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

Obstacle avoidance by Monte Carlo Model Predictive Control with sample generation based on the C/GMRES method

Takahiro Onizawa[1†], Kazuma Sekiguchi[2], and Kenichiro Nonaka[3]

[1,2,3]Department of Mechanical System Engineering, Tokyo City University, Tokyo, Japan
(E-mail: [1]g2481012@tcu.ac.jp,[2]ksekiguc@tcu.ac.jp,[3]knonaka@tcu.ac.jp)

Abstract: This paper proposes an obstacle avoidance controller based on Monte Carlo Model Predictive Control (MCMPC), utilizing samples generated by the C/GMRES method. MCMPC, a sample-based approach, is less prone to local minima, making it suitable for non-convex problems, such as mobile robot navigation. However, MCMPC tends to produce discontinuous control input that causes wear and tear on actuators. To mitigate this, the present study utilizes smooth control inputs generated by the C/GMRES method, reducing oscillations while exploring global solutions. This allows the solution to track an optimum, enabling it to find alternative global solutions and avoid deadlock. This paper validates the performance through comparisons with conventional approaches and ensures feasibility through experiments.

Keywords: Model predictive control, Monte Carlo, C/GMRES method, Leg/wheel mobile robot, Obstacle avoidance

1. INTRODUCTION

For planetary rovers, to overcome communication delays and bandwidth limitations, autonomous navigation is essential. It should be capable of obstacle avoidance of rocks and craters, adapt to various terrains, and minimize mechanical stress on the robot body [1]. Various types of rovers have been developed and deployed in past missions [2] [3]. Leg/wheel mobile robots [4] [5] are equipped with both legged and wheeled locomotion mechanisms. Its hybrid mechanism allows the robot to navigate through narrow and rough terrain, while dexterous control algorithm is required for operation.

Previous studies on leg/wheel mobile robots include PID control for obstacle avoidance and steep slopes [6], and the development of an adaptive leg placement control using point-to-point control [7]. However, these control approaches face difficulties in simultaneously satisfying multiple requirements, such as posture stabilization and constraint-aware motion planning, and may lead to singular postures. To address these challenges, Model Predictive Control (MPC) has been applied to handle multiple control objectives under constraints, for example, trajectory tracking control [8] [9]. For obstacle avoidance, an optimal configuration controller [10] and distributed MPC [11] for planar leg/wheel mobile robot were proposed. These studies utilized gradient-based optimization methods, however, they are known to prone to have convergence to local optima in complex environments, which may cause stacking of robots. To address this limitation, Monte Carlo Model Predictive Control (MCMPC), a sample-based optimization method [12–17], has been applied to handle multiple local optimal solutions. However, applying MCMPC can generate discontinuous inputs, potentially causing wear and tear on actuators.

This study proposes a method combining a gradient-based approach for smooth control inputs with a sampling-based approach to avoid local optima. While this combination has been explored previously [18–20], our prior work [23] integrated MCMPC with the C/GMRES method [21] [22]. It generates one MCMPC sample from the C/GMRES solution with smooth inputs and enables multimodal obstacle avoidance. However, significant control chattering still appeared due to persistent sampling of random inputs. To address this issue, the present study proposes an MCMPC approach where samples are generated by adding random perturbations to the C/GMRES solution. The method's effectiveness is verified through simulation and experiment.

† Takahiro Onizawa is the presenter of this paper.

2. PROPOSED METHOD

2.1. Nonlinear model predictive control

Suppose the model of nonlinear dynamics is described by $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ where $\boldsymbol{x}$ and $\boldsymbol{u}$ denote the state and control input, respectively, and $\boldsymbol{f}$ is a vector-valued function. Nonlinear model predictive control computes the optimal input that minimizes the evaluation function

$$J = \varphi(\boldsymbol{x}, t+T) + \int_t^{t+T} L(\boldsymbol{x}, \boldsymbol{u}, \tau) d\tau \tag{1}$$

where φ and L denote a terminal and stage cost.

2.2. Monte Carlo Model Predictive Control [12–16]

MCMPC computes control input by exploiting a sampling-based optimization technique known as the Monte Carlo method. In this approach, a large number of time-series samples for $\boldsymbol{x}$ and $\boldsymbol{u}$ that obeys randomly perturbated dynamics are generated around a quasi-optimal solution. These samples are then evaluated and resampled using the evaluation function (1), and samples with the small evaluation are used for control input. Since it does not require the computation of gradients of the evaluation function, it can handle discontinuous functions and cope with multimodal optimization problems.

2.3. C/GMRES method [21] [22]

The C/GMRES method is a real-time optimization algorithm for nonlinear model predictive control. This

Algorithm 1 Algorithm of the proposed method

Compute $\boldsymbol{\mu}_{\mathrm{cg,0:H-1|t}}$ using $\boldsymbol{\mu}_{\mathrm{cg,0:H-1|t-\Delta}}$ and $\boldsymbol{x}(t-\Delta)$
$\boldsymbol{\mu}_{\mathrm{p}}^{1:N-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathrm{p}}^{*1:N-1}, \sigma^2\right)$
$\boldsymbol{\mu}_{\mathrm{mc,0:H-1|t}}^{1:N-1} = \boldsymbol{\mu}_{\mathrm{cg,0:H-1|t}} + \boldsymbol{\mu}_{\mathrm{p}}^{1:N-1}$
$\boldsymbol{\mu}_{0:H-1|t}^{1:N} = [\boldsymbol{\mu}_{\mathrm{cg}}\, \boldsymbol{\mu}_{\mathrm{mc}}^{1:N-1}]^\top$
Calculation of predicted trajectories $\boldsymbol{x}_{k|t}^{1:N}$ using $\boldsymbol{\mu}_{k|t}^{1:N}$
Evaluate $\boldsymbol{J}^{1:N}$ using $\boldsymbol{\mu}_{k|t}^{1:N}$ and $\boldsymbol{x}_{k|t}^{1:N}$
Sort $\boldsymbol{J}^{1:N}$ in descending order to determine $\boldsymbol{u}(t)$
Resampling $\boldsymbol{\mu}_{\mathrm{p}}^{*1:N-1}$ with likelihood $\exp(-\boldsymbol{J}^{1:N-1})$
if J_{cg} is not the best rated **then**
  $\boldsymbol{\mu}_{\mathrm{cg,0:H-1|t}} = \boldsymbol{u}(t)$
end if
Update state $\boldsymbol{x}(t)$ using $\boldsymbol{u}(t)$ and $t \leftarrow t + \Delta$

method combines the continuation method with the GMRES algorithm and computes the optimal control input by solving the algebraic equations $\boldsymbol{F}(\boldsymbol{u}, \boldsymbol{x}, t) = 0$ with

$$\boldsymbol{F}(\boldsymbol{u}, \boldsymbol{x}, t) := \begin{bmatrix} H_u^\top(\boldsymbol{x}_0, \lambda_1, \boldsymbol{u}_0, \mu_0) \\ \boldsymbol{C}(\boldsymbol{x}_0, \boldsymbol{u}_0) \\ \vdots \\ H_u^\top(\boldsymbol{x}_{N-1}, \lambda_N, \boldsymbol{u}_{N-1}, \mu_{N-1}) \\ \boldsymbol{C}(\boldsymbol{x}_{N-1}, \boldsymbol{u}_{N-1}) \end{bmatrix} \tag{2}$$

where H is the Hamiltonian function given by

$$H(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}, \boldsymbol{\mu}) = L(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\mu}^\top \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{u}) \tag{3}$$

and H_u denotes the partial derivative of the Hamiltonian function with respect to the input $\boldsymbol{u}$. $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ denote the costate variables and Lagrange multipliers, respectively, and $\boldsymbol{C}$ represents the equality constraints on the state and control input, expressed as $\boldsymbol{C}(\boldsymbol{x}, \boldsymbol{u}) = 0$. By solving Eq. (2), the control input that satisfies the stationarity condition of the evaluation function can be obtained. In general, it is difficult to obtain the control input in real-time that $\boldsymbol{F}(\boldsymbol{u}, \boldsymbol{x}, t) = 0$ always holds. Thus, it is approximated using a positive constant α:

$$\dot{\boldsymbol{F}}(\boldsymbol{U}, \boldsymbol{x}, t) = -\alpha \boldsymbol{F}(\boldsymbol{U}, \boldsymbol{x}, t). \tag{4}$$

Since the optimality error $\boldsymbol{F}$ converges to zero through evolution over time, $\boldsymbol{F} = 0$ can be satisfied. The equation Eq. (4) is treated as a system of linear equations concerning $\dot{\boldsymbol{U}}$, which is solved using the GMRES method. The resulting $\dot{\boldsymbol{U}}$ is then integrated to obtain the control input.

2.4. Combination of MCMPC and C/GMRES

In this study, we obtain a quasi-optimal solution by evaluating samples generated by adding random perturbations to the C/GMRES solution. The algorithm is presented in Algorithm 1. First, a single input $\boldsymbol{\mu}_{\mathrm{cg}}(t)$ is computed via the C/GMRES method. Next, $N-1$ random perturbations $\boldsymbol{\mu}_{\mathrm{p}}^{1:N-1}$ are generated from a normal distribution, and these are added to the input obtained by the C/GMRES method. Then, using both $\boldsymbol{\mu}_{\mathrm{cg}}(t)$ and $\boldsymbol{\mu}_{\mathrm{mc,0:H-1|t}}^{1:N-1}$, a total of N input samples $\mu_{k|t}^{1:N}$ are constructed. Based on these input samples, predicted trajectories $\boldsymbol{x}_{k|t}^{1:N}$ are computed, and the corresponding evaluation values $\boldsymbol{J}^{1:N}$ are evaluated using a evaluation function. Subsequently, resampling is performed using the $N-1$ input samples excluding the one with the worst evaluation. If the C/GMRES solution yields the best evaluation, it directly becomes the initial value for the next time step. Conversely, if a different sample is optimal, it initializes the subsequent C/GMRES iteration. The system state is then updated using the best-evaluated input. Fig. 1 illustrates the switching instance between multiple optimal trajectories, where the red solid line represents the optimal solution, the yellow solid line indicates the input obtained from the C/GMRES method, the blue solid lines correspond to the top 10 % of samples in terms of evaluation, and the gray solid lines represent the remaining samples. Even when the C/GMRES method results in a local optimum, our approach utilizes randomly perturbed samples to switch to a different optimum.

3. MPC FOR OBSTACLE AVOIDANCE

3.1. Evaluation function for optimal input calculation

As an evaluation function formulated for MPC, the terminal cost $\varphi(\boldsymbol{x}, t+T)$ and stage cost $L(\boldsymbol{x}, \boldsymbol{u}, \tau)$ of the proposed MCMPC are respectively defined as

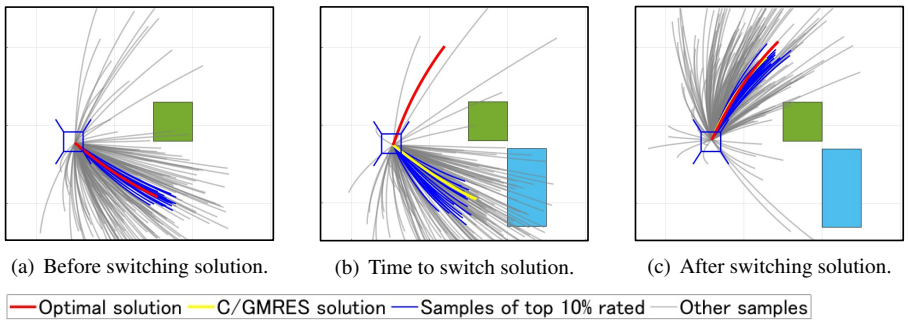
$$\begin{aligned} \varphi(\boldsymbol{x}, \cdot) =& \frac{1}{2}\tilde{\boldsymbol{x}}(\cdot)^\top \boldsymbol{P}\tilde{\boldsymbol{x}}(\cdot) + \frac{1}{2} W_{\mathrm{of}} \sum_{k=1}^{m} \rho(\boldsymbol{x}(\cdot)) \\ L(\boldsymbol{x}, \boldsymbol{u}, \tau) =& \frac{1}{2}\tilde{\boldsymbol{x}}(\tau)^\top \boldsymbol{Q}\tilde{\boldsymbol{x}}(\tau) + \frac{1}{2}\tilde{V}(\tau)^\top R\tilde{V}(\tau) \\ &+ \frac{1}{2} W_{\mathrm{o}} \sum_{k=1}^{m} \rho(\boldsymbol{x}(\tau)) + f_V(\boldsymbol{u}(\tau), \boldsymbol{u}(\tau-\Delta)), \end{aligned}$$

where $\tilde{\boldsymbol{x}}$ and $\tilde{V}(\tau)$ represent the deviation from the target state and velocity, respectively. $\boldsymbol{P}$, $\boldsymbol{Q}$ and R are positive definite constant used as weights for corresponding terms. The term $\rho(\boldsymbol{x})$ is the artificial potential field for avoiding m obstacles, with W_{o}, and W_{of} being its corresponding weights. To account for obstacle size, this study employs the elliptical potential field method [10]. Finally, f_V is a penalty term that increases the cost if the velocity change from the previous time step exceeds a threshold.

3.2. Evaluation functions for the C/GMRES method

In this study, the same terminal cost as defined in Section 3.1 was used for the C/GMRES method. However, a different stage cost was employed, as evaluating the input tends to yield better obstacle avoidance trajectories, and incorporating penalty terms is challenging in gradient-based methods. The stage cost used in the C/GMRES method is defined as

$$\begin{aligned} L(\boldsymbol{x}, \boldsymbol{u}, \tau) =& \frac{1}{2}\tilde{\boldsymbol{x}}(\tau)^\top \boldsymbol{Q}_{\mathrm{cg}}\tilde{\boldsymbol{x}}(\tau) + \frac{1}{2}\tilde{\boldsymbol{u}}(\tau)^\top \boldsymbol{R}_{\mathrm{cg}}\tilde{\boldsymbol{u}}(\tau) \\ &+ \frac{1}{2} W_{\mathrm{o,cg}} \sum_{k=1}^{m} \rho(\boldsymbol{x}(\tau)), \end{aligned} \tag{5}$$



(a) Before switching solution. (b) Time to switch solution. (c) After switching solution.

Fig. 1 The switch from a locally optimal solution to a globally optimal solution.

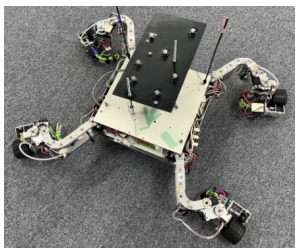


Fig. 2 Actual robot.

where $\tilde{\boldsymbol{u}}(\tau)$ represents the deviation from the target control input, while $\boldsymbol{Q}_{\mathrm{cg}}$, $\boldsymbol{R}_{\mathrm{cg}}$, $W_{\mathrm{o,cg}}$ are the weights for the corresponding terms.

3.3. Change in target values for C/GMRES method

To allow large deviation from the reference trajectory, we dynamically update the C/GMRES reference trajectory. At each time step, this update involves a selection process in which the existing reference is compared against a new trajectory projected from the previous step's optimal input. The superior trajectory, evaluated using the evaluation function J, is chosen as the new reference. Finally, a second-order filter is applied to this selected path to ensure its smoothness and continuity.

4. SIMULATION

This chapter first outlines the controlled object and simulation conditions. Subsequently, we conduct a comparative study of three methods to verify the effectiveness of our approach: the proposed method, standalone MCMPC, and the method of [23] augmented with the C/GMRES reference modification described in Section 3.3 (hereafter Mixed-MPC).

4.1. Control object

In this simulation, we verify the effectiveness of the proposed method by applying it to the obstacle avoidance control of a planar leg/wheel mobile robot, as shown in Fig. 2. For simplicity, only the pose of the robot body is considered as the controlled variable, while the joint angles and wheel positions are controlled using the method from [24]. The pose, consisting of the $x-y$ coordinates of the body center and its orientation angle, is defined as the state $\boldsymbol{p}_{\mathrm{b}}$, and its time derivative, the velocity $\boldsymbol{\mu}_{\mathrm{b}}$, is taken as the input. Consequently, the state equation is given by $\dot{\boldsymbol{p}}_{\mathrm{b}}(t) = \boldsymbol{\mu}_{\mathrm{b}}(t)$.

4.2. Condition

As the simulation conditions, the target trajectory was set as a straight line with $y = 0$, and obstacles were placed as shown in Fig. 3. The robot's sensor range was set to a 1.5 m radius, and no obstacles were recognized initially. Table 1 lists the parameters of the controlled system configured in the program: number of discretization steps, prediction horizon, number of samples for Mixed-MPC and the proposed method, number of samples for MCMPC, number of iterations for the GMRES method, damping coefficient and natural angular frequency of the second-order filter, the parameter α representing the weights for MPC, control period Δ, simulation duration, initial position, and target velocity.Additionally, the value indicating the convergence speed of the C/GMRES method equation is $\alpha = 10.0$, the weights of the stage cost and terminal cost of the elliptical potential are $W_{\mathrm{o}} = 0.9$ and $W_{\mathrm{of}} = 2.7$, and the weights for the stage cost and terminal cost of the elliptic potential in the C/GMRES method are set to $W_{\mathrm{o,cg}} = 0.036$ and $W_{\mathrm{of,cg}} = 0.108$, respectively.

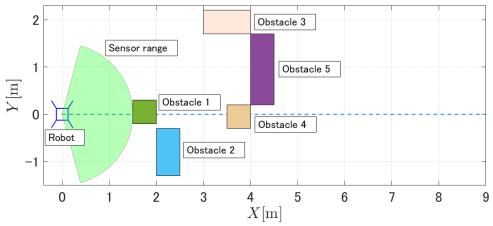


Fig. 3 Reference trajectory, robot, and obstacles.

Table 1 Parameters of planar model.

Parameter		Value
Number of divisions		30
Horizon		6.0
Number of samples		180
Number of MCMPC samples		3000
Number of GMRES iterations		3
Damping coefficient		1.0
Natural angular frequency		55.0
$\boldsymbol{Q}$		diag(30, 10, 3)
R		100
$\boldsymbol{P}$		diag(90, 30, 10)
$\boldsymbol{Q}_{\mathrm{cg}}$		diag(27, 27, 3)
$\boldsymbol{R}_{\mathrm{cg}}$		diag(50, 50, 5)
$\boldsymbol{P}_{\mathrm{cg}}$		diag(81, 81, 10)
Δ	[ms]	54
Simulation duration	[s]	40
Initial position	[m]	(0.0, 0.0)
Target velocity	[m/s]	0.25

4.3. Result

The results for MCMPC, Mixed-MPC, and the proposed method are presented in Fig. 4, Fig. 5, and Fig. 6, respectively. In these figures, (a) details the trajectories of the robot body and wheels, (b) the x- and y-

coordinates of the body, and (c) the control inputs. While all methods successfully avoided the obstacles, a comparison of their control inputs (c) reveals significant differences. As shown in Fig. 4(c), the MCMPC input was persistently oscillatory, whereas the Mixed-MPC's input, seen in Fig. 5(c), stabilized after an initial oscillatory period of approximately $26.7\,\mathrm{s}$. In contrast, the proposed method, illustrated in Fig. 6(c), maintained largely stable inputs with only slight oscillations punctuated by abrupt changes around $3.5\,\mathrm{s}$, $13.0\,\mathrm{s}$, and $24.5\,\mathrm{s}$.

4.4. Discussion

Figure 4, Fig. 5, and Fig. 6 compare the standalone sample-based method with the combined sample-based and gradient-based approaches. All methods successfully avoided the obstacles with no significant differences. A comparison of the control inputs reveals MCMPC produced the most fluctuated behavior. In contrast, both the proposed method and Mixed-MPC suppressed these oscillations to some extent. These results demonstrate the advantage of combining sample-based and gradient-based approaches for suppressing input vibrations.

Regarding the number of samples, the proposed method and Mixed-MPC required 180 samples, whereas MCMPC alone used 3000. These results confirm that the combined approach suppresses input oscillations even with a significantly smaller sample size. This effectiveness is attributed to the fact that the combination of sample-based and gradient-based approaches extensively utilizes the C/GMRES solution, which tends to locate near the true optimum. Incorporating this solution makes other samples more likely to cluster near the optimal solution during resampling.

Next, a comparison between Fig. 5 and Fig. 6 reveals that the proposed method better suppresses input oscillations. This improvement is attributed to the difference in sample generation. Mixed-MPC uses the C/GMRES solution as only one of its samples; consequently, if purely random samples are repeatedly selected, the control input can lose temporal consistency and exhibit large oscillations. In contrast, the proposed method generates all samples by adding random perturbations directly to the C/GMRES solution. Thus, the added noise can be seen as compensating for the C/GMRES solution's optimality error $\boldsymbol{F}$ at each time step. Because this solution is already near the optimum, the resulting input changes are minor, leading to smoother, more continuous behavior. For these reasons, the proposed method is more effective at suppressing input oscillations.

Finally, the intended effect of the proposed method is demonstrated. Figure 7 illustrates that when the robot is suddenly blocked, the distribution of noise-perturbed samples enables the system to explore alternative paths different from the one suggested by the C/GMRES solution. This exploration allows the system to find viable trajectory. Subsequently, as indicated by the state shown in Fig. 7(b), the C/GMRES method resumes its computation from this newly discovered path, successfully suppressing input oscillations. This process confirms that the proposed method achieves its objective of enabling continuous motion by navigating unforeseen obstacles.

5. EXPERIMENT

This section evaluates the performance with the proposed method implemented into the robot of Fig. 2.

5.1. Condition

The target trajectory was set to $y = 0$, and five obstacles were placed as in Section 4. Obstacle 1-2 were placed in the same locations as in Fig. 3, and Obstacle 3-5 were placed $1\,\mathrm{m}$ behind the locations in Fig. 3 in the positive direction of the x-axis. In addition, the detection range of the robot's sensors was set to a radius of $3\,\mathrm{m}$. Furthermore, the parameters set in the program are shown in Table 2. Additionally, the values indicating the convergence speed of the C/GMRES method equations, the elliptical potential weights, and the elliptical potential weights of the C/GMRES method are set to $\alpha = 10.0$, $W_{\mathrm{o}} = 0.9$, $W_{\mathrm{of}} = 2.7$, $W_{\mathrm{o,cg}} = 0.036$, and $W_{\mathrm{of,cg}} = 0.108$. In this experiment, the visual feedback system described in [23] was used. Parameters and experimental conditions were modified from the simulation in order to reproduce the stack.

Table 2 Parameters of planar model.

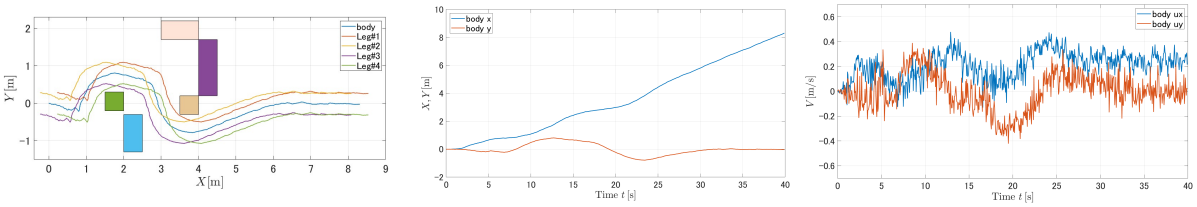
Parameter		Value
Number of divisions		30
Horizon		6.0
Number of samples		180
Number of GMRES iterations		3
Damping coefficient		1.0
Natural angular frequency		50.0
$\boldsymbol{Q}$		diag(15, 12, 3)
R		100
$\boldsymbol{P}$		diag(45, 36, 10)
$\boldsymbol{Q}_{\mathrm{cg}}$		diag(27, 27, 3)
$\boldsymbol{R}_{\mathrm{cg}}$		diag(50, 50, 5)
$\boldsymbol{P}_{\mathrm{cg}}$		diag(81, 81, 10)
Δ	[ms]	54
Initial position	[m]	(0.0, 0.0)
Target velocity	[m/s]	0.25

5.2. Result and discussion

Figure 8 shows the results of the actual experiment, where (a) shows the path traveled by the vehicle body and wheels, (b) shows the position of the vehicle body in the x and y directions, and (c) shows the input in the x and y directions of the vehicle body. Figure 9 shows the robot's posture and the sample's state at each time point, which confirmed that obstacle avoidance was achieved.

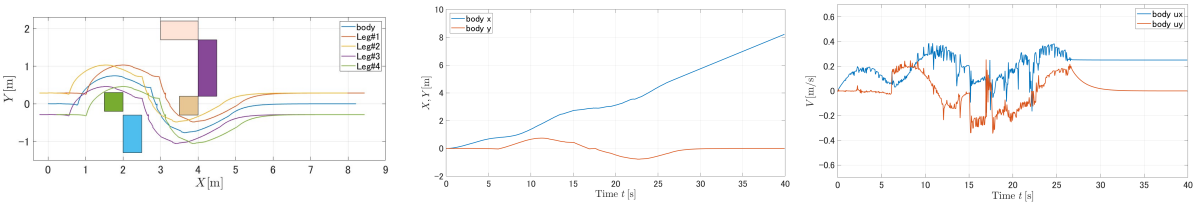
Figure 8 depicts the practical effectiveness of the proposed method on an actual robot, which successfully suppressed input vibrations while avoiding obstacles, corroborating the simulation results from Chapter 4.

Furthermore, as shown in Fig. 9(a), Fig. 9(b), and Fig. 9(c), at $4.48\,\mathrm{s}$, the optimal solution and the



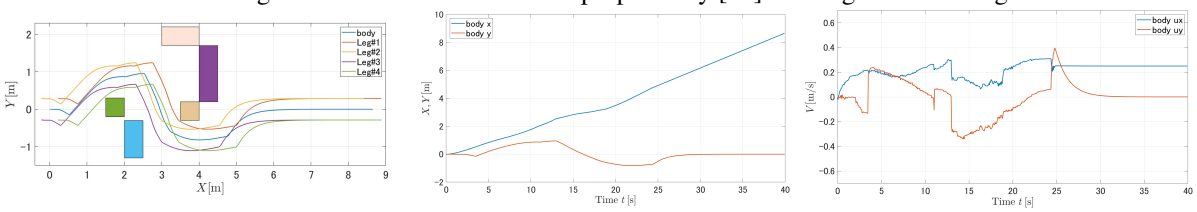
(a) Path of body and wheel. (b) Position of body. (c) Input.

Fig. 4 MCMPC.



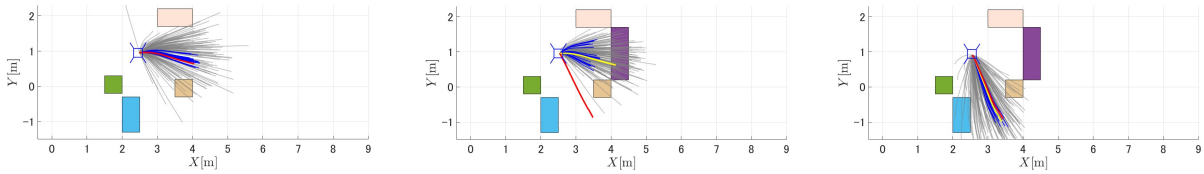
(a) Path of body and wheel. (b) Position of body. (c) Input.

Fig. 5 Combination of methods proposed by [23] and target value changes.



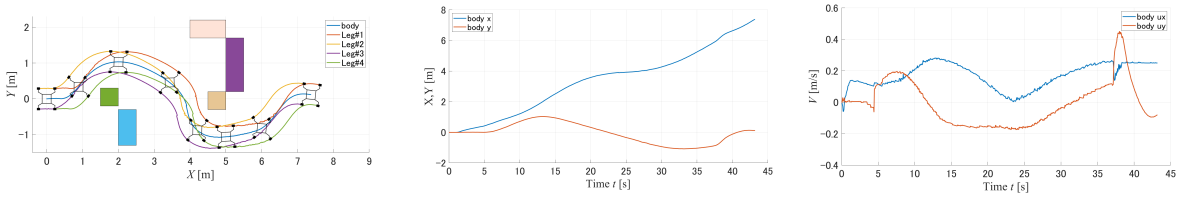
(a) Path of body and wheel. (b) Position of body. (c) Input.

Fig. 6 Proposed method.



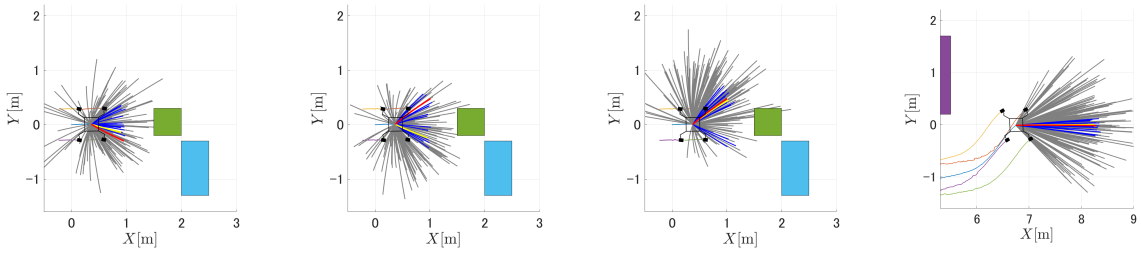
(a) t=12.91 s. (b) t=13.07 s. (c) t=13.23 s.

Fig. 7 Switching solutions in a simulation



(a) Path of body and wheel. (b) Position of body. (c) Input.

Fig. 8 Results of the experiment.



(a) t=4.48 s. (b) t=4.53 s. (c) t=4.64 s. (d) t=40.66 s.

Fig. 9 Images of robot posture, wheels, and samples at each time.

C/GMRES method solution are attempting to avoid Obstacle 1 from the negative y-axis direction. At 4.53 s, the optimal solution is avoiding Obstacle 1 in the positive direction of the y-axis, while the C/GMRES method solution is avoiding it in the negative direction of the y-axis. Subsequently, both solutions obtain solutions that avoid Obstacle 1 in the positive direction of the y-axis. This behavior is attributed to the fact that, similar to the simulation, adding random numbers to the C/GMRES method enabled the exploration of a global optimal solution.

Additionally, from Fig. 8(a) and Fig. 8(b), a slight overshoot of the target path occurs around $x = 6.8\,\mathrm{m}$, attributed to a tracking error in the separate optimization routine that calculates the joint angles based on the velocity command from the proposed method. Although the velocity command is accurate for following the path, as shown in Fig. 9(d), the subsequent joint angle optimization introduces a transient error, causing this deviation.

6. CONCLUSION

This study proposes a method for generating MCMPC samples based on the solution of the C/GMRES method. It validates its effectiveness through comparisons with a previously developed approach and its implementation on an actual robot. The previous method had the issue that when the C/GMRES solution was not selected, the control inputs exhibited excessive chattering. In contrast, the proposed method generates MCMPC samples based on the C/GMRES solution, successfully suppressing input chattering while enabling obstacle avoidance, and its performance was demonstrated on an actual robot. This work was partially supported by JSPS KAKENHI Grant Number JP25K01255.

REFERENCES

[1] K. Otsu et al, "Fast approximate clearance evaluation for rovers with articulated suspension systems", Journal of Field Robotics, 37(5), 2020, pp 768-785.

[2] R. Dodge et al, "Dynamics associated with the Corer on M2020 Perseverance Rover", 2021 IEEE Aerospace Conference, 2021, pp. 1-13.

[3] A. Rankin et al, "Driving Curiosity: Mars Rover Mobility Trends During the First Seven Years", 2020 IEEE Aerospace Conference, 2020, pp. 1-19.

[4] B. H. Wilcox, "ATHLETE: A cargo and habitat transporter for the moon", 2009 IEEE Aerospace conference, Big Sky, MT, USA, 2009, pp. 1-7.

[5] T. Yoshioka et al, "Hybrid Locomotion of Leg-Wheel ASTERISK H", J. Robot. Mechatron., Vol.20 No.3, 2008, pp. 403-412.

[6] M. Yakubu et al., "A Novel Mobility Concept for Terrestrial Wheel-Legged Lunar Rover," in IEEE Access, vol. 13, 2025, pp. 15618-15638.

[7] W. Reid et al, "Actively articulated suspension for a wheel-on-leg rover operating on a Martian analog surface", 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 5596-5602.

[8] J. He et al, "Model Predictive Control of a Novel Wheeled–Legged Planetary Rover for Trajectory Tracking", Sensors 2022, 22, 4164.

[9] B. Wang et al, "Parallel Structure of Six Wheel-legged Robot Model Predictive Tracking Control based on Dynamic Model", 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 5143-5148.

[10] N. Takahashi et al. "Optimal Configuration Control of Planar Leg/Wheel Mobile Robots for Flexible Obstacle Avoidance", Control Engineering Practice, Vol.101, 2020, pp.1-10.

[11] Y. Hagimori et al, "Model predictive control for leg/wheel mobile robots using partitioned model", Transactions of the JSME, 82(842), 16-00144-16–00144, 2016.

[12] S. Ohyama et al, "Parallelized nonlinear model predictive control on GPU", 2017 11th Asian Control Conference (ASCC), 2017, pp. 1620-1625.

[13] G. Williams et al, "Aggressive driving with model predictive path integral control", 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 1433-1440.

[14] S. Nakatani et al, "Swing up Control of Inverted Pendulum on a Cart with Collision by Monte Carlo Model Predictive Control", 2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), 2019, pp. 1050-1055.

[15] A. Muraleedharan et al, "Real-Time Implementation of Randomized Model Predictive Control for Autonomous Driving", in IEEE Transactions on Intelligent Vehicles, Vol. 7, No. 1, 2022, pp. 11-20.

[16] S. Satoh et al, "An Iterative Method for Nonlinear Stochastic Optimal Control Based on Path Integrals", in IEEE Transactions on Automatic Control, vol. 62, no. 1, Jan. 2017, pp. 262-276.

[17] G. Williams et al, "Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving", in IEEE Transactions on Robotics, vol. 34, no. 6, Dec. 2018, pp. 1603-1622.

[18] F. Rozzi et al. "Combining Sampling-and Gradient-based Planning for Contact-rich Manipulation", arXiv preprint arXiv:2310.04822 (2023).

[19] J. Park et al, "Robot Navigation with Model Predictive Equilibrium Point Control", IEEE International Conference on Intelligent Robots and System, 2012, pp. 4945 - 4952.

[20] C. Rosmann et al, "Planning of Multiple Robot Trajectories in Distinctive Topologies", IEEE European Conference on Mobile Robots, 2015.

[21] T. Ohtsuka, "A continuation/GMRES method for fast computation of nonlinear receding horizon control", Automatica, Volume 40, Issue 4, 2004, Pages 563-574.

[22] T. Ohtsuka, "A tutorial on C/GMRES and automatic code generation for nonlinear model predictive control", 2015 European Control Conference (ECC), 2015, pp. 73-86.

[23] T. Onizawa et al, "Model predictive obstacle avoidance for a leg/wheel mobile robot utilizing sample-based optimization", IECON 2024 - 50th Annual Conference of the IEEE Industrial Electronics Society, 2024, pp. 1-8.

[24] K. Nonaka et al, "Velocity Estimation for Mobile Robots Using Caster Odometer and Nonlinear Observer", Trans. of The Japan Society of Mechanical Engineers Series C, 78(791), 2012, pp2509–2525.