of the major axis of the ellipsoid indicates how easily the robot can change its velocity, while the direction of the minor axis reflects the relative difficulty of making such velocity changes.

2) Nondimensionalization of velocity vector: In the previous section, the control input was non-dimensionalized using the covariance matrix. This section further develops this approach by also non-dimensionalizing the velocity change vector, $\dot{\mathbf{V}}$, which is a mapping of the control input.

In the manipulability analysis of robotic arms, the ease of end-effector movement can be evaluated by defining the volume of the manipulability ellipsoid as a manipulability measure [18]. However, this metric is insufficient for evaluating wheeled robots as it primarily focuses on translational motion. This is because wheeled robots are capable of turning ($\omega_z \neq 0$) even when their translational velocity is zero ($V_x = V_y = 0$), a maneuver known as turn-in-place. Especially

TABLE IV: Comprehensive Evaluation Results Across Three States and Maze Navigation

Blue values are among the top 3 scores, while red values are among the bottom 3 scores.

	State 1			State 2			State 3			Success Rate
Control Space	**Vol.** ↑	**Iso.** ↓	**Index** [×100] ↑	**Vol.** ↑	**Iso.** ↓	**Index** [×100] ↑	**Vol.** ↑	**Iso.** ↓	**Index** [×100] ↑	**in Maze** [%] ↑
$[\omega_1, \delta_1, \delta_2]$	0.05	9.18	0.56	0.14	16.27	0.89	-	-	-	29
$[\omega_1, \delta_1, \delta_3]$	0.05	41.55	0.12	-	-	-	0.36	33.63	1.06	26
$[\omega_1, \delta_1, \delta_4]$	-	-	-	0.00	9.25×10^7	0.00	0.25	30.97	0.82	0
$[\omega_2, \delta_2, \delta_1]$	0.05	9.18	0.56	0.14	16.27	0.89	-	-	-	21
$[\omega_2, \delta_2, \delta_3]$	-	-	-	0.14	16.27	0.89	0.25	30.97	0.82	0
$[\omega_2, \delta_2, \delta_4]$	0.05	41.55	0.12	-	-	-	0.25	28.00	0.91	1
$[\omega_1, \delta_1, \omega_2]$	-	-	-	1.31	22.01	5.94	0.65	36.29	1.80	37
$[\omega_1, \delta_1, \omega_3]$	0.65	36.29	1.80	0.65	9.69	6.74	0.49	11.74	4.15	78
$[\omega_1, \delta_1, \omega_4]$	0.16	45.33	0.36	0.00	1.08×10^8	0.00	0.18	23.41	0.78	41
$[\omega_2, \delta_2, \omega_1]$	-	-	-	1.31	22.25	5.87	0.65	36.29	1.80	26
$[\omega_2, \delta_2, \omega_3]$	0.65	36.29	1.80	1.31	22.25	5.87	0.73	22.46	3.25	82
$[\omega_2, \delta_2, \omega_4]$	0.65	36.29	1.80	0.65	9.69	6.74	0.49	11.74	4.15	12
$[\omega_1, \omega_2, \delta_1, \delta_2]$	0.07	12.90	0.56	0.93	4.59	20.25	0.92	25.68	3.60	42
$[\omega_1, \omega_3, \delta_1, \delta_3]$	0.92	12.82	7.21	0.93	9.07	10.26	0.60	12.82	4.66	86
$[\omega_1, \omega_4, \delta_1, \delta_4]$	0.92	25.68	3.60	0.93	22.67	4.10	0.60	18.19	3.28	44
$[\omega_2, \omega_3, \delta_2, \delta_3]$	0.92	16.04	5.76	0.93	5.67	16.40	0.60	11.52	5.19	81

in obstacle-rich environments, this turn-in-place capability directly impacts the navigation success rate. Conventional manipulability analysis, which is limited to the xy-plane, is insufficient for wheeled robots as it fails to capture both translational and rotational performance simultaneously.

To address this, we propose a new metric. In our approach, we nondimensionalize the differential velocity vector $\dot{\mathbf{V}} = [\dot{V}_x, \dot{V}_y, \dot{\omega}_z]^T$. Specifically, the translational components $\dot{V}_x$ and $\dot{V}_y$ are normalized by the maximum translational velocity $r_{\text{wheel}}\omega_{\max}$, while the rotational component $d\omega_z$ is normalized by the maximum yaw rate $r_{\text{wheel}}\omega_{\max}/(D\sin(\alpha))$. This allows for a unified evaluation of both motion types. The $\dot{\mathrm{V}}$ discussed in this paper is not acceleration itself, but rather an indicator of how easily velocity can be changed—that is, responsiveness. Therefore, for normalization, the maximum acceleration (e.g., $r_{\text{wheel}}\dot{\omega}_{\max}$) was not used; instead, the maximum velocity was employed. Furthermore, given the physical characteristic that a motor's acceleration/deceleration limits generally correlate with its velocity limits, this approach can be considered valid. From the volume of this nondimensionalized manipulability ellipsoid, we then define a manipulability measure. The overall concept is summarized in Fig. 6.

3) Maneuverability measure and Isotropy: The volume of the maneuverability ellipsoid, defined in the previous section, can be calculated by multiplying all the singular values of the Jacobian matrix. We define this volume as the **Maneuverability Measure**. A larger value indicates that the robot can more easily change its velocity in all directions. However, for certain control space combinations, we observed a highly elongated ellipsoid with a significantly long major axis, similar to those found near a singular configuration in robotic arms. In fact, the navigation success rate in these control spaces is low, suggesting the validity of the hypothesis that reduced maneuverability affects performance.

Therefore, as another key metric alongside the Maneuverability Measure, we define **Isotropy**. Isotropy is the ratio of the maximum singular value, $\sigma_{\max}$, to the minimum singular value, $\sigma_{\min}$, of the maneuverability ellipsoid, given by the following equation:

$$\text{Isotropy} = \frac{\sigma_{\max}}{\sigma_{\min}} \tag{9}$$

The closer this value is to one, the more the ellipsoid resembles a sphere, indicating a uniform ease of velocity change in all motion. The evaluation of isotropy is particularly crucial in obstacle-rich environments, where the ability to change velocity in any direction, rather than being biased towards specific ones, is required.

In this study, to integrate these two metrics, we propose a new index calculated as **Maneuverability Measure / Isotropy**. This allows for the simultaneous evaluation of both how easily the robot can change its velocity (maneuverability) and how uniformly it can change its velocity in all directions (isotropy) with a single index.

B. Validation of Correlation between the Maneuverability Index and Success Rate

In this section, we validate the correlation between the integrated index (Maneuverability Measure / Isotropy) proposed in the previous section and the navigation success rate observed in our simulations.

As previously noted, the maneuverability ellipsoid is state-dependent, changing its shape according to the robot's velocity vector. Therefore, for our evaluation, we selected

the following three representative and particularly critical velocity states:

- **State 1: Forward motion** $(V_x, V_y, \omega_z) = (0.02, 0, 0)$
- **State 2: Turn-in-place** $(V_x, V_y, \omega_z) = (0, 0, 0.04)$
- **State 3: Curve motion** $(V_x, V_y, \omega_z) = (0.02, 0, 0.04)$

The primary reason for selecting these states is related to navigation in obstacle-rich environments. The robot's performance in maneuvers such as turning in place (State 2) and transitioning to translational movement at low speeds (State 1) is key to task success. When surrounded by obstacles or when the vehicle is facing directly opposite the reference path, it is necessary to reduce the robot's speed and then turn-in-place.

Table IV presents the relationship between maneuverability, isotropy, and navigation success rate based on three representative robot states (State 1–3). From this table, it is confirmed that combinations with a high navigation success rate—such as $(\omega_1, \delta_1, \omega_3, \delta_3)$, $(\omega_2, \delta_2, \omega_3, \delta_3)$, and $(\omega_2, \delta_2, \omega_3)$—tend to exhibit high maneuverability and isotropy in all states. This result suggests that mobility performance metrics derived from the manipulability ellipsoid have a positive correlation with the navigation success rate.

However, exceptional cases were observed. For certain combinations like $(\omega_2, \delta_2, \omega_4)$, the navigation success rate was low despite the maneuverability and isotropy metrics being relatively high. This fact indicates that performance evaluations based on these indices are not always universally applicable. Although this study limited the analysis to three states, it is conceivable that the mobility performance of the $(\omega_2, \delta_2, \omega_4)$ combination could be evaluated as lower when other states are taken into account.

Furthermore, since the singular values of the Jacobian matrix can be derived analytically, it is possible that a robot mechanism achieving both high maneuverability and isotropy could be theoretically designed in the future.

VII. CONCLUSIONS

In this paper, we proposed an analytical approach for 4WIDS maneuverability for evaluating the appropriate combination of the sampling space for the MPPI control framework. We first implemented MPPI with various control space combinations and evaluated their navigation success rates through simulation. We then introduced the concept of maneuverability ellipsoid as an index to quantify the motion performance of the wheeled robot. Specifically, we defined a single, integrated index derived from the volume (Maneuverability Measure) and shape (Isotropy) of the maneuverability ellipsoid, thereby representing complex motion performance.

A strong correlation was observed between our proposed index and the navigation success rate. Consequently, this study presents a guideline for selecting an effective sampling space in MPPI based on kinematic theory.

While this research provides a new perspective on selecting an effective sampling space, several challenges remain. Future research topics include establishing methods to dynamically adapt the sampling space according to the reference path, robot state, and environment; extending the method to dynamic obstacle environments; and, to confirm the validity of the proposed kinematic model, implementing the entire proposed framework on a real robot to verify its effectiveness in real environments.

REFERENCES

[1] P. Hang and X. Chen, Towards autonomous driving: Review and perspectives on configuration and control of four-wheel independent drive/steering electric vehicles, Actuators, vol. 10, no. 8, pp. 184, Aug. 2021.

[2] X. Lai, X. B. Chen, X. J. Wu, and D. Liang, A Study on Control System for Four-Wheels Independent Driving and Steering Electric Vehicle, Applied Mechanics and Materials, vol. 701-702, pp. 807–811, Dec. 2014.

[3] K. Kanjanawanishkul, Omnidirectional wheeled mobile robots: wheel types and practical applications, International Journal of Advanced Mechatronic Systems, vol. 6, no. 6, pp. 289–302, Feb. 2015.

[4] L. Jin, L. Gao, Y. Jiang, M. Chen, Y. Zheng, and K. Li, Research on the control and coordination of four-wheel independent driving/steering electric vehicle, Advances in Mechanical Engineering, vol. 9, no. 4, pp. 1–13, Apr. 2017.

[5] N. Zhang, J. Wang, Z. Li, N. Xu, H. Ding, Z. Zhang, K. Guo, and H. Xu, Coordinated Optimal Control of AFS and DYC for Four-Wheel Independent Drive Electric Vehicles Based on MAS Model, Sensors, vol. 23, no. 7, pp. 3505, Mar. 2023.

[6] Y. Jeong and S. Yim, Model predictive control-based integrated path tracking and velocity control for autonomous vehicle with four-wheel independent steering and driving, Electronics, vol. 10, no. 22, pp. 2812, Nov. 2021.

[7] D. Fox, W. Burgard, and S. Thrun, The Dynamic Window Approach to Collision Avoidance, IEEE Robotics and Automation Magazine, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[8] J. Frey, K. Baumgärtner, G. Frison, D. Reinhardt, J. Hoffmann, L. Fichtner, S. Gros, and M. Diehl, Differentiable Nonlinear Model Predictive Control, arXiv preprint arXiv:2505.01353, 2025.

[9] M. Ławryńczuk and R. Nebeluk, Computationally Efficient Nonlinear Model Predictive Control Using the L1 Cost-Function, Sensors, vol. 21, no. 17, pp. 5835, Aug. 2021.

[10] G. Williams, P. Drews, B. Goldfain, J. 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, Jul. 2017.

[11] M. Xanthidis, J. M. Esposito, I. Rekleitis, and J. M. Kane, Motion Planning by Sampling in Subspaces of Progressively Increasing Dimension, arXiv preprint arXiv:1802.00328, 2018.

[12] Y. Zhao, J. Wu, J. Han, Y. Zhao, R. Li, J. Shao, and J. Zhou, "Path tracking of a 4wis–4wid agricultural machinery based on variable look-ahead distance," *Computers and Electronics in Agriculture*, vol. 206, p. 107718, 2023.

[13] Y.-S. Yang, Y. Liu, Z. Li, H.-T. Chang, C. Fan, and X. Hu, "Latent linear quadratic regulator: Learning efficient MPC for nonlinear systems," in *Robotics: Science and Systems (RSS)*, 2023.

[14] J. Nubert, R. Grandia, F. Farshidian, and M. Hutter, "Benchmarking QP solvers for dynamic quadrupedal walking," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[15] Y. Liu, Z. Li, H.-T. Chang, Y.-S. Yang, C. Fan, and X. Hu, "Implicit communication in human-robot collaboration," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[16] N. Aoki, K. Honda, H. Okuda, and T. Suzuki, Switching Sampling Space of Model Predictive Path-Integral Controller to Balance Efficiency and Safety in 4WIDS Vehicle Navigation, arXiv preprint arXiv:2409.08648, 2024.

[17] 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. 1–14, Jan. 2017.

[18] T. Yoshikawa, "Manipulability of Robotic Mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 67–77, Jun. 1985.