Preprints of the 22nd IFAC World Congress
Yokohama, Japan, July 9-14, 2023

Self-driving Electric Wheelchair in Crowded Environments Using a Fuzzy Potential Model Predictive Control ⋆

Eisuke Kawaguchi * **Kazuma Sekiguchi** * **Kenichiro Nonaka** *

* *Tokyo City University, 1-28-1 Tamazutsumi, Setagaya, Tokyo, Japan (e-mail: g2281018@tcu.ac.jp, ksekiguc@tcu.ac.jp, knonaka@tcu.ac.jp)*

Abstract: In the recent development of small autonomous vehicles, such as electric wheelchairs, one of the challenging environments to be operated is a congested area in which vehicles are mixed with pedestrians. In addition to prompt avoidance in response to the surrounding non-stationary environment, optimal maneuvering must be conducted between the multiple feasible trajectories. In the present study, the fuzzy potential method is combined with model predictive control to avoid obstacles based on predictions of future behavior. Furthermore, the global search for a solution and flexible switching of avoidance paths were achieved utilizing the Monte Carlo optimization in model predictive control. This also enables the optimization of a nonlinear and discontinuous evaluation function including membership functions for fuzzy inference. The method's effectiveness was verified through simulations and real-time experiments for a self-driving electric wheelchair equipped with a LiDAR, and verification confirmed flexible obstacle avoidance.

Keywords: Model predictive control, Self-driving wheelchair, Obstacle avoidance, Fuzzy potential method, Monte Carlo optimization

1. INTRODUCTION

In recent years, autonomous vehicles have been used for transporting goods and assisting human mobility. Among autonomous vehicles, the unique driving environment for small autonomous vehicles, such as electric wheelchairs, includes the presence of multiple dynamic obstacles, such as pedestrians, in a building and narrow passageways due to walls and other objects. Autonomous driving in such an environment requires safe avoidance of unknown and dynamic obstacles in addition to the walls included in the map. Various route planning methods have been proposed for autonomous vehicles. A simple method is the artificial potential field method, which uses potential fields to define attraction and repulsion (Warren, 1990; Zhu et al., 2006). Methods that consider vehicle dynamics include the dynamic window approach (Fox et al., 1997) and the follow the gap method (Sezer and Gokasan, 2012). By considering the vehicle dynamics, it is possible to limit the output of the controller to the inputs that are feasible for the control target, which leads to improved control performance. These methods have been applied to autonomous vehicles (Sezer, 2022; Li et al., 2017). Model predictive control (MPC) is a method for predicting a dynamically feasible trajectory based on a model and calculating the optimal control inputs while predicting the behavior up to a finite time in the future. By optimizing the future behavior, MPC is expected to achieve high control performance, and there are several implementations to small autonomous vehicles (Bardaro et al., 2018; Matsuura et al., 2022; Cervolo et al., 2017; Shibata et al., 2018). The application of MPC to vehicles in crowded environments has been attempted based on stochastic model predictive control. Shibata et al. (2019) applied stochastic model predictive control to the control of an electric wheelchair. The expectation of relative velocity to obstacles was suppressed so that when the risk of collision was high, the speed was reduced in order to achieve a safe maneuver. However, since optimization is based on the gradient method, there is the possibility of convergence to locally optimal solutions when multimodalities occur in the evaluation function. Maki et al. (2022) used stochastic model predictive control to create constraints based on probabilities in an automobile and made it possible to drive in a congested environment by relaxing the constraints and slowing down. In their study, a control method that can reactively respond to changes in the conditions of surrounding obstacles was investigated, implicitly assuming the unimodality in the optimization problem.

The fuzzy potential method (FPM)(Suzuki and Takahashi, 2009, 2011) has been proposed as a path planning method based on the geometric relationship between the control target and the surrounding environment. In this method, the control input is calculated by creating a potential membership function using the location of the target point, the location of obstacles, and the state of the control target, and then performing fuzzy inferences. The obstacle position can be directly used in the FPM from the relative position measured by a 2D-LiDAR. The FPM enables obstacle avoidance based on obstacle information detected online but does not take into account the dynamics of the robot and may calculate inputs that would make collisions

⋆ The present study was supported by JSPS KAKENHI Grant Number JP 22K19801.

with obstacles unavoidable for a real robot. In order to address this problem, Nagata et al. (2014) combined the FPM and MPC to account for dynamics. In their study, an omni-directional mobile robot was able to avoid obstacles, but the obstacles were limited to stationary obstacles. In their method, the FPM is used to calculate the reference path, and MPC is used to calculate the control input to track the reference. Therefore, the available inputs are limited, and the method is not suitable for dynamic environments.

In the present study, the potential membership function (PMF) is integrated as part of the MPC evaluation function to calculate the input without limiting the solution. The gradient method is widely used for optimization, but since PMFs are generally discontinuous functions, optimization is difficult. In addition, the evaluation function may be multimodal in a crowded dynamic environment. Therefore, measures for convergence to a local solution are also necessary. In this study, the optimization is performed using the Monte Carlo method that resolves both the discontinuity of the PMF and the multimodality of the optimal trajectory. Although the number of samples must be large to bring the solution closer to the exact solution, we took advantage of parallelization to speed up the algorithm. With recent improvements in CPU performance and multithreading, the computational speed of Monte Carlo optimization is high. In fact, we show that the optimization can be performed in real-time.

2. CONTROLLED OBJECT

In the present study, obstacle detection and localization are performed using measurement obtained by light detection and ranging (LiDAR) sensor VLP-16 shown in Figure 1. The controlled object is the electric wheelchair shown in Figure 2, which is characterized by rear wheels that can be driven independently of the left and right wheels, and front wheels that can move in all directions. The wheelchair is equipped with rotary encoders that measure the amount of rotation of the left and right rear wheels. It is also possible to provide control input from an external source.

Figure 3 shows a geometric model of an electric wheelchair. The state of the electric wheelchair is defined as $\boldsymbol{x} = [\, x \; y \; \theta \,]^{\mathrm{T}}$, where x and y are the center coordinates between the rear wheels, and θ is the heading angle. Moreover, the input is defined as $\boldsymbol{u} = [\, V \; \omega \,]^{\mathrm{T}}$ where V and ω represent the translational and angular velocities, respectively. Assuming that the wheels do not slip, the state equation of the discrete system is as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \begin{bmatrix} \cos\theta_k & 0 \\ \sin\theta_k & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u}_k \Delta, \tag{1}$$

where Δ denotes the discrete-time interval.

3. METHODS

3.1 Fuzzy potential method (FPM)

The FPM (Suzuki and Takahashi, 2009, 2011) is a method for obstacle avoidance that takes into account the geometric positional relationship between the control target and

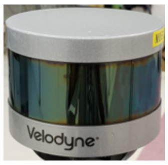


Fig. 1. LiDAR

Fig. 2. Electric wheelchair

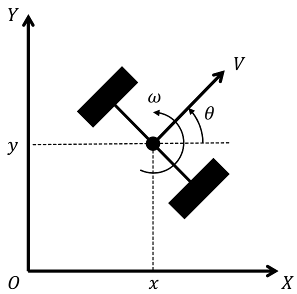


Fig. 3. The kinematic model of a wheelchair

the obstacles. As an example, the PMF shown in Figure 4 is designed based on the attitude angle of the control target, the size and position of the obstacle, and the target position in the situation shown in Figure 4(a). The target direction is determined based on the designed PMFs. The function shown in Figure 4(b) is designed by combining the PMF for the target point shown in Figure 4(c) and the PMF for the obstacle shown in Figure 4(d), where $\varphi_{\mathrm{r,g}}$ is the attitude angle toward the target point, and $\varphi_{\mathrm{r,o}}$ is the attitude angle toward the obstacles. The PMF for the target point is set so that the grade is largest in the $\varphi_{\mathrm{r,g}}$ direction and smallest in the $\varphi_{\mathrm{r,g}} - \pi$ direction. The parameters are defined as follows:

$$g_{\mathrm{a}} = 1, \tag{2}$$

$$g_{\mathrm{b}} = \eta g_{\mathrm{a}} \quad (0 < \eta < 1), \tag{3}$$

where η is a parameter that determines how much to decrease the grade at the opposite position of the target direction. The PMF for obstacles is set to lower the grade near $\varphi_{\mathrm{r,o}}$ based on $\varphi_{\mathrm{r,o}}$, the radius of the target, and the radius of the obstacle. φ is determined by the following equation:

$$\varphi = \begin{cases} \sin^{-1}\left(\dfrac{D}{\| r_{\mathrm{r,o}} \|}\right) & \text{if} \quad D < \| r_{\mathrm{r,o}} \| \\ \pi - \sin^{-1}\left(\dfrac{\| r_{\mathrm{r,o}} \| - d_{\mathrm{s}}}{D - d_{\mathrm{s}}}\right) & \text{otherwise} \end{cases}, \tag{4}$$

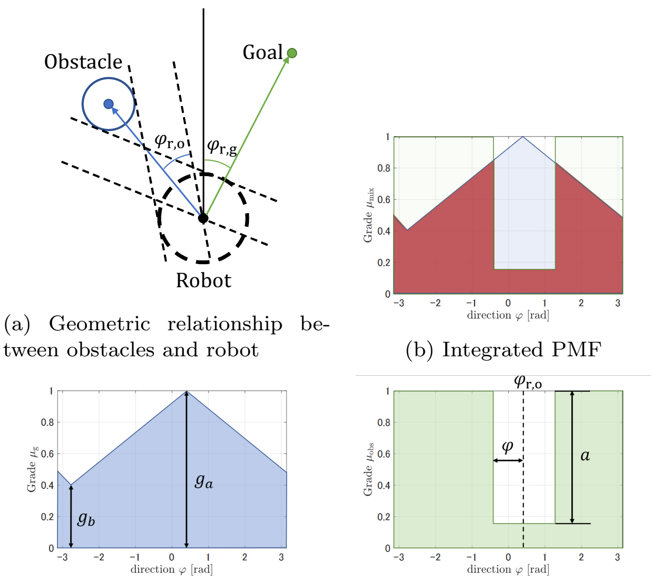
where $r_{\mathrm{r,o}}$ is the distance between the robot and the obstacle. Moreover, D is defined as follows:

$$D = r_{\mathrm{veh}} + r_{\mathrm{obs}} + d_{\mathrm{s}}, \tag{5}$$

where r_{veh} is the radius of a cylinder enveloping the robot, r_{obs} is the radius of the obstacle, and d_{s} is the margin.

3.2 Model predictive control (MPC)

MPC is an optimal control method that predicts and optimizes the future behavior of a controlled object from



(a) Geometric relationship between obstacles and robot

(b) Integrated PMF

(c) PMF for the goal

(d) PMF for the obstacles

Fig. 4. Example of potential membership functions

the current time to a finite time in the future based on a numerical model. The optimal input at the current time is determined by solving an optimization problem at each sampling time. The advantages of MPC include its ability to explicitly consider constraints and to adapt to multi-input, multi-output systems.

Now, consider the nonlinear state equation of the controlled object described as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{6}$$

with the state $\boldsymbol{x}_k$ and the control input $\boldsymbol{u}_k$. The inequality constraints and equality constraints are defined as follows:

$$\boldsymbol{g}_i(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq \boldsymbol{0}(i = 1, 2, ..., N), \tag{7}$$

$$\boldsymbol{h}_j(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{0}(j = 1, 2, ..., M), \tag{8}$$

where M and N are numbers for each constraint. Then, the evaluation function J can be expressed as follows:

$$J = \phi(\boldsymbol{x}_H) + \sum_{k=0}^{H-1} L(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{9}$$

where H is the number of the predictive horizon and $\phi(\cdot)$ is the terminal cost, and $L(\cdot)$ is the stage cost. The MPC optimization problem can be expressed as follows:

Problem: General MPC

$$\begin{aligned} \text{minimize} \quad & J \\ \text{with respect to} \quad & \boldsymbol{u}_k (k = 0, 1, ..., H \quad 1) \\ \text{subject to} \quad & (6)(7)(8) \end{aligned}$$

In this optimization problem, the decision variable $\boldsymbol{u}_k$ is computed such that the evaluation function J is minimized. Of the decision variables obtained here, the input corresponding to the current time is applied to the controlled object.

4. FPMPC USING MONTE CARLO OPTIMIZATION

4.1 Fuzzy potential model predictive control

In this section, fuzzy potential model predictive control (FPMPC), which incorporates the PMF in the evaluation function of MPC, is described. Let $\mu_{\mathrm{mix}}(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{obs},k+1}, \varphi)$ represent the grade of the PMF considering obstacles and the target point computed by the FPM, as shown in Figure 4. This PMF is designed based on the state of the controlled object in step k and the obstacle position $\boldsymbol{x}_{\mathrm{obs}}$ in step $k+1$. Since the PMF is described in the coordinate system of the controlled object, the grade at step $\varphi = 0$ is used as the evaluation value. The position of the obstacle at step $k+1$ is used to account for the movement of the obstacle. Then, $\phi(\cdot)$ and $L(\cdot)$ are designed as follows:

$$\phi(\boldsymbol{x}_H) = Q_{\mathrm{f}}(1 - \mu_{\mathrm{mix}}(\boldsymbol{x}_H, \boldsymbol{x}_{\mathrm{obs},H+1}, \varphi)), \tag{10}$$

$$\begin{aligned} L(\boldsymbol{x}_k, \boldsymbol{u}_k) =& Q(1 - \mu_{\mathrm{mix}}(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{obs},n+1}, \varphi)) \\ &+ \tilde{\boldsymbol{u}}_k^{\mathrm{T}} \boldsymbol{R} \tilde{\boldsymbol{u}}_k + W_{\mathrm{obs}} \frac{(\boldsymbol{V}_n)^2}{\mid d_{\mathrm{obs}} + \epsilon \mid}, \end{aligned} \tag{11}$$

where $\tilde{\boldsymbol{u}}_k$ is the deviation of the input from the pre-defined target input $\boldsymbol{u}_{\mathrm{ref}}$, Q_{f}, and Q are weights for the evaluation of the PMF, and $\boldsymbol{R}$ is the weight for suppressing the input. Moreover, W_{obs} is the weight for the term used to decelerate when an obstacle is approaching. Moreover, $\boldsymbol{V}_k$ is the input sequence of velocity, d_{obs} is the distance to the nearest obstacle, and ϵ is a small positive constant. The term for deceleration enables the system to decelerate, stop, or retreat when an obstacle approaches the controlled object. ϵ prevents the evaluation value from becoming too large when the obstacle approaches in immediate proximity.

4.2 Monte Carlo optimization

We use Monte Carlo optimization method to obtain the optimal solution for FPMPC. Since Monte Carlo optimization is sampling-based, it can optimize the discontinuous evaluation function of FPMPC. Optimization is performed by generating randomized input sequences, computing predictive paths, evaluating each predictive path, and resampling. In order to generate an input sequence, first define an input sequence as follows:

$$\boldsymbol{u}^i = \{\boldsymbol{u}_0^i, \boldsymbol{u}_1^i, \boldsymbol{u}_2^i, \ldots, \boldsymbol{u}_H^i\}, \tag{12}$$

Where i denotes the particle number. The input sequence $\boldsymbol{u}^i$ is generated as follows using $\boldsymbol{u}_{\mathrm{r}}^i$ obtained by resampling as described below:

$$\boldsymbol{u}^i = \boldsymbol{u}_{\mathrm{r}}^i + \boldsymbol{w}^i, \tag{13}$$

where $\boldsymbol{w}^i$ is a random signal obeying a normal distribution.

Next, the predicted path is calculated as shown in Figure 5 by substituting the input sequence into the kinematic model of the controlled object as follows:

$$\boldsymbol{x}_{k+1}^i = \boldsymbol{x}_k^i + \begin{bmatrix} V_k^i \cos\theta \\ V_k^i \sin\theta \\ \omega_k^i \end{bmatrix} \Delta, \tag{14}$$

where Δ is the discrete-time interval, and $\boldsymbol{x}_n^i$ is the state and V_n^i and ω_n^i are the inputs at step n of the i-th sample, respectively. The initial state of the predicted path is set to be the state at the current time, as follows:

$$\boldsymbol{x}_0^i = \boldsymbol{x}, \tag{15}$$

The evaluation function of FPMPC is as follows:

$$J^i = \phi(x_H^i) + \sum_{n=0}^{H-1} L(x_k^i, u_k^i), \tag{16}$$

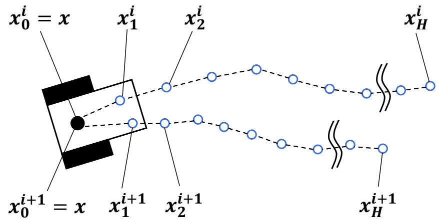


Fig. 5. Calculation of predicted paths

and for each predicted path, the evaluation value is calculated based on (16). To prevent collision with the wall, the upper and lower limits $\overline{\boldsymbol{x}}$, $\overline{\boldsymbol{y}}$ and $\underline{\boldsymbol{x}}$, $\underline{\boldsymbol{y}}$ of the position of x and y were set in advance according to the position of the wall. The positions to which the constraints were applied were calculated at four points for $\boldsymbol{x}_{\mathrm{v}}$ and $\boldsymbol{y}_{\mathrm{v}}$ in the vertex of the rectangular approximation of the control target, and the constraints were applied to all four points. Figure 6 shows an example of position constraints as the MPC step time changes. Only the constraint for $j = 1$ is shown. In addition, an upper limit $V_{\max}$ of the speed V was set. The set constraint conditions are as follows:

$$\underline{x}_k < x_{\mathrm{v},k}^j < \overline{x}_k \quad \text{for } j = 1, 2, 3, 4 \tag{17}$$

$$\underline{y}_k < y_{\mathrm{v},k}^j < \overline{y}_k \quad \text{for } j = 1, 2, 3, 4 \tag{18}$$

$$V < V_{\max} \tag{19}$$

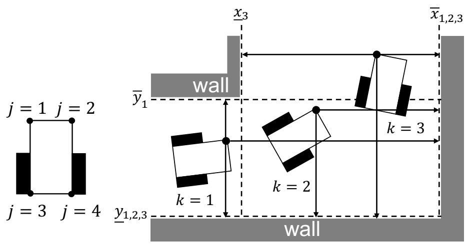


Fig. 6. Example of positional constraints

Then, Monte Carlo optimization is summarized below:

Problem: FPMPC using Monte Carlo optimization

$$\begin{aligned} \text{minimize} \quad & J^i \\ \text{with respect to} \quad & V, \omega \\ \text{subject to} \quad & (14)(15)(17)(18)(19). \end{aligned}$$

Finally, the input sequence $\boldsymbol{u}_{\mathrm{r}}^i$ is updated at the resampling step based on low-variance sampling (Thrun, 2002), in which the weights are the inverse of the evaluation values. This makes it possible to search for more paths in the neighborhood of paths with superior evaluation values at the next time, and to reduce the number of paths in the neighborhood of paths with poor evaluation values.

5. SIMULATION

5.1 Conditions

Figure 7 shows the locations of the obstacles and the simulation environment. The simulation was conducted assuming a crowded environment with pedestrians in a corridor. The deceleration of the electric wheelchair when it cannot proceed due to obstacles was also verified. Obstacles in blue and red represent dynamic and static obstacles, respectively. Table 1 shows the parameters used in the simulation.

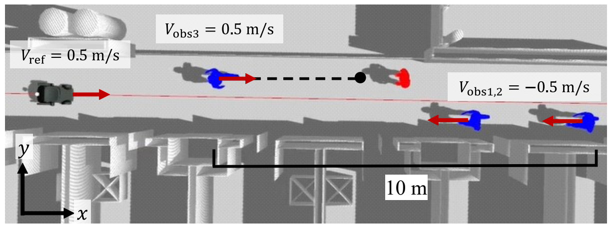


Fig. 7. Initial position and conditions of simulation

Table 1. Parameter setting in simulation

Parameter	Value
Horizon	21
Predictive step [s]	0.5
Number of predictive paths	1,000
Target velocity [m/s]	0.5
Target angular velocity [rad/s]	0

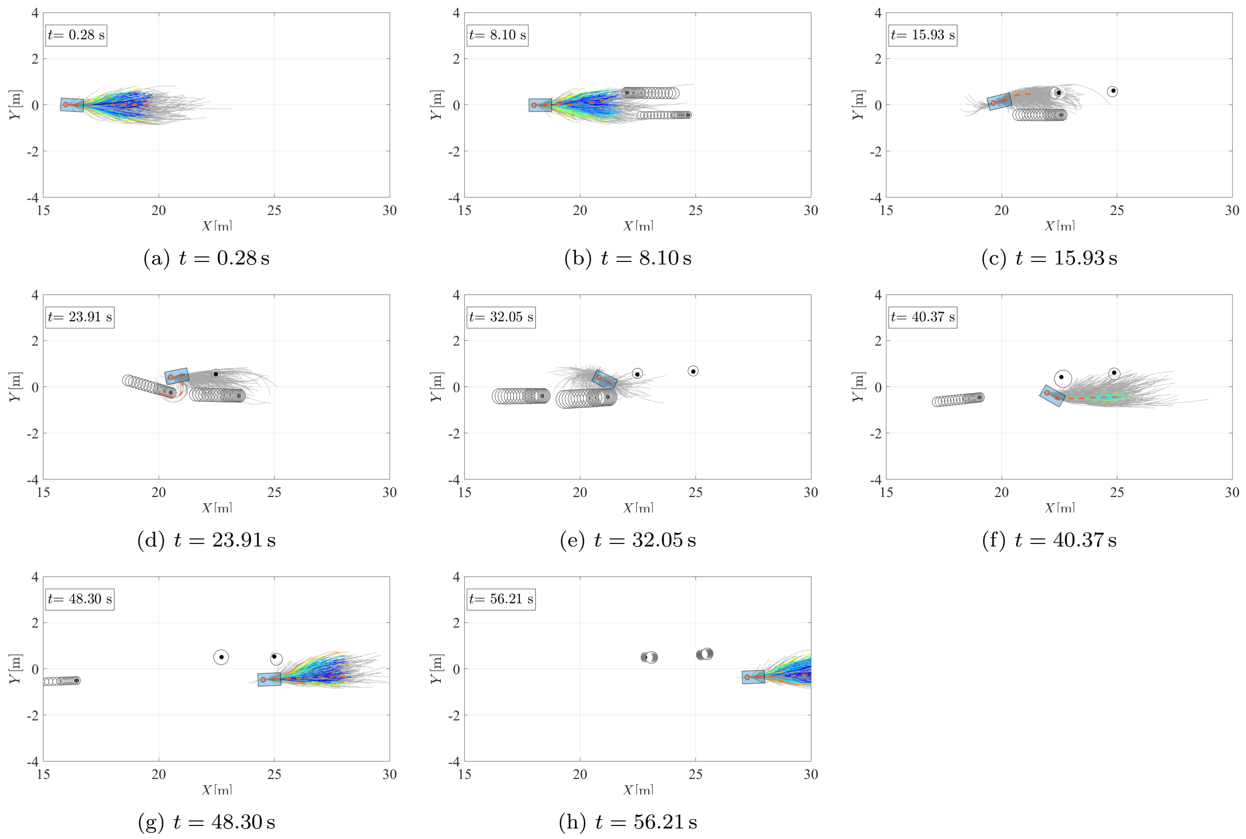
5.2 Results and discussion

Figure 8 shows how the electric wheelchair avoids pedestrians. Figure 9 shows the control inputs and states of the electric wheelchair. The results for the input in Figure 9(a) show both velocity and angular velocity. The state results in Figure 9(b) show the true value of the position obtained from the simulator and estimated state value obtained by localization, respectively. Figures 8(b), 8(c), and 9 show that the wheelchair decelerates in response to changes in the speed of the dynamic obstacle. Figure 8(e) also shows that there is a candidate path behind the controlled object. This caused by the fact that the sample distribution is updated by resampling in Monte Carlo optimization to generate more samples in the area where the evaluation value is expected to be higher. In this case, it is considered that decelerating or backing up is judged to have a better evaluation. In Figures 8(f) through 8(h), the wheelchair is considered to have moved forward again when an obstacle passes by and the wheelchair is judged to be able to proceed. Thus, the wheelchair decides whether to yield the path or move forward depending on the surrounding obstacles.

6. EXPERIMENT

6.1 Experimental system and estimation

Figure 10 shows the configuration of the experimental system. The experimental system is roughly divided into an electric wheelchair and a PC for computation. LiDAR obtains environmental information for localization, Raspberry Pi handles communication between the electric wheelchair and the PC, MATLAB calculates control inputs, and Autoware performs localization. The state, control inputs, and sensor data are communicated via Robot Operating System (ROS).

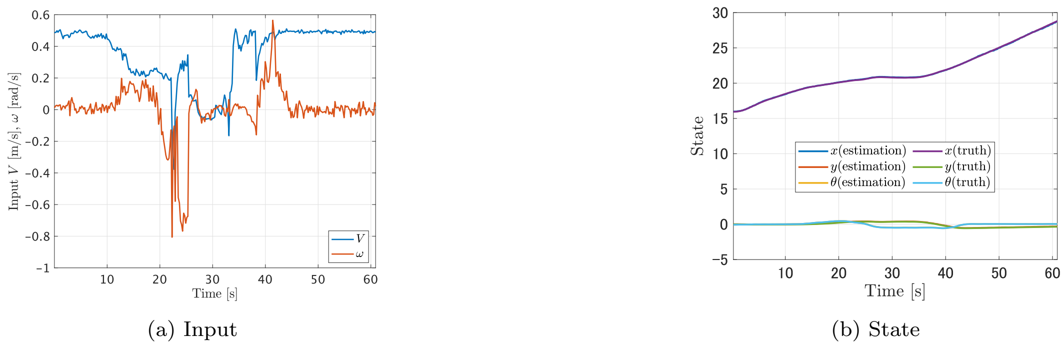


(a) $t = 0.28\,\mathrm{s}$ (b) $t = 8.10\,\mathrm{s}$ (c) $t = 15.93\,\mathrm{s}$

(d) $t = 23.91\,\mathrm{s}$ (e) $t = 32.05\,\mathrm{s}$ (f) $t = 40.37\,\mathrm{s}$

(g) $t = 48.30\,\mathrm{s}$ (h) $t = 56.21\,\mathrm{s}$

Fig. 8. Snapshots of the electric wheelchair and obstacles in the simulation. The orange circle indicates the position of the electric wheelchair. The black circle indicates the obstacle and its predicted position, and the lines drawn from the electric wheelchair indicate each predicted path. The color of the predicted paths indicates the magnitude of the evaluation value.



(a) Input (b) State

Fig. 9. Simulation results

Localization is based on map-matching, a method of obtaining the self-position by matching a pre-created three-dimensional point cloud map with point cloud data obtained from LiDAR. The two point clouds are matched using the normal distribution transform (NDT) algorithm (Biber and Strasser, 2003), which is computationally fast. The point cloud map was created using NDT-SLAM, a method for estimating self-position while creating a point cloud map by overlapping point clouds using NDT.

In order to detect obstacles, the position and size of obstacles were estimated using LiDAR point cloud data. The obstacle location and size were computed from the obstacle point cloud obtained by ground removal and clustering from the point cloud data. The computed obstacle positions are input to the joint probabilistic data association filter (JPDAF)(Bar-Shalom et al., 2009) in order to estimate the position and velocity of the obstacle. The JPDAF is a probabilistic method of data association and can track multiple targets.

6.2 Conditions

Figure 11 shows the positioning of pedestrians, the setting of the coordinate axes, and the width of the corridor in the experiment. In this experiment, a pedestrian approaching from the front crossed in front of the controlled object in a 2.5 m-wide corridor. The pedestrians behind the

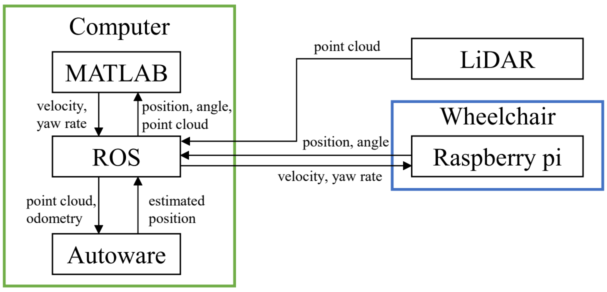


Fig. 10. Experimental system

control target filmed and monitored the experiment while maintaining a certain distance from the control target. Table 2 shows the parameters used in the experiment.

Table 2. Parameter setting in experiment

Parameter	Value
Horizon	21
Predictive step [s]	0.5
Number of predictive paths	1,000
Target velocity [m/s]	0.5
Target angular velocity [rad/s]	0
Weight Q_{f}, Q	1
Weight $\boldsymbol{R}$	diag(1.5,0.9)
Weight W_{obs}	0.05

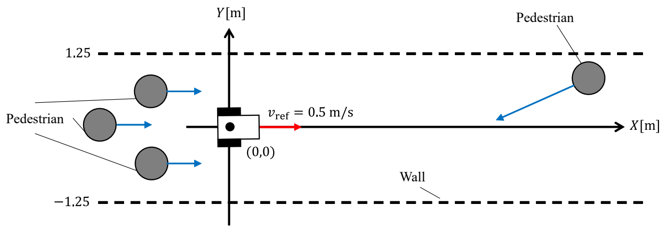


Fig. 11. Initial position and conditions of the experiment

6.3 Results and discussion

Figure 12 shows how the electric wheelchair avoids pedestrians. Figure 13 shows the control input and state of the electric wheelchair, the calculation cycle, and the evaluation value. From Figure 12, the system tracks obstacles in the surrounding area and estimates the future position of the obstacle based on the velocity estimation. The calculated obstacle positions are used to compare the candidates for the paths for the controlled object, which are created based on the kinematic model. Therefore, when a pedestrian moves in front of the wheelchair, the system predicts in advance that the pedestrian will pass from the right side of the control target, and avoids the obstacle by moving to the left. Although there are areas where the predicted positions of the pedestrian behind the wheelchair overlap those of the control target, comparison of the predicted positions of the obstacle and the control target at the same time indicates that there is no deterioration in the evaluation value. Figures 13(a) and 13(d) show that the vehicle decelerates when the obstacle is avoided at approximately 20 seconds. This can be attributed to the fact that the MPC evaluation function deteriorates when the speed of the control target is high and an obstacle is approaching and that the evaluation deteriorates when the control input is high. Figure 13(c) shows that, except for the initial time, the computation period was within 200 ms and was 75 ms on average for the controller only and 107 ms on average for the entire system.

7. CONCLUSION

The present paper proposes fuzzy potential model predictive control using Monte Carlo optimization as a method to enable electric wheelchairs to travel flexibly in dynamic and congested environments. The effectiveness of the proposed method was verified through experiments and by simulation. We believe that this research will enable autonomous electric wheelchairs to transport people in crowded environments.

REFERENCES

Bar-Shalom, Y., Daum, F., and Huang, J. (2009). The probabilistic data association filter. *IEEE Control Systems Magazine*, 29(6), 82–100.

Bardaro, G., Bascetta, L., Ceravolo, E., Farina, M., Gabellone, M., and Matteucci, M. (2018). Mpc-based control architecture of an autonomous wheelchair for indoor environments. *Control Engineering Practice*, 78, 160–174.

Biber, P. and Strasser, W. (2003). The normal distributions transform: a new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, volume 3, 2743–2748 vol.3.

Ceravolo, E., Gabellone, M., Farina, M., Bascetta, L., and Matteucci, M. (2017). Model predictive control of an autonomous wheelchair. *IFAC-PapersOnLine*, 50(1), 9821–9826. 20th IFAC World Congress.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1), 23–33.

Li, Z., Xiong, Y., and Zhou, L. (2017). Ros-based indoor autonomous exploration and navigation wheelchair. In *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, 132–135. IEEE.

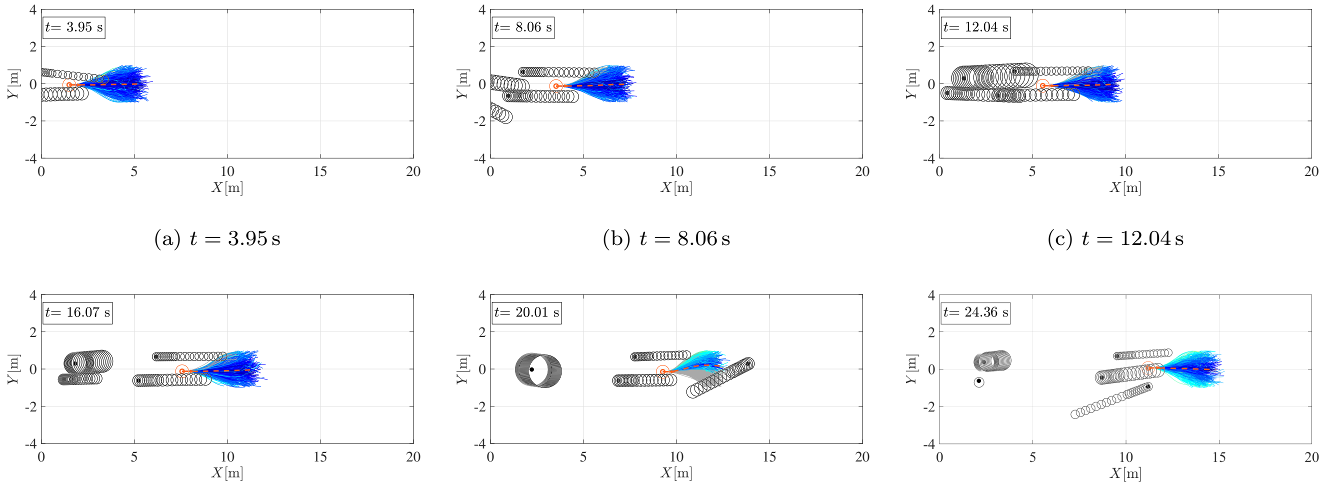
Maki, R., Okawa, I., and Nonaka, K. (2022). Stochastic model predictive obstacle avoidance with velocity reduction in accordance with chance constraints in crowded environments. *International Journal of Automotive Engineering*, 13(3), 114–121.

Matsuura, H., Nonaka, K., and Sekiguchi, K. (2022). Model predictive obstacle avoidance control for an electric wheelchair in indoor environments using artificial potential field method. In *2022 IEEE/SICE International Symposium on System Integration (SII)*, 19–24.

Nagata, A., Nonaka, K., and Sekiguchi, K. (2014). Model predictive obstacle avoidance control for omnidirectional mobile robots based on fuzzy potential method. In *2014 European Control Conference (ECC)*, 352–357.

Sezer, V. (2022). An optimized path tracking approach considering obstacle avoidance and comfort. *Journal of Intelligent & Robotic Systems*, 105(1), 1–14.

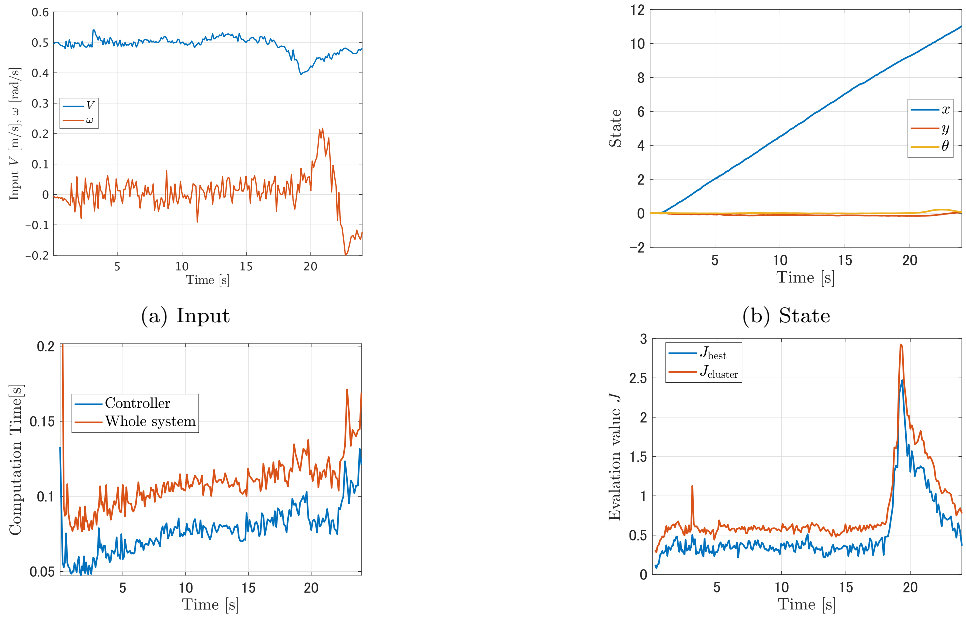
Sezer, V. and Gokasan, M. (2012). A novel obstacle avoidance algorithm: "follow the gap method". *Robotics and Autonomous Systems*, 60(9), 1123–1134.



(a) $t = 3.95\,\mathrm{s}$ (b) $t = 8.06\,\mathrm{s}$ (c) $t = 12.04\,\mathrm{s}$

(d) $t = 16.07\,\mathrm{s}$ (e) $t = 20.01\,\mathrm{s}$ (f) $t = 24.36\,\mathrm{s}$

Fig. 12. Movement of the electric wheelchair and obstacles in the experiment



(a) Input (b) State

(c) Computation time (d) Evaluation value

Fig. 13. Experimental results

Shibata, K., Nonaka, K., and Sekiguchi, K. (2019). Model predictive obstacle avoidance control suppressing expectation of relative velocity against obstacles. In *2019 IEEE Conference on Control Technology and Applications (CCTA)*, 59–64.

Shibata, K., Shibata, N., Nonaka, K., and Sekiguchi, K. (2018). Model predictive obstacle avoidance control for vehicles with automatic velocity suppression using artificial potential field. *IFAC-PapersOnLine*, 51(20), 313–318.

Suzuki, T. and Takahashi, M. (2009). Translational and rotational movement control considering width for autonomous mobile robot by using fuzzy inference. In *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 257–262.

Suzuki, T. and Takahashi, M. (2011). Obstacle avoidance for autonomous mobile robots based on position prediction using fuzzy inference. In J. Awrejcewicz (ed.), *Numerical Analysis*, chapter 27. IntechOpen, Rijeka.

Thrun, S. (2002). Probabilistic robotics. *Communications of the ACM*, 86–88.

Warren, C. (1990). Multiple robot path coordination using artificial potential fields. In *Proceedings., IEEE International Conference on Robotics and Automation*, 500–505 vol.1.

Zhu, Q., Yan, Y., and Xing, Z. (2006). Robot path planning based on artificial potential field approach with simulated annealing. In *Sixth International Conference on Intelligent Systems Design and Applications*, volume 2, 622–627.