Trajectory Generation based on Model Predictive Control with Obstacle Avoidance between Prediction Time Steps

Kazuyuki Kon * **Hiroaki Fukushima** * **Fumitoshi Matsuno** *

* *Department of Mechanical Engineering and Science, Graduate School of Engineering, Kyoto University (e-mail: k.kon@fs4.ecs.kyoto-u.ac.jp)*

Abstract: In this paper, model predictive control (MPC) based trajectory generation for a unicycle with obstacle avoidance is addressed. Since MPC based methods can only consider obstacle avoidance at discrete time steps, a collision may occur in intervals between prediction time steps. In order to prevent this problem, we propose two methods which guarantee the obstacle avoidance not only at prediction time steps but also in intervals between them. The flrst method reduces the maximum velocity near obstacles. The second one constrains transition of collision avoidance constraints to exclude possibilities of collisions between time steps. Since both methods do not need iteration of optimization, it is expected that the computation time is not signiflcantly increased compared with existing methods. Numerical examples and experiments show the efiectiveness of the proposed methods.

Keywords: Model predictive control, Trajectory generation, Obstacle avoidance, Mixed integer quadratic programming

1. INTRODUCTION

Trajectory generation for a unicycle with obstacle avoidance has been an active research fleld (e.g. [Guo and Tang (2008)], [Ferguson and Stentz (2006)]). A systematic tool for handling obstacle avoidance problems is model predictive control (MPC). MPC determines the current control action by solving on-line a flnite horizon open-loop control optimization problem [see e.g. Mayne et al. (2000)]. MPC has been applied to trajectory generation [e.g. Xi and Baras (2007)] and path tracking for mobile robot[Gomez-Ortega and Camacho (1995), Klancar and Skrjanc (2007)].

One way to address trajectory generation with obstacle avoidance by MPC is to formulate as a mixed integer programming(MIP). Since the obstacle avoidance conditions can be described by linear inequality constraints including binary integer variables, MPC can be implemented using the MIP [e.g. Richards and How (2003)]. Since MPC can similarly deal with collision avoidance among vehicles, Multi-vehicle formation control methods, which handle collision avoidance among vehicles using MPC, have been reported in the literature [e.g. Kopfstedt et al. (2006), Kuwata et al. (2007), Fukushima et al. (2005), Kon et al. (2007)].

In MPC based trajectory generation methods, obstacle avoidance is guaranteed at discrete time steps over a prediction horizon. Thus, collisions may occur in intervals between time steps. One way to prevent this problem is to reduce the sampling period or to enlarge obstacles virtually [Schouwenaars et al. (2004)]. However, these approaches have limitations that the problem becomes too complex to solve on-line since a longer prediction horizon is necessary and that resulting trajectories become conservative, respectively. In order to overcome this problem, [Earl and D'Andrea (2005)] has proposed two methods to guarantee the obstacle avoidance in the interval between prediction time steps. One of the methods reduces the sampling period locally in an interval, where a collision is predicted to occur. The sampling period is repeatedly reduced, until a collision free trajectory is obtained. The other method virtually enlarges an obstacle, with which the vehicle is predicted to collide. Both methods need to solve a mixed integer optimization problem repeatedly, which tend require heavy computational burden.

In this paper, we propose two MPC based trajectory generation methods for a unicycle, which guarantee obstacle avoidance not only at prediction time steps but also in intervals between them. The flrst method reduces the maximum velocity near obstacles. The second one constrains transition of collision avoidance constraints to exclude possibilities of collisions between time steps. Since both methods do not need iteration of optimization, it is expected that the computation time is not signiflcantly increased compared with existing methods. Furthermore we investigate the efiectiveness of the proposed methods by numerical examples and experiments.

2. PROBLEM FORMULATION

2.1 Description of vehicle

We consider a unicycle :

$$x = v\cos\theta, \quad y = v\sin\theta, \quad \theta = \omega, \tag{1}$$

where v and ω are the linear and angular velocities of the vehicle respectively, and (x, y, θ) denotes the measurable coordinate with respect to a global frame (see Fig. 1).

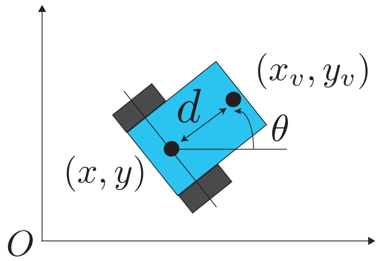


Fig. 1. Description of vehicle

The controlled point of the vehicle is defined as follows:

$$z_v := \begin{matrix} x_v \\ y_v \end{matrix} = \begin{matrix} x + d\cos\theta \\ y + d\sin\theta \end{matrix} . \tag{2}$$

From (1) and (2), we have

$$z_v = Gu, \tag{3}$$
$$G := \begin{matrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \end{matrix} , \quad u := \begin{matrix} v \\ \omega \end{matrix} .$$

The tracking error $e := z_v - z_d$ is now described as $e = Gu$. Since G is always invertible for $d \neq 0$, if we set the control input for the vehicle as $u = G^{-1}\alpha$, then the system is linearized as follows:

$$e = \alpha, \tag{4}$$

where α is a new input for the system.

The goal in this paper is to steer the controlled point z_v to the reference position z_d while avoiding static obstacles described in the following section.

2.2 Description of obstacle

We assume that n obstacles exist in the environment. Each obstacle is approximated by a convex polygon ${}_i (i = 1, \ldots, n)$ surrounding it, as shown in Fig. 2, where n_i denote the number of the edges and ${}_i := [x_i, y_i]^T$ denotes the centroid of the polygon. The obstacle ${}_i$ can be described by

$$(x_v - x_i)\cos\left(\sum_{k=1}^{j}\phi_{ik}\right) + (y_v - y_i)\sin\left(\sum_{k=1}^{j}\phi_{ik}\right) < {}_{ij} \tag{5}$$
$$j = 1, \cdots, n_i,$$

where ${}_{ij}$s are given in advance from the size of the vehicle and the obstacles. We assume that a su– cient condition for collision avoidance between the vehicle and the obstacles is given as follows:

$$\|D^{-1}_{\ \ i}\ {}_i(z_v)\|_\infty \geq 1, D_{\ \ i} := \mathrm{diag}({}_{i1}, \cdots, {}_{in_i}), \tag{6}$$

where ${}_i(z_v) := [\ {}_{i1}(z_v), \cdots, {}_{in_i}(z_v)]^T$, ${}_{ij}(z_v) := (x_v - x_i)\cos(\sum_{k=1}^{j}\phi_{ik}) + (y_v - y_i)\sin(\sum_{k=1}^{j}\phi_{ik})$.

It is known in the literature that (6) can be written as the following linear inequalities [Schouwenaars et al. (2001)] :

$$\begin{aligned} {}_i(z_v) &\geq D_{\ \ i} - MK_i \\ K_i^T\mathbf{1} &\leq n_i - 1 \end{aligned} \tag{7}$$

where $D_{\ \ i} := [\ {}_{i1}, \cdots, {}_{in_i}]^T$, $K_i := [\ {}_{i1}, \cdots, {}_{in_i}]^T$, ${}_{ij}$s are binary integer variables, $\mathbf{1} := [1, 1, \cdots, 1]^T$ and M is a positive number much larger than the possible values of z_v. Another way is to impose the following constraint, which uses the combinations of the binary integer variables [see Kon et al. (2007) for a rectangular obstacle].

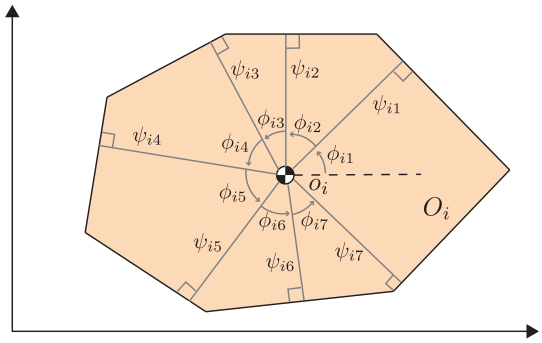


Fig. 2. Description of a convex polygon

$$\begin{aligned} {}_i(z_v) &\geq D_{\ \ i} - G_iK_i - H_i \\ \sum_{k=1}^{m_i} 2^k\ {}_{ik} &\leq 2(n_i - 1), \end{aligned} \tag{8}$$

$$G_i := \begin{bmatrix} g_{11}M & g_{12}M & \cdots & g_{1m_i}M \\ g_{21}M & g_{22}M & \cdots & g_{2m_i}M \\ \vdots & \vdots & \ddots & \vdots \\ g_{n_i1}M & g_{n_i2}M & \cdots & g_{n_im_i}M \end{bmatrix}, \quad H_i := \begin{bmatrix} \sum_{k=1}^{m_i} h_{1k}M \\ \sum_{k=1}^{m_i} h_{2k}M \\ \vdots \\ \sum_{k=1}^{m_i} h_{n_ik}M \end{bmatrix},$$

where $K_i := [\ {}_{i1}, \cdots, {}_{im_i}]^T$ and ${}_{ik}(k = 1, \cdots, m_i)$ are binary integer variables. m_i denotes a number of binary integer variables, which satisfies $2^{m_i - 1} < n_i \leq 2^{m_i}$. $g_{jk}(j = 1, \cdots, n_i)$ is determined by

$$g_{jk} = \begin{matrix} 1 & \text{if} \quad h_{jk} = 0 \\ -1 & \text{otherwise,} \end{matrix} \tag{9}$$

where $h_{jk}(= 0$ or $1)$ satisfies $\sum_{k=1}^{m_i} h_{jk}2^k = 2(j-1)$. Note that if $n_i = 2^{m_i}$, then the second inequality in (8) is always satisfied. Since (8) only requires $m_i(< n_i)$ binary integer variables, the computational burden can be reduced compared with (7). In this paper, we mostly use (8) except for the cases where (7) is necessary for the proposed method in Section 4.1.

Example 1. Here we consider the obstacle approximated by a rectangle as shown in Fig. 3, where $n_i = 4$, ${}_{i1} = {}_{i3}$, ${}_{i2} = {}_{i4}$ and $\phi_{ij} = \frac{(j-1)}{2}(j = 1, 2, 3, 4)$. In this case, the obstacle avoidance constraints in (7) is described as follows:

$$\begin{aligned} x_v - x_i &\leq M\ {}_{i1} - {}_{i1} \\ -x_v + x_i &\leq M\ {}_{i2} - {}_{i1} \\ y_v - y_i &\leq M\ {}_{i3} - {}_{i2} \\ -y_v + y_i &\leq M\ {}_{i4} - {}_{i2} \\ \sum_{j=1}^{4} {}_{ij} &\leq 3. \end{aligned} \tag{10}$$

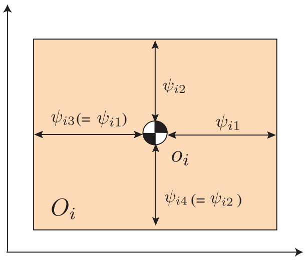


Fig. 3. Rectangular obstacle

Fig. 4 shows the regions allowed by constraints corresponding to $_{ij} = 0$ $(j = 1, 2, 3, 4)$. Note that at most two $_{ij}$s can be 0 at the same time due to geometric constraints. Fig. 4(v)-(viii) show the regions in such cases. On the other hand, the obstacle avoidance constraint in (8) is described as follows:

$$
\begin{aligned}
x_v - x_i &\leq M\ _{i1} + M\ _{i2} - \ _{i1} \\
y_v - y_i &\leq -M\ _{i1} + M\ _{i2} + M - \ _{i2} \\
-x_v + x_i &\leq M\ _{i1} - M\ _{i2} + M - \ _{i1} \\
-y_v + y_i &\leq -M\ _{i1} - M\ _{i2} + 2M - \ _{i2}.
\end{aligned} \tag{11}
$$

In this representation, only Fig. 4 (i)-(iv) are allowed. In other words, by restricting the feasible solution space, (8) reduces the computational burden.

In the rest parts of this paper, we consider the case of all obstacles are approximated by rectangular polygons as shown in Example 1, for simplicity of exposition.

3. TRAJECTORY GENERATION USING MPC

3.1 Optimization problem for MPC

In order to avoid the obstacles, MPC solves the following optimal control problem at every sampling interval , and applies the first component of α.

The trajectory generation problem P_0 at $t = k$ is formulated as a mixed-integer quadratic problem (MIQP) as follows:

MPC based trajectory generation problem

$$\min_{\alpha} \sum_{=k+1}^{k+N} \Big(e(\tau|k)^T Q e(\tau|k) + \alpha(\tau-1|k)^T R \alpha(\tau-1|k) \Big) \tag{12}$$

subject to

$$e(\tau|k) = e(\tau-1|k) + \ \alpha(\tau-1|k),\ e(k|k) = e(t) \tag{13}$$

$$\|D_i^{-1}\ _i(e(\tau|k) + z_d)\|_\infty \geq 1 \tag{14}$$

$$\|\alpha(\tau-1|k)\|_\infty \leq \tag{15}$$

$$\|e(k+N|k)\|_\infty \leq \tag{16}$$

$$i = 1,\ \cdots,\ n,\quad \tau = k+1,\ k+2,\ \cdots,\ k+N,$$

where N denotes the prediction horizon, $e(\tau|k)$ denotes the predicted value of $e(\tau\ )$ at $t = k$, and is a design parameter. $Q \in \mathbf{R}^{2N\ \ 2N}, R \in \mathbf{R}^{2N\ \ 2N}$ are a given symmetric matrix and a given symmetric positive define matrix, respectively.

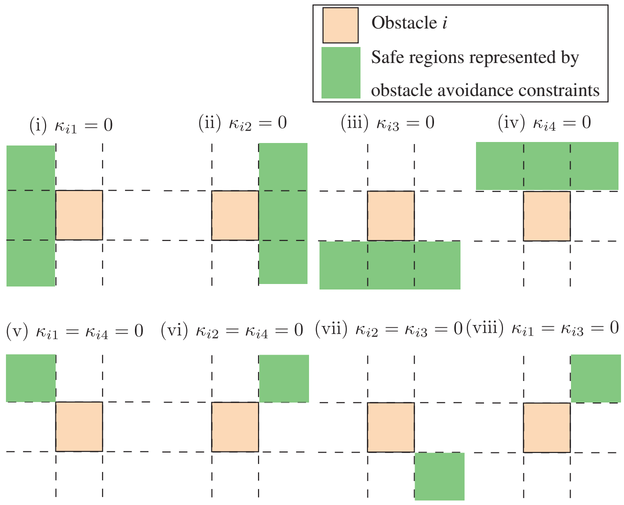


Fig. 4. Possible regions for the constraint (10)

The equality constraint in (13) is a prediction model for e, obtained by discretizing the system in (4) using zero-order hold with sampling interval . The constraint in (14) is equivalent to the obstacle avoidance constraint in (6) which is transformed to (10) or (11). The input u is constrained by (15), which is also used in [Fukushima et al. (2005)]. The terminal constraint in (16) is introduced to guarantee the convergence [Mayne et al. (2000)].

3.2 Limitations of MPC based trajectory generation

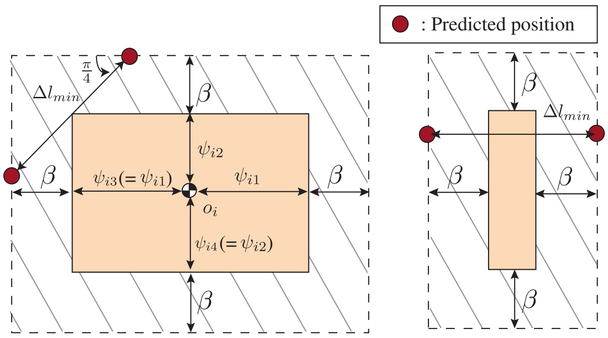
Since the optimization P_0 imposes the obstacle avoidance constraints at discrete time steps, collisions may occur in intervals between prediction time steps. One way to prevent this problem is to enlarge obstacles virtually in the collision avoidance constraints at the discrete time steps, so that the original obstacle avoidance constraint is satisfied not only at the discrete time steps but also in intervals of between them [Schouwenaars et al. (2004)]. The enlargement is quantified by as in Fig. 5. A condition for the amount of enlargement is described in the following lemma.

Lemma 1. Suppose that the obstacle avoidance constraint in (14) is satisfied at prediction time steps $t = \tau$ $(\tau = k+1,\ k+2,\ \cdots,\ k+N)$. Then, a su– cient condition for to guarantee obstacle avoidance in each interval between prediction time steps is given as follows:

$$> \max\ \frac{}{2}, \tag{17}$$

$$:= \max_i \Big(\frac{\overline{2}\delta\eta\ \ 2\min(\psi_{i1}\ \ \psi_{i2})}{2} \Big),\quad i = 1, \cdots, n.$$

Proof. From the prediction model in (13), the trajectory in each interval between two consecutive time steps is described by a line segment, since the input α is constant in the interval. For a line segment which intersects with an original rectangle $_i$, we define the smallest possible length l_{min} of intersection with the enlarged rectangle. Depending on the size of the obstacle, the solution for l_{min} has two possibilities for a fixed value of . One is the case where the line segment does not intersect with the interior of the original obstacle but at a vertex, as shown



(a) Large obstacle (a) Small obstacle

Fig. 5. Enlarged obstacle (shaded area) and l_{min}

in Fig. 5(a). The other is the case where the line segment intersects with the interior of the original obstacle as in Fig. 5(b). Since the predicted position of the vehicle is out of the obstacle at each time step $t = \tau$, collision avoidance is guaranteed between time steps, if l_{min} is not less than the maximum distance $l = \sqrt{\ ^2 + \ ^2} = \sqrt{2}$ that the vehicle can travel for one sampling period.

In the case of Fig. 5(a) where $l_{min} = 2\sqrt{2}$, a su– cient condition is given from $l_{min} \geq l$ as

$$\geq \frac{}{2}. \tag{18}$$

In the case of Fig. 5(b) where

$$l_{min} = 2 \ + 2\min(\ _{i1}, \ _{i2}), \tag{19}$$

a su– cient condition is given from $l_{min} > l$ as

$$> \frac{\sqrt{2} \quad - 2\min(\ _{i1}, \ _{i2})}{2}. \tag{20}$$

By taking into account (18) and (20) for all obstacles, a su– cient condition for is given as (17). ∎

The method based on enlarged obstacles has a limitation that resulting trajectories become conservative if a large is used for all obstacles. Since depends on the sampling period and the maximum velocity of the vehicle as shown in Lemma 1, the value of can be limited by reducing these values. However, if the sampling period or the maximum velocity is significantly decreased for the entire trajectories, the optimization problem typically becomes too complex to solve on-line because a much longer prediction horizon is needed. The methods proposed in [Earl and D'Andrea (2005)] try to locally decrease the sampling period or increase the size of obstacles, based on iterative computation of MIP. In the next section, we propose two methods which guarantee obstacle avoidance between prediction time steps, without iterative computation of MIP.

4. PROPOSED METHODS

4.1 Variable maximum velocity method

The variable maximum velocity method proposed in this section dynamically changes the size of obstacles. In order to guarantee obstacle avoidance between prediction time steps, the maximum velocity of the vehicle is also dynamically changed depending on the size of obstacles, based on Lemma 1. Furthermore, to prevent from generating an extremely conservative trajectory, the size of obstacles is chosen such that the cost function for the trajectory is not significantly increased.

More precisely, for a constant satisfying the condition in Lemma 1, the obstacle avoidance constraint in (14) is modified as

$$\begin{aligned} &\|D^{-1}_{\ i}(\ (\tau|k)) \ _i(e(\tau|k) + z_d)\|_\infty \geq 1 \\ &D \ _i(\ ) := D \ _i + \ (1 - \ )I_{n_i}. \end{aligned} \tag{21}$$

which implies that the length of the rectangle sides is increased by $2 \ (1 - \ (\tau|k))$, where a new variable $(\tau|k) \in [0, \ 1]$ is introduced to adjust the size of obstacles. Since the size of obstacles is modified, the maximum velocity , i.e. the upper bound of $\alpha(\tau-1|k)$ for $t \in ((\tau-1) \ , \tau \ )$ also needs to be modified based on the relationship in Lemma 1. However, the size of obstacles are different at $(\tau - 1)$ and τ unlike Lemma 1. Therefore, the input constraint is modified taking into account the worse case where enlarged obstacle is smaller:

$$\|\alpha(\tau - 1|k)\|_\infty \leq \ (1 - \max\{ \ (\tau - 1|k), \ (\tau|k)\}). \tag{22}$$

By decreasing , the size of obstacles is increased, which typically increases the cost function in (12) and makes the trajectory conservative. In order to prevent from generating an extremely conservative trajectory, is chosen by an optimization problem taking into account the cost function in (12).

The optimization problem P_1 at $t = k$ is described as follows:

Variable maximum velocity method

$$\min_{\alpha,\nu} \sum_{=k+1}^{k+N} \Big(e(\tau|k)^T Q e(\tau|k) + \alpha(\tau-1|k)^T R \alpha(\tau-1|k) + \ (\tau - 1|k)^T S \ (\tau - 1|k) \Big) \tag{23}$$

subject to *eq.* (13), (16)

$$\|D^{-1}_{\ i}(\ (\tau|k)) \ _i(e(\tau|k) + z_d)\|_\infty \geq 1 \tag{24}$$

$$\|\alpha(\tau - 1|k)\|_\infty \leq \ (1 - \ (\tau - 1|k)) \tag{25}$$

$$\|\alpha(\tau - 1|k)\|_\infty \leq \ (1 - \ (\tau|k)) \tag{26}$$

$$0 \leq \ (\tau - 1|k) \leq 1 \tag{27}$$

$$i = 1, \ \cdots, \ n \qquad \tau = k + 1, \ k + 2, \ \cdots, \ k + N,$$

where $S \in \mathbf{R}^{N \ N}$ is a given symmetric matrix. The constraint (24) is same as (21). The constraints (25) and (26) are equivalent to (22).

The variable maximum velocity method has the following property.

Theorem 2. Assume z_v satisfies the obstacle avoidance constraint at $t = k$, and satisfies (17). Then, a feasible solution of the optimization P_1 is a collision free trajectory at $t \in [k \ , (k + N) \ ]$.

Proof. It is su– cient to show the collision avoidance at $t \in ((\tau-1)\ ,\tau\ )$, since no collision occurs at $t = \tau$ ($\tau = k+1,\ k+2,\ \cdots,\ k+N$) and $t = k$ from the assumption.

Similarly to the proof in Lemma 1, the collision avoidance at $t \in ((\tau-1)\ ,\tau\ )$ is guaranteed, if $l_{min} \geq\ l\ (\ l_{min} >\ l$ in the case of Fig. 5(b)). However, the constraint in (24) implies that the size of enlarged obstacles is difierent at $t = (\tau-1)$ and $t = \tau$. Thus, we need to show $l_{min} \geq\ l(\ l_{min} >\ l$ in the case of Fig. 5(b)) for both $(1-\ (\tau-1|k))$ and $(1-\ (\tau|k))$. We deflne $l_{min}(\tau-1)$ and $l_{min}(\tau)$ as the values of l_{min} for $(1-\ (\tau-1|k))$ and $(1-\ (\tau|k))$. Without loss of generality, we assume that $(\tau|k) \geq\ (\tau-1|k)$. From the input constraint in (22), the maximum velocity is $(1-\ (\tau|k))$ which implies that $l = \sqrt{2}\ \ (1-\ (\tau|k))$.

In the case of Fig. 5(a), since we have

$$l_{min}(\tau-1) \geq\ l_{min}(\tau) = 2\sqrt{2}\ \ (1-\ (\tau|k)), \quad (28)$$

it is su– cient to show $l_{min}(\tau) \geq\ l$. It holds

$$l_{min}(\tau) -\ l = \sqrt{2}(2\ \ -\ \ )(1-\ (\tau|k)) \geq 0, \quad (29)$$

since $\ >\ /2$ from the assumption, and $(\tau|k) \leq 1$ due to (27). In the case of Fig. 5(b), since we have

$$\begin{aligned} l_{min}(\tau-1) &\geq\ l_{min}(\tau) \\ &= 2\ \ (1-\ (\tau|k)) + 2\min(\ _{i1},\ \ _{i2}), \end{aligned} \quad (30)$$

it is su– cient to show $l_{min}(\tau) >\ l$. It holds

$$\begin{aligned} &l_{min}(\tau) -\ l = 2\ \ (\tau|k)\min(\ _{i1},\ \ _{i2}) \\ &+ \left(2\ \ -\sqrt{2}\ \ + 2\min(\ _{i1},\ \ _{i2})\right)(1-\ (\tau|k)) > 0, \end{aligned}$$

since (20) is satisfled from the assumption, and $0 \leq\ (\tau|k) \leq 1$ due to (27). Therefore, the collision avoidance at $t \in ((\tau-1)\ ,\tau\ )$ is guaranteed, which completes the proof. ∎

Remark 1. The variable maximum velocity method can be applied to more complex obstacle, if we can obtain the su– cient condition for in the same manner as Lemma 1.

4.2 Constrained transition method

Next we propose another method which focuses on transition of active constraints for obstacle avoidance.

As mentioned in Section. 2.2, the collision avoidance constraint in (10) has 8 modes switched depending on the values of binary integer variables. Since the mode transition between consecutive time steps is not constrained in existing methods, any transition is allowed as shown in Fig. 6(a). For instance, (ii)→(viii) and (ii)→(vi) in Fig. 6(a) possibly cause a collision between time steps.

The constrained transition method chooses only safe transition conditions as shown in Fig. 6(b), by restricting $_{ij}$ in the constraint (10). For example, from the region (ii) corresponding to $_{i1} = 0$ and $_{ij} = 1$ ($j \neq 1$), the vehicle can move to a region (i), (ii) or (iii) where $_{i1} = 0$. From (i) corresponding to $_{i1} =\ _{i4} = 0$ and $_{i2} =\ _{i3} = 1$, the vehicle can move to a region (i), (ii), (iii), (vii) or (viii),

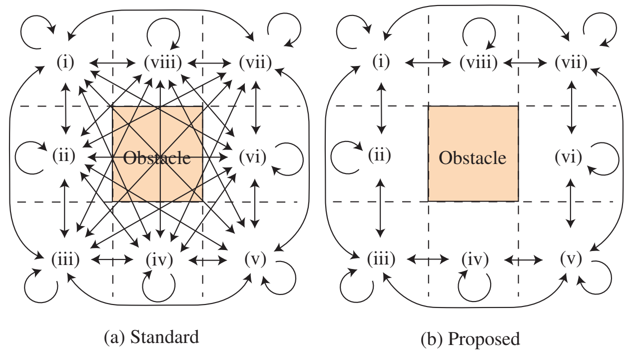


Fig. 6. Transition of the obstacle avoidance constraints

where $_{i1} = 0$ or $_{i4} = 0$. In other words, the transition conditions in Fig. 6(b) can be described as follows:

$$_{ij} =\ _{ij} = 0, \quad \exists j\ (=1,2,3,4), \quad (31)$$

where $_{ij}$ denotes the value of $_{ij}$ at one step before. The condition in (31) can be realized as follows:

$$\begin{aligned} _{i1} +\ _{i1} &\leq M\ _{i1} + M\ _{i2} \\ _{i2} +\ _{i2} &\leq -M\ _{i1} + M\ _{i2} + M \\ _{i3} +\ _{i3} &\leq M\ _{i1} - M\ _{i2} + M \\ _{i4} +\ _{i4} &\leq -M\ _{i1} - M\ _{i2} + 2M. \end{aligned} \quad (32)$$

where $_{i1}$ and $_{i2}$ are new binary variables.

The optimization problem P_2 for the constrained transition method is described as follows:

Constrained transition method

$$\min_{\alpha} \sum_{=k+1}^{k+N} \left(e(\tau|k)^T Q e(\tau|k) + \alpha(\tau-1|k)^T R \alpha(\tau-1|k) \right)$$

subject to *eq.* (13), (15), (16)

$$\begin{bmatrix} e(\tau|k) + z_d -\ _i \\ -e(\tau|k) - z_d +\ _i \end{bmatrix} \leq M \begin{bmatrix} _{i1}(\tau|k) \\ _{i3}(\tau|k) \\ _{i2}(\tau|k) \\ _{i4}(\tau|k) \end{bmatrix} - \begin{bmatrix} _{i1} \\ _{i2} \\ _{i1} \\ _{i2} \end{bmatrix} \quad (33)$$

$$\sum_{j=1}^{4}\ _{ij}(\tau|k) \leq 3$$

$$\begin{bmatrix} _{i1}(\tau-1|k) +\ _{i1}(\tau|k) \\ _{i2}(\tau-1|k) +\ _{i2}(\tau|k) \\ _{i3}(\tau-1|k) +\ _{i3}(\tau|k) \\ _{i4}(\tau-1|k) +\ _{i4}(\tau|k) \end{bmatrix} \leq \begin{bmatrix} M & M \\ -M & M \\ M & -M \\ -M & -M \end{bmatrix} \begin{bmatrix} _{i1}(\tau|k) \\ _{i2}(\tau|k) \end{bmatrix} + \begin{bmatrix} 0 \\ M \\ M \\ 2M \end{bmatrix} \quad (34)$$

$$i = 1,\ \cdots,\ n,\quad \tau = k+1,\ k+2,\ \cdots,\ k+N,$$

where $_{ij}(k|k)$ is determined by $e(k|k) = e(k\ )$ such that (33) is satisfled for $\tau = k$. The inequality constraint (33) is the collision avoidance constraints in (10), and (34) is the transition constraint in (32).

The constrained transition method has the following property.

Theorem 3. Assume z_v satisfles the obstacle avoidance constraint at $t = k$. Then, a feasible solution of the

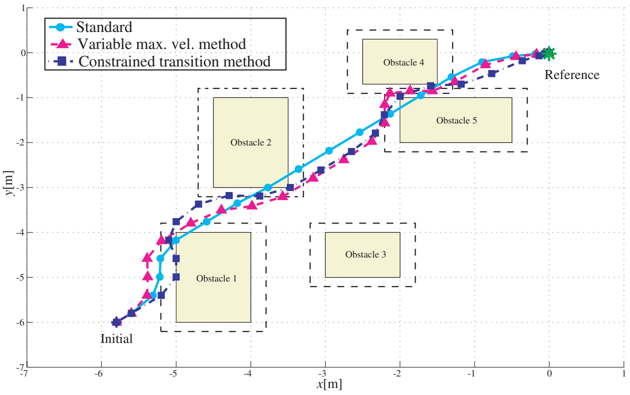


Fig. 7. Path of the vehicle in simulation

optimization P_2 is a collision free trajectory in $t \in [k\ ,(k+N)\ ]$.

Proof. A feasible solution of the optimization P_2 satisfies the collision avoidance constraint at $t = \tau$. From the assumption, the collision avoidance constraint is also satisfied at the initial time $t = k$. Due to the constraint in (34), the transition from $t = (\tau - 1)$ to $t = \tau$ of the region, to which the vehicle belongs, is restricted to the ones in Fig. 6(b). Therefore, since the trajectory for $t \in [(\tau - 1)\ ,\tau\ ]$ is described by a line segment, no collision occurs at $t \in [k\ ,(k+N)\ ]$. ∎

Remark 2. The constrained transition method can be easily applied to the convex polygonal obstacle, since by changing (31) as follows:

$$_{ij} = \quad _{ij} = 0, \exists j(=1,\cdots,n_i). \tag{35}$$

5. NUMERICAL EXAMPLES

In this section, we investigate the effectiveness of two methods proposed in Section. 4.1 and Section. 4.2 by numerical examples. The MIQP problems are solved by CPLEX [ILOG (2006)] on a PC (CPU : Intel Core2 Duo 2.0GHz, RAM : 2GB).

The initial coordinate of the vehicle is given as $(-6,-6,0)$, while the reference position is $(0,0)$ in the global frame. Other parameters are given as $d = 0.2$, $= 0.41$, $= 0.205$, $Q = I$, $R = I$, $S = I$, $M = 30$, $N = 20$, $= 1.0$ and $_i = 0.5$. Five obstacles are placed as shown in Fig. 7. The dashed lines around the obstacles in Fig. 7 show the region where the maximum velocity is decreased in the variable maximum velocity method.

The solid line in Fig. 7 shows the trajectory generated by a standard MPC method taking into account collision avoidance only at discrete time steps. It is seen that the trajectory in this case violates the collision avoidance constraints between time steps. The dashed and dash-dotted lines show the trajectories by the variable maximum velocity method and the constrained transition method, respectively. These trajectories show that the obstacle avoidance constraints are satisfied not only at prediction time steps but also in the intervals between them.

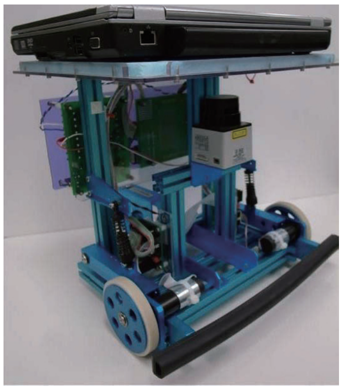


Fig. 8. Vehicle for experiment

6. EXPERIMENTS

Two methods proposed in Section. 4.1 and Section. 4.2 are applied to a mobile robot platform beego (Techno Craft), which is a two wheeled skid-steer robot with one caster wheel (Fig. 8). The coordinate of the vehicle is measured by dead reckoning, and the position and size of the obstacles are given in advance. Trajectory generation methods are implemented using Matlab engine and CPLEX. The specification of the PC used for experiments is CPU : Intel Core2 Duo 2.0GHz, RAM : 2GB.

The initial coordinate of the vehicle is given as $(-3.95, 0, 0)$, while the reference position is $(0,0)$ in the global frame. Other parameters are $d = 0.2$, $= 0.41$, $= 0.205$, $Q = I$, $R = I$, $S = I$, $M = 30$, $N = 12$, $= 1.0$ and $_i = 0.5$. Two obstacles are placed as shown in Fig. 9. The dashed lines around the obstacles in Fig. 9 show the region where the maximum velocity is decreased in the variable maximum velocity method.

The solid line in Fig. 7 shows that a collision occurred in the case of a standard MPC method taking into account collision avoidance only at discrete time steps. On the other hand, no collision occurs when the proposed methods are applied. A feature of the variable maximum velocity method shown both in simulations and experiments is that the vehicle tends to keep more distance from the obstacles since the maximum velocity is decreased near the obstacles. Fig. 10 shows the time plots of the control input, i.e. the velocity of the vehicle. It can be seen that the velocity of the vehicle is reduced in the interval from 3 [sec] to 8 [sec] when the vehicle runs near the obstacles.

Note that the maximum computation time to solve optimization problem is $0.296[sec]$ for the conventional method, $0.170[sec]$ for the variable maximum velocity method and $0.199[sec]$ for the constrained transition method.

7. CONCLUSION

In this paper, we have proposed two MPC based trajectory generation methods for a unicycle, which guarantee obstacle avoidance not only at prediction time steps but also in intervals between them. The first method reduces the maximum velocity near obstacles. The second one constrains transition of collision avoidance constraints to exclude possibilities of collisions between time steps. The

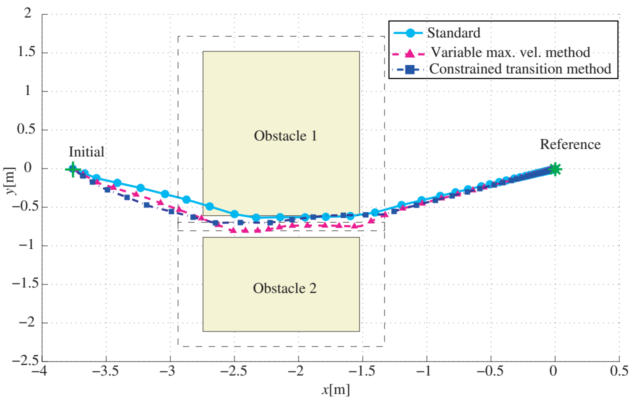


Fig. 9. Path of the vehicle in experiment

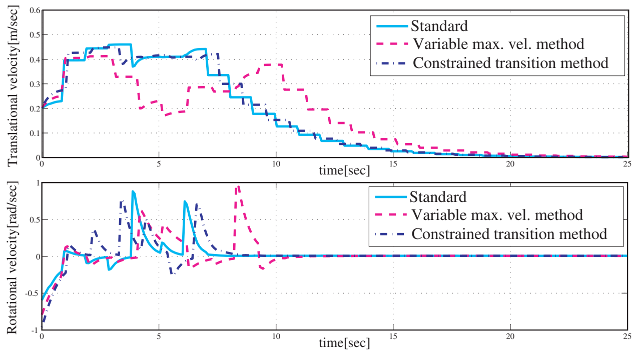


Fig. 10. Velocity input of the vehicle in experiment

first one can describe the problem with fewer binary variables than the second one, and can adjust the safety of the trajectory by changing the weighting matrix. The second one can describe the problem without enlarging the obstacle. Furthermore, the effectiveness of the proposed methods has been investigated by numerical examples and experiments.

REFERENCES

Earl, M.G. and D'Andrea, R. (2005). Iterative milp methods for vehicle-control problems. *IEEE Trans. on Robotics*, 21(6), 1158 1167.

Ferguson, D. and Stentz, A. (2006). Using interpolation to improve path planning: The field d* algorithm. *Journal of Field Robotics*, 23(2), 79 101.

Fukushima, H., Kon, K., and Matsuno, F. (2005). Distributed model predictive control for multi-vehicle formation with collision avoidance constraints. *IEEE Conf. on Decision and Control*, 5480 5485.

Gomez-Ortega, J. and Camacho, E.F. (1995). Neural network mbpc for mobile robot path tracking. *Journal of Robotics and Computer-Integrated Manufacturing*, 11(4), 271 278.

Guo, Y. and Tang, T. (2008). Optimal trajectory generation for nonholonomic robots in dynamic environments. *IEEE Int. Conf. on Robotics and Automation*, 2552 2557.

ILOG (2006). *ILOG CPLEX 10.1 User s manual*.

Klancar, G. and Skrjanc, I. (2007). Tracking-error model-based predictive control for mobile robots in real time. *Journal of Robotics and Autonomous Systems*, 55, 460 469.

Kon, K., Fukushima, H., and Matsuno, F. (2007). Multi-vehicle formation control based on branch-and-bound method compatible with collision avoidance problem. *IEEE Conf. on Intelligent Robots and Systems*, 3771 3776.

Kopfstedt, T., Mukai, M., Fujita, M., and Sawodny, O. (2006). A networked formation control for group of mobile robots using mixed integer programming. *IEEE Int. Conf. on Control Applications*, 579 584.

Kuwata, Y., Richards, A., Schouwenaars, T., and How, J.P. (2007). Distributed robust receding horizon control for multivehicle guidance. *IEEE Int. journal of Control Systems Technology*, 15(4), 627 641.

Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Proc. of American Control Conference*, 36, 789 814.

Richards, A. and How, J. (2003). Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility. *Proc. of American Control Conference*, 4034 4040.

Schouwenaars, T., How, J., and Feron, E. (2004). Receding horizon path planning with implicit safety guarantees. *Proc. of American Control Conference*, 5576 5581.

Schouwenaars, T., Moor, B.D., Feron, E., and How, J. (2001). Mixed integer programming for multi-vehicle path plannning. *Procedings of the European Control Conference*, 2603 2608.

Xi, W. and Baras, J.S. (2007). Mpc based motion control of car-like vehicle swarms. *Proc. of Mediterranean Conference on Control and Automation*, T33 005.