Real-time model predictive obstacle avoidance control for vehicles with reduced computational effort using constraints of prohibited region

Kazuki KIMURA*, Kenichiro NONAKA* and Kazuma SEKIGUCHI*
* Department of Mechanical Systems Engineering, Tokyo City University
1-28-1 Tamazutsumi, Setagaya-ku, Tokyo, 158-8557, Japan
E-mail: knonaka@tcu.ac.jp

Abstract
This paper focuses on a real-time obstacle avoidance control method for vehicles using model predictive control (MPC). MPC can optimize the motion of the vehicle over a finite time horizon while satisfying various constraints such as vehicle dynamics, the road width and the steering range. However, the computational cost is too large for conducting real-time control. In this paper, a collision avoidance is realized by MPC with constraints for avoiding prohibited regions represented as circles. We approximate this region into a half plane separated by the tangent of the prohibited region. By handling approximated regions as constraints of the road width of MPC, we can implement the collision avoidance algorithm into the controller without increasing the computational cost. Moreover, in order to reduce the computational effort, we transform the nonlinear vehicle dynamics into reduced order and linearizable subsystems called time-state control form (TSCF). The effectiveness of the proposed method is proved by comparative simulations with conventional method where artificial potential method is applied to MPC. In addition, we conduct two experiments using a 1/10 scale vehicle which is equipped with a laser range finder to execute obstacle detection and localization. We show that real-time control can be realized even if we use an on-board embedded CPU which runs at the frequency of 500 MHz.

Key words: Vehicle control, Model predictive control, Obstacle avoidance, Time state control form, Front steering vehicle, Obstacle detection, Laser range finder

1. Introduction

Automatic control of vehicles has been studied intensively in recent years. They are expected to be applied to, for example, guard robots which patrol buildings, navigation robots in museums to lead visitors and autonomous wheel chairs for elder person. Model predictive control (MPC) seems to be promising because it can optimize the motion of the controlled object over a finite time horizon while satisfying various constraints. However, the computational cost for solving the optimization problem in real-time is an issue to be resolved; MPC has been applied to a real-time control of relatively slow-reacting systems (Souza, et al., 2010). To address this challenge, several advanced methods have been proposed like a custom solver for solving the optimization problem at high speed (Mattingley, et al., 2011) and an optimization algorithm utilizing continuation (Ohtsuka, 2004). Hence, MPC has been applied to the various vehicle control problems such as path tracking (Keviczky, et al., 2006) (Gu and Hu, 2006) (Li, et al., 2004), obstacle avoidance (Choi, Kang and Lee, 2012) (Xi and Baras, 2007) (Abbas, et al., 2012) and adaptive cruise control (Shakouri, 2011). In these studies, solving the nonlinear MPC problem is required because of the nonlinearity of the vehicle dynamics. However, the high computational cost in the nonlinear optimization problem is still a problem even with the efficient solvers. In (Falcone, et al.,2007), a linear time-varying model predictive control is proposed to reduce the computational cost and applied to the double lane change problem, but it apparently suffers from the linear approximation errors. For this challenge, a model predictive parking control with constraints of the road width and the steering angle is proposed.
in (Oyama and Nonaka, 2013), where time-state control form (TSCF) (Sampei, et al., 1999) is utilized in order to reduce the control effort for the kinematic model which works well especially in the low-speed running. In (Oyama and Nonaka, 2013), the third order nonlinear vehicle dynamics can be divided into second and first order subsystems by TSCF: the nonlinear programming problem is transformed into a quadratic programming (QP) problem by exact linearization of the subsystems to reduce the computational costs of MPC.

Since obstacle avoidance is one of the most important tasks for autonomous vehicles, it has been studied for a long time. Artificial potential field (APF) method has been applied to various obstacle avoidance problems (Khatib, 1985)(Ge and Cui, 2002)(Ito, et al., 2011) as a common technique since the structure of APF is simple and tractable. However, if we apply APF to vehicles having restrictions on both the steering angle and the movable range, an infeasible path might be generated depending on the surrounding objects and controller parameters. In our previous study, we proposed an obstacle avoidance control method combining APF with MPC to take into account vehicle dynamics, where the limitation of the steering angle and the road width (Kimura and Nonaka, 2013) directly. Although we can predict the optimal and feasible behavior by using MPC which can consider these constraints, the tuning problem of the weight constants still remains: it is difficult to determine an appropriate weight constants for guaranteeing the collision avoidance on various situations. As a method without this tuning problem, the prohibited region including the obstacle is introduced as constraints (Mukai, et al., 2008). Although the obstacle avoidance is achieved using MPC with mixed integer programming, it might require excessive computational cost. In (Nilsson, et al., 2013), the constraint of the prohibited region is approximated to the linear constraint, where the optimization problem can be represented as a QP which can be solved at high speed by using solver like (Mattingley and Boyd, 2012). But in (Mukai, et al., 2008) (Nilsson, Ali, Falcone and Sjoberg, 2013), some additional constraints are required to achieve obstacle avoidance. Moreover in (Nilsson, et al., 2013), two linear constraints for obstacle avoidance are switched by introducing slack variables which vary depending on the relative positions between the vehicle and the obstacle. Increasing the number of constraints and variables of the optimization problem renders the computational cost high.

In this paper, we propose an obstacle avoidance control based on MPC which guarantees the collision avoidance with reduced computational cost. We suppose prohibited region of a circle, and we approximate the prohibited region into a linear constraint utilizing the half plane separated by the tangent of the region; then we assume this constraint as a variable road width so that the collision avoidance algorithm can be implemented into the controller based on (Oyama and Nonaka, 2013) without increasing the computational cost. We transform the nonlinear vehicle dynamics into the reduced order linear subsystem using TSCF and exact linearization. The effectiveness of the proposed method is proved through comparative simulations with conditions where the conventional method (Kimura and Nonaka, 2013) cannot avoid the prohibited region. In addition, the effectiveness of the proposed method is demonstrated using the 1/10 scale vehicle equipped with the laser range finder for obstacle detection and localization. We show that collision avoidance is guaranteed using our proposed method in the situation including mobile obstacle. We also show that real-time control can be realized even with an on-board embedded CPU which runs at the frequency of 500 MHz.

2. Vehicle Model Described by Time-State Control Form (Sampei, et al., 1999)

Figure 1 depicts a single-track model for a front steering vehicle. Since we consider obstacle avoidance control at low speed, in this study, we assume that side-slip angles of wheels are negligible. The rear wheel position and attitude angle of the vehicle are represented as \((x_v, y_v, \theta)\) on \(X - Y\) coordinates. The state space equation of the vehicle is represented as follows:

\[
\begin{align*}
\frac{d}{dt} \begin{bmatrix} x_v \\ y_v \\ \theta \end{bmatrix} &= \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix} V + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \frac{V}{L_b} \tan(\delta), \\
\end{align*}
\]

where \(V\) is the rear wheel velocity, \(\delta\) is the steering angle of the front wheel, \(L_b\) is the wheelbase and \(y_v(x)\) is the desired path. Since the dynamics Eq. (1) is nonlinear, the computational effort on MPC is large; further reduction of the computational one is expected in order to utilize MPC for real-time control. In order to reduce the computational effort, the time-state control form (TSCF) (Sampei, et al., 1999) and exact linearization are applied to the vehicle dynamics Eq. (1). We divide the second and third row of the Eq. (1) by the first one to get the following equations:

\[
\frac{dy_v}{dx_v} = \tan(\theta)
\]
\[
\frac{d\theta}{dx_c} = \frac{1}{L \cos(\theta)} \tan(\delta).
\]
Equation (2) is differentiated with respect to the state \(x_c\), and then we define the right hand side of the following equation as \(\mu_2\):
\[
\frac{d^2y_v}{dx_c^2} = \frac{1}{L \cos^3(\theta)} \tan(\delta) =: \mu_2.
\]
We take the state \(x_c\) as the time axis instead of the time \(t\), and the nonlinear state space equation can be transformed into the two subsystems represented as follows:
\[
\frac{dx_c}{dt} = \mu_1 \tag{5}
\]
\[
\frac{d}{dx_c} \xi = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mu_2 \tag{6}
\]
where \(\xi := [y_v, \tan(\theta)]^T\) is the new state vector, \(\mu_1 := V \cos(\theta)\) is a positive value which monotonically increases time axis \(x_c\), \(\mu_2\) is a virtual input computed by solving the optimization problem. It is noted that Eq. (5) and Eq. (6) are independent linear dynamics with reduced dimensions: drastic reduction of the computational cost on MPC can be expected. The actual steering angle \(\delta\) can be computed with \(\mu_2\) in Eq. (4) as follows:
\[
\delta = \tan^{-1}(L_b \cos^3(\theta) \mu_2). \tag{7}
\]
### 3. Model Predictive Control

#### 3.1. General formulation

In model predictive control (MPC), the states of the controlled object from current time till finite time future are predicted based on its dynamics, and the optimal control input minimizing the index function is computed at each control step, while constraints are kept satisfied. We briefly review the general form of MPC for discrete time systems. The state space equation of the controlled object is represented as
\[
\xi[k + 1] = f(\xi[k], u[k], k) \tag{8}
\]
where \(\xi[k] \in \mathbb{R}^n\) is the state vector, \(u[k] \in \mathbb{R}^m\) is the input vector, \(k \in \mathbb{Z}_{\geq 0}\) is the discrete time, \(n\) is the dimension of the state, and \(m\) is the dimension of the input. Constraints are represented as scalar-valued equalities or inequalities for \(g_i\) and \(h_j\) as follows:
\[
g_i(\xi[k], u[k], k) = 0 \quad (i = 0, 1, \ldots, l), \tag{9}
\]
\[
h_j(\xi[k], u[k], k) \leq 0 \quad (j = 0, 1, \ldots, p), \tag{10}
\]
where \(l\) and \(p\) are the number of equality and inequality constraints, respectively. The index function of MPC is
\[
J = \varphi(\xi[H]) + \sum_{k=0}^{H-1} L(\xi[k], u[k], k) \tag{11}
\]
where $H$ is the number of the prediction steps, $q(\zeta[H])$ is the terminal cost and $L(\zeta[k], u[k], k)$ is the stage cost. MPC is an optimal control technique which computes the optimal control input minimizing the index function Eq. (11) at each control cycle while satisfying the constraints Eq. (8) (9) and (10).

3.2. **MPC for front steering vehicle (Oyama and Nonaka, 2013)**

Since the optimization problem over a finite time horizon must be solved at each control cycle, the issue of MPC is that computational cost becomes high, especially for nonlinear dynamics like Eq. (1). But for the reduced order and linearized subsystem Eq. (6), MPC becomes a linear optimization problem and computational cost is far small.

Equation (6) is transformed into a discrete-time system:

\[
\begin{pmatrix}
1 & \Delta \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
\zeta[k+1] \\
\mu_2[k]
\end{pmatrix} = 
\begin{pmatrix}
\Delta^2 \\
\Delta
\end{pmatrix}
\begin{pmatrix}
\zeta[k] \\
\mu_2[k]
\end{pmatrix}
\]

(12)

where $\Delta$ is the step width with respect to time-state $x_o$. It is noted that $k$ indicates discrete steps with respect to the motion along $x_o$. $\zeta[k] := [\zeta_1[k], \zeta_2[k]]^T = [y_o[k], \tan(\theta)[k]]^T$ is the state vector at time $k$. We take the following index function of MPC for path tracking control:

\[
J = \hat{\zeta}[H]^T Q_f \hat{\zeta}[H] + \sum_{i=0}^{H-1} \hat{\zeta}[k]^T Q \hat{\zeta}[k] + R \mu_2[k]^2
\]

(13)

where $R$ is the positive weight constant, $Q$ and $Q_f$ are positive definite weighting matrices, $H$ is the number of the prediction steps with respect to the time axis and $\hat{\zeta} = \zeta - \zeta_r$ is an error between the state vector $\zeta$ and its target $\zeta_r$. In addition to the constraints due to the dynamics Eq. (12), the physical constraints on the road width and steering angle are represented by the following inequalities:

\[
y_o[k] \leq \zeta_1[k] \leq \bar{y}_o[k]
\]

(14)

\[
\tan(\hat{\delta}) \leq \Gamma(\zeta_2[k], \mu_2[k]) \leq \tan(\hat{\delta})
\]

(15)

where $y_o[k]$ and $\bar{y}_o[k]$ are lower and upper bounds of the road width, $\hat{\delta}$ and $\hat{\delta}$ are limitation of the steering angle, and $\Gamma(\zeta_2[k], \mu_2[k]) := L_o \cos^3 \left(\tan^{-1}(\zeta_2[k])\right) \mu_2[k]$. In order to linearize the nonlinear constraints (15), the following approximated function $\hat{\Gamma}$ indicating the first order Taylor series expansion of $\Gamma$ at each prediction step is considered:

\[
\hat{\Gamma}(\zeta_2[k], \mu_2[k]) := \Gamma(\hat{\zeta}_2[k], \mu_2[k]) + \frac{\partial \Gamma}{\partial \zeta_2} (\hat{\zeta}_2[k] - \zeta_2[k]) + \frac{\partial \Gamma}{\partial \mu_2} (\mu_2[k] - \hat{\mu}_2[k])
\]

(16)

where $\hat{\mu}_2[k]$ and $\tan(\hat{\delta}[k])$ are the input and the state of previous sampling at prediction step $k$, respectively. We replace the nonlinear constraints Eq.(15) with the following linear constraints:

\[
\tan(\hat{\delta}) \leq \hat{\Gamma}(\zeta_2[k], \mu_2[k]) \leq \tan(\hat{\delta})
\]

(17)

Thus, nonlinear programming problem can be treated as a QP which computes the optimal control input $\mu_2$ minimizing the quadratic index function Eq. (13) under three sets of linear constraints Eq. (12)(14)(17). In general, QP problems are tractable both theoretically and mathematically since we can get a globally optimal solution without recursive computation. Thus, the reduction of the computational effort of MPC is expected by designing optimization problem as the QP with linear constraints.

4. **Artificial potential field for avoidance (Kimura and Nonaka, 2013)**

As a conventional method, we consider the obstacle avoidance by applying artificial potential field (APF) method to the index function of MPC. The following penalty term Eq. (18) with respect to the distance between the vehicle and the obstacle is added to the index function of MPC given in Eq. (13) at each predicted step:

\[
J_p = \sum_{i=1}^{N_o} \sum_{k=0}^{H} \frac{Q_o}{\left( x_o[k] - x_o[z][k] \right)^2 + \left( y_o[k] - y_o[z][k] \right)^2}
\]

(18)

where $(x_o[k], y_o[k])$ is the position of the vehicle and $(x_o[z][k], y_o[z][k])$ is the position of the obstacle at prediction step $k$, respectively, $N_o$ is the number of the obstacles, and $Q_o$ is the positive weight constant. In order to treat the optimization problem as a quadratic one, we use a second order Taylor series approximation of Eq. (18) at the position of the vehicle.
Table 1  Simulation condition

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial state</td>
<td>((x_0, y_0, 0))</td>
</tr>
<tr>
<td>Desired path (y_i(x))</td>
<td>(Y = 2.5) m</td>
</tr>
<tr>
<td>Radius of the prohibited region(Case 1)</td>
<td>(R_{p1}, R_{p2}) 0.39 m, 0.49 m</td>
</tr>
<tr>
<td>Radius of the prohibited region(Case 2)</td>
<td>(R_{p1}, R_{p2}) 0.49 m, 0.49 m</td>
</tr>
<tr>
<td>Obstacle position(Case 1)</td>
<td>((x_{o1}, y_{o1}), (x_{o2}, y_{o2})) (3.0,2.7) m, (4.5,2.0) m</td>
</tr>
<tr>
<td>Obstacle position(Case 2)</td>
<td>((x_{o1}, y_{o1}), (x_{o2}, y_{o2})) (3.0,2.7) m, (4.5,2.3) m</td>
</tr>
<tr>
<td>Limitation of the road width ((y, \bar{y}))</td>
<td>(1.0,4.0) m</td>
</tr>
<tr>
<td>Limitation of the steering angle ((\delta, \bar{\delta}))</td>
<td>((-\pi/9, \pi/9)) rad</td>
</tr>
<tr>
<td>Step width (\Delta)</td>
<td>0.2 m</td>
</tr>
<tr>
<td>Horizon</td>
<td>7</td>
</tr>
<tr>
<td>Weighting matrix (Q)</td>
<td>diag(3.5, 0.5)</td>
</tr>
<tr>
<td>Weight constant (R)</td>
<td>0.15</td>
</tr>
<tr>
<td>Weighting matrix (Q_f)</td>
<td>diag(3.5, 0.5)</td>
</tr>
<tr>
<td>Weight constant (Q_o)</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Fig. 2  Simulation result using the conventional method. There are two obstacles near the desired path. The vehicle succeeds in avoiding obstacles in case 1 (blue) with the weight constant which have already tuned well. On the other hand, the vehicle fails collision avoidance in case 2 (red) due to the change of the environment, which requires tuning of the weights.

at each predicted step. In this paper, we aim to realize guaranteed collision avoidance while the vehicle tracks the desired path; we define the prohibited region to reflect the size of the vehicle and obstacles. The simulation results using the conventional method is depicted in Fig. 2 where there are two obstacles located near the desired path. The simulation conditions are shown in Table 1. In case 1, the vehicle successfully avoid the prohibited region. In case 2, the prohibited region of the obstacle No.1 becomes large and the position of the obstacle No.2 is shifted to the target path. In this case, when we use the same weight constant, the vehicle collides with the both obstacles, which requires tuning of the weight constant according to the change of environments.

To avoid such tuning problem, we use the constraint of the prohibited region for guaranteeing collision avoidance, but the computational cost will increase because the prohibited region becomes a quadratic constraint. In our proposed method, the prohibited region is approximated as the half plane of the obstacle side separated by the tangent of the prohibited region; the computational cost can be reduced because the quadratic constraint becomes linear one.

5. Proposed method for obstacle avoidance

5.1. Outline of the proposed method

In this section, the outline of the proposed method is summarized. The configuration procedure of the constraint is as follows:

1)  The prohibited region is represented as a circle.

2)  The prohibited region corresponding to the each predicted position of the vehicle is linearized as the tangent of the prohibited region.

3)  The boundary value along this tangent is calculated at each predicted position of the vehicle.
4) The lower and upper bound of road width constraints are determined based on the boundary value.

It is noted that the road width bounds are directly applied to the constraints of (Oyama and Nonaka, 2013) without increasing the number of constraints. The schematic view of the prohibited region at each prediction step till 3 steps future is depicted in Fig. 3 where the approximated prohibited regions are represented as the half planes of the cross-hatching side separated by the each tangent.

5.2. Constraint represented as a half plane

Suppose the radius of the prohibited circle region is $R_p$. The point $(x_c, y_c)$ where the line connecting the positions between the vehicle and the obstacle intersects the circle of the prohibited region is represented as follows:

$$
\begin{align*}
x_c &= x_o + R_p \frac{(x_o - x_v)}{\sqrt{(x_o - x_v)^2 + (y_o - y_v)^2}} \\
y_c &= y_o + R_p \frac{(y_o - y_v)}{\sqrt{(x_o - x_v)^2 + (y_o - y_v)^2}}
\end{align*}
$$  \hspace{1cm} (19)

The tangent at the intersection $(x_c, y_c)$ is

$$(x_v - x_o)(x - x_v) + (y_v - y_o)(y - y_v) = 0.$$  \hspace{1cm} (20)

The boundary value $y_b$ of Eq. (20) corresponding to the position of the vehicle is represented as follows:

$$y_b = y = -\frac{\ddot{x}}{\ddot{y}} x_o + x_v \ddot{x} + y_v \ddot{y} \quad \text{(21)}$$

where $\ddot{x}$ is $x_v - x_o$ and $\ddot{y}$ is $y_o - y_v$, respectively. In the neighborhood of the singular point with $\ddot{y} = 0$, Eq. (21) is approximated as

$$
\begin{align*}
y_b &= -\frac{\ddot{x}}{\ddot{y}} x_o + x_v \ddot{x} + y_v \ddot{y} \\
y_b &= -\frac{\ddot{x}}{\ddot{y}} x_o + x_v \ddot{x} - y_o \ddot{y} 
\end{align*} \quad (0 \leq \ddot{y})
$$

when $-\epsilon \leq \ddot{y} \leq \epsilon$. To decide the constraint based on the boundary value calculated at each control cycle, 4 cases classified by the relative position between the vehicle and the obstacle are depicted in Fig. 4, where deep pink region is the quadratic constraint, and pink region is the linear constraint utilizing the proposed method. The lower and upper bound of the road width $y[k]$ and $\bar{y}[k]$ are determined based on Fig. 4. Although the prohibited region is approximated, the road width constraints can guarantee collision avoidance. It is noted that the multiple obstacle avoidance can be achieved using the most restrictive boundary value. Since we have included the obstacle avoidance into the road width constraints, the proposed method can be implemented without increasing the computational cost.

5.3. Vehicle speed control

The vehicle should decelerate to the safe velocity when the vehicle approaches obstacles. The reference velocity $V_r$ for $V$ is determined as an interior division between maximum and minimum velocities, $V_{max}$ and $V_{min}$, by the distance
between the vehicle and the closest obstacle as follows:

\[ V_r = V_{\text{max}} - (V_{\text{max}} - V_{\text{min}}) \frac{d_p^2}{d^2} \]  

where \( d \) is the distance between the vehicle and the obstacle and \( d_p \) is the smallest distance guaranteed by the constraints of MPC.

### 5.4. Model of the mobile obstacle

We assume that the maximum size of the obstacle is known and the motion of the obstacle is linear with a constant velocity. The predicted position of the obstacle is represented as

\[
\begin{bmatrix}
    x_{o}[k+1] \\
    y_{o}[k+1]
\end{bmatrix} = \begin{bmatrix}
    x_{o}[k] \\
    y_{o}[k]
\end{bmatrix} + \begin{bmatrix}
    v_x \\
    v_y
\end{bmatrix} \frac{\Delta}{V \cos(\theta[k])} \quad (k = 0, \cdots, H - 1),
\]

where \( H \) is the number of the prediction step of MPC, \( \Delta \) is the step width on the virtual time axis, \( V \) is the vehicle velocity, and \( \theta[k] \) is attitude angle at the prediction step \( k \) and \((v_x, v_y)\) is the velocity of the obstacle.

### 5.5. Simulation result using proposed method

In our proposed method, there are three constraints: state space equation (12) linearized by TSCF, the linear road width Eq. (14) introduced for obstacle avoidance and steering angle Eq. (17) linearized by first order Taylor series approximation. Hence, we can realize obstacle avoidance control within the framework of path tracking control (Oyama and Nonaka, 2013) where the optimization problem is described as a QP problem. In order to solve the QP, we use CVXGEN (Mattingley and Boyd, 2012) which is able to generate an optimized C code to solve QP problems based on an interior-point method. This solver is suitable for QP problems of small size, and they can be solved at high speed because CVXGEN optimizes the code for solving the optimization problem by exploiting the structure of the QP. Let \( n \) and \( m \) are dimension of the state and the input, respectively, and \( H \) is the horizon. The order of operations in an interior-point method to solve the MPC optimization is \( O(H(n + m)^3) \), as opposed to \( O(H^3(n + m)^3) \) if the structure of the QP is not exploited (Wang and Boyd, 2008).

Figure 5 depicts the simulation result under the same condition with Fig. 2. In the proposed method, the vehicle can avoid the prohibited region even if the size or the position of the prohibited region is changed, which proves the effectiveness of the proposed method.

### 6. Experiment

#### 6.1. Obstacle detection and estimating the radius of prohibited region

In our experiments, the obstacle position is measured by using laser range finder (LRF). We have to perform clus-
Fig. 5 Simulation result using the proposed method. There are two obstacles near the desired path. The vehicle succeeds collision avoidance in both case 1 (blue) and case 2 (red). Thus the proposed method can avoids obstacles without tuning controller parameters even if the environment is changed.

Fig. 6 Clustering algorithm of obstacles for LRF. We distinguish the obstacles by utilizing the distance between the successive data points. The gravity centers (cross mark) for each separated obstacle are set to be the obstacle positions.

tering so that we can distinguish between the stationary obstacle and the mobile one. The distance data and the direction angle to obstacles can be obtained by LRF. For explanation, suppose that M distance data to the obstacle are obtained. We define $O_j = (X_j, Y_j)$ as the coordinates of the data calculated from $j$-th sensor data $(j = 1, \cdots, M)$, the vehicle position $(x_v, y_v)$ and attitude angle $\theta$. The schematic is as depicted in Fig. 6, where the algorithm of clustering is comprised of the following steps:

1) The first coordinates data $O_j$ $(j = 1)$ is tagged as the obstacle No.1.

2) $O_{j+1}$ is recognized as the same obstacle number if the distance $D_j$ between $O_j$ and $O_{j+1}$ is less than the threshold value. Otherwise, $O_{j+1}$ is recognized as the next obstacle number.

3) Let $j := j + 1$, and then go to 2) if $M > j$.

4) The gravity center position for each separated obstacle are set to be obstacle position.

Though we cannot obtain the accurate center point of the obstacle in this method, collision avoidance is achieved since the estimated obstacle position is located between the vehicle and the actual obstacle position. In this study, suppose each
obstacle is included in a circle whose radius is known. We describe the prohibited region as a circle whose radius is $R_p$, which is the sum of the radius of an obstacle and the maximum distance between the vehicle position and the outline of the vehicle. This maximum distance is determined based on an experimental vehicle.

6.2. Estimation of position and velocity of the obstacle

In this study, we estimate both the position and the velocity of the obstacle based on LRF data. This section describes the estimation of the $x$-direction for brevity. Let $q = [x, v]^T$ be the state vector, where $x$ and $v$ are the position and the velocity, respectively. The stochastic state space equation of the mobile obstacle moving at a constant speed is

$$
\dot{q} = \begin{bmatrix}
0 & 1 \\
0 & 0
\end{bmatrix} q + \begin{bmatrix}
0 \\
1
\end{bmatrix} n_u
$$

(25)

$$
y_k = \begin{bmatrix}
1 & 0
\end{bmatrix} q + n_y
$$

(26)

where $n_u$ is the system noise of the $x$-directional acceleration and $n_y$ is the observation noise which represents measurement error of the LRF. The $n_u$ and $n_y$ are assumed to be Gaussian white noise with zero mean and covariance matrices $Q_k$ and $R_k$, respectively. $R_k$ is determined based on the variance of the LRF data, while $Q_k$ represents the variance of the moving velocity. The estimation of $q$ is computed by Kalman filter at every control step.

6.3. Experimental system and condition

In this study, we use RoboCar® 1/10 manufactured by ZMP Inc. RoboCar is a front steering and rear wheel driving...
vehicle, whose photograph is depicted in Fig. 7. Table 2 shows the specification of RoboCar. The computation of controller is performed by an on-board embedded CPU which runs at the frequency of 500 MHz. The velocity of the vehicle is measured by rotary encoders equipped to each wheel. LRF is equipped on the RoboCar for obstacle detection and localization (Hiromachi, Nonaka and Sekiguchi, 2014). Table 3 shows the specification of LRF. A cylinder whose radius is 0.1 m made from expanded polystyrene is used for the obstacle. The maximum distance between the vehicle position and the outline of the vehicle is 0.39 m. In this study, to investigate the feasibility of our collision avoidance method, we assume that the radius of all prohibited regions is $R_p = 0.49$ m. Fig. 8 depicts the block diagram of the system for experiments. Localization of the vehicle and the obstacle are performed based on distance data measured by LRF. The position $(\hat{x}_o, \hat{y}_o)$ and the velocity $(\dot{x}_o, \dot{y}_o)$ of the obstacle are estimated by Kalman filter based on the data of the obstacle positions $(x_o, y_o)$. The optimal steering angle $\delta$ is computed by MPC based on the position and attitude angle of the vehicle with the position and the velocity of the obstacle. In this paper, we conduct two experiments, whose conditions are shown in Table 4. We show that the real-time control of the proposed method can be implemented into a 500 MHz CPU, and successfully achieves the collision avoidance.
6.4. Avoiding two stationary obstacles

Figure 9 depicts the initial condition of the vehicle and the obstacles, where this condition is same with the case 2 in Fig. 3. The running path of the vehicle and the positions of obstacles are shown in Fig. 11 where the vehicle successfully avoids the obstacles. The computation time is shown in Fig. 12 where real-time control is achieved at the computational time less than 10 ms. The front steering angle is shown in Fig. 13 which shows that the steering angle constraint is satisfied. However, the chattering phenomenon of the steering angle is observed at around $t = 2$ s. The chattering depends on the length of the step size; we confirmed that it can be suppressed by reducing the length of the step width. The velocity of the vehicle is shown in Fig. 14 where the velocity of the vehicle can be changed within the limits of the lower and upper bound of velocity which depends on the distance between the vehicle and the obstacle. The experimental results at $t = 0, 6, 12, 18$ s are depicted in Fig. 15–Fig. 18 where the red points are the current positions of the vehicle and the obstacles, while other colors are predicted positions of the vehicle, the interior of the gray circles are prohibited regions and pink lines are the lower and upper bounds of road width constraints. The vehicle successfully avoids the prohibited region along with proper change of bounds of road width constraints. The experimental results indicate that the vehicle achieves the minimum avoidance along each prohibited region.

6.5. Avoiding a mobile obstacle

Figure 10 depicts the initial condition of the vehicle and the obstacles, where the stationary obstacle is located in the lower side of the desired path and the mobile obstacle is located in the upper side of the desired path, respectively. In this experiment, we set that the vehicle reaches the mobile obstacle which approaches toward the vehicle just before reaches the stationary one. Hence, it is desirable that the vehicle avoids through the lower side of the obstacles in advance by considering the future situation. The running path of the vehicle and the positions of obstacles are shown in Fig. 19 where the vehicle can avoid the obstacles. The computation time is shown in Fig. 20 where real-time control is achieved at the computational time less than 10 ms. The front steering angle is shown in Fig. 21 which shows that the steering angle constraint is satisfied. The velocity of the vehicle and the mobile obstacle is shown in Fig. 22 where the velocity of the vehicle can be changed within the limits of the lower and upper velocity depending on the distance between the vehicle and the obstacle. In addition, the velocity of the mobile obstacle is estimated. The experimental results at $t = 0, 3, 6, 9$ s are depicted in Fig. 23–Fig. 26 where the red points are the current positions of the vehicle and the obstacles, while other colors are predicted positions of them, the inside of the each color circles are prohibited regions corresponding to the each predicted position of the obstacle and pink lines are the lower and upper bounds of road width constraints. The prohibited regions of the mobile obstacle are drawn at prediction steps $k = 0, 3, 6$. By changing the lower and upper bounds of road width constraints, the vehicle successfully avoids the prohibited regions. In Fig. 24, the vehicle took the ideal path for avoiding through the lower side of the obstacles, since the constraint of the prohibited region of the predicted position becomes effective in this case. Thus, the experimental results indicate that the minimum avoidance along each prohibited region is achieved by considering the future motion of the mobile obstacle, and optimizing the dynamics of the vehicle.

7. Conclusion

In this paper, a fast model predictive obstacle avoidance control for vehicles with constraints of the road width and the steering angle is proposed. By approximating the quadratic constraint of the prohibited region to a linear one, the approximated prohibited regions can be treated as linear road width constraints of MPC which reduces the computational cost. The effectiveness of the proposed method is proved by both simulations and actual experiments, which show that smooth collision avoidance is guaranteed. In addition, experimental results indicate that the proposed method can achieve real-time control at the computational time less than 10 ms using an embedded CPU with clock frequency of only $500$ MHz. In this study, we assume that the radius of the obstacle is known, which might be estimated using measured distance of LRF. In addition, the avoidance route depends on the parameters of MPC like horizon, step width and location of obstacles, which should be investigated. We left them for future works.

8. Acknowledgment

The authors gratefully acknowledge the support of Grant-in-Aid for Scientific Research (C) No.24560555 of Japan.
Fig. 11 Trajectory

Fig. 12 Computational time

Fig. 13 Steering angle

Fig. 14 Velocity

Fig. 15 Experimental results (t = 0 s)

Fig. 16 Experimental results (t = 6 s)

Fig. 17 Experimental results (t = 12 s)

Fig. 18 Experimental results (t = 18 s)
Fig. 19 Trajectory

Fig. 20 Computational time

Fig. 21 Steering angle

Fig. 22 Velocity

Fig. 23 Experimental results ($t = 0$ s)

Fig. 24 Experimental results ($t = 3$ s)

Fig. 25 Experimental results ($t = 6$ s)

Fig. 26 Experimental results ($t = 9$ s)
References


