SICE-ICASE International Joint Conference 2006
Oct. 18-21, 2006 in Bexco, Busan, Korea

Constrained Model Predictive Control: Applications to Multi-Vehicle Formation and an Autonomous Blimp

Hiroaki Fukushima[1], Kazuyuki Kon[1], Fumitoshi Matsuno[1],
Yasushi Hada[2], Kuniaki Kawabata[2], and Hajime Asama[3]

[1] The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
[2]RIKEN (The Institute of Physical and Chemical Research), Hirosawa 2-1, Wako, Saitama 351-0198, Japan
[3]Research into Artifacts, Center for Engineering, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba 277-8568, Japan

Abstract: Model predictive control (MPC) is one of the few ways to handle input and output constraints explicitly. Although MPC has been widely applied to chemical process control, the effectiveness for mechanical systems with shorter time constants has not been fully investigated. The goal of this paper is to investigate the applicability of model predictive control to mechanical systems based on experimental examples of multi-vehicle formation and autonomous blimp control.

Keywords: Constrained system, mixed integer programming, multi-parametric programming

1. INTRODUCTION

Model predictive control (MPC) is one of the few ways to handle input and output constraints explicitly[1]. Also, recent studies suggest that MPC is possiblly effective for hybrid systems. Although MPC has been widely applied to chemical process control, the effectiveness for mechanical systems with shorter time constants has not been fully investigated. The goal of this paper is to investigate the applicability of model predictive control to mechanical systems based on experimental examples of multi-vehicle formation and autonomous blimp control.

Section 2 describes an application example of MPC to an autonomous blimp. Lighter Than Air vehicles have attracted much attention due to their potential utilization in surveillance, exploration, transportation and so on [2][3]. One of the key issues in flight control system design is how to deal with input constraints due to actuator saturations and output limitations for safety reasons. We take into account inequality constraints on position, velocity and thrust of the blimp in MPC design, in order to keep the blimp within the prescribed regions. To this end, a simple linear model connected to a deadzone nonlinearity is constructed based on experimental data. Then, MPC controllers are derived offline as piecewise affine state feedback laws based on a robust MPC approach[4] to take into account additive uncertainties. We show an example of indoor experiments using the MPC controllers obtained based on the simple model. See [5] for more examples.

Section 3 deals with an MPC problem of a multi-vehicle formation. Coordinated control of multiple vehicles has been a significant field of research in recent years[6]-[8]. One of the challenging issues in formation control is collision avoidance between vehicles. Since the collision avoidance constraints can be described by linear inequality constraints including integers, on-line optimal control problems in MPC are solved by using mixed-integer programming. In this paper, we show an experimental example of a distributed MPC method for a group of unicycles proposed in [9]. This method first stabilizes the system by using feedback linearization, and then a collision avoidance method based on MPC is applied to the linearized system.

2. MPC OF AUTONOMOUS BLIMP

2.1 Outline of the experimental system

The unmanned blimp for our indoor experiments is depicted in Fig. 1. It has inertial sensors (three accelerometers, three magnetic gyroscopes, and an axis meter), actuators, a Linux PC (CPU: Intel 486dx2), and batteries. It can communicate with a ground base using wavelan (IEEE 802.11b). The actuator consists of two cross-shaped vectored thrusters in the center of balance and two thrusters on the tail. Localization of the blimp is based on the inertial sensors of the blimp and a position measurement system on the ground. See [10] for the detail of the experimental system.

Fig. 1 Blimp for experiments

89-950038-5-5 98560/06/$10

In this paper, we forcus on a position control problem, and it is assumed that the yaw angle, which is controlled independently by a given feedback law, is sufficiently small The set point and the initial point can be defined on the global frame as

$$(x_r(t), y_r(t), z_r(t)) = (0, 0, 0) \quad (1)$$
$$(x(0), y(0), z(0)) = (x_0, y_0, z_0), \quad (2)$$

respectively, where x_0, y_0, z_0 are given constant numbers. For the safety reasons, upper and lower bounds of the position and velocity of the blimp are given as constraints. We define control input variables u_x, u_y, u_z as thrust levels in x, y, z directions, respectively. By taking accout of the saturation levels, control input constraints are given as

$$|u_x(t)| \le 100, \ |u_y(t)| \le 70, |u_z(t)| \le 70. \quad (3)$$

Double integrator models in x, y, z directions are separately constructed using step response data.

2.2 Controller design

Based on double integrator models, the following discrete-time state space models are constructed using the zero-order-hold discretization with sampling time $T_s = 0.8$:

$$\xi_x[k+1] = A\xi_z[k] + B_1 u_x[k], \quad (4)$$
$$\xi_y[k+1] = A\xi_y[k] + B_2 u_y[k], \quad (5)$$
$$\xi_z[k+1] = A\xi_z[k] + B_3 u_z[k], \quad (6)$$

where $\xi_x := [x, \dot{x}]^T, \xi_y := [y, \dot{y}]^T, \xi_z := [z, \dot{z}]^T$. We independently design a controller for each model in (4)-(6). Note that $\dot{x}$, $\dot{y}$ and $\dot{z}$ are estimated from the position measurements x, y and z, since the experimental system has no velocity sensor.

An example for $x_0 = -18$, $y_0 = 0$, $z_0 = -4$ is shown in this paper. See [5] for other examples. The following constraints are given for the position and velocity:

$$x(t) \le 0.5, \ |y(t)| \le 0.5, \ z(t) \le 0.5 \text{ [m]} \quad (7)$$
$$|\dot{x}(t)| \le 0.8, \ |\dot{y}(t)| \le 0.8, \ |\dot{z}(t)| \le 0.8 \text{ [m/s]}. \quad (8)$$

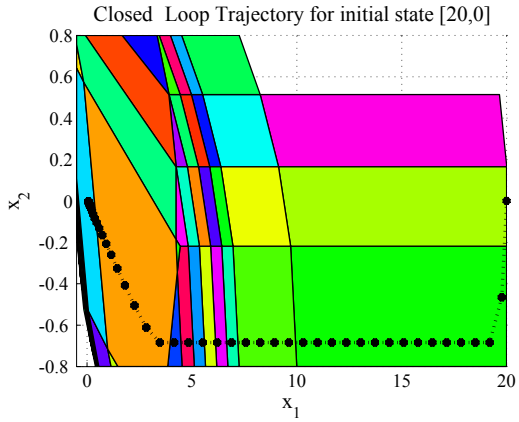

Fig. 2 Partitions in state space for MPC

First, an LQ feedback controller

$$u_x = -K_1 \xi_x, \quad K_1 = [31, 183] \quad (9)$$

is obtained by using the weighting matrices $Q := \text{diag}\{100, 2000\}$, $R := 0.1$ for ξ_x and u_x, respectively. The performance of linear feedback controllers as in (9) largely depends on the initial states for constrained systems. In this example, $\dot{x}$ violates the velocity constraint in (8), and goes beyond the measurement ablity of the position sensor. To take into account the constraints, we modify the feedback law as $u_x = -K_1\xi + v_x$, and choose v_x by MPC. Thus, the following closed-loop model is used for MPC

$$\xi_x[k+1] = A_{c1}\xi_x[k] + B_1 v_x[k], \quad (10)$$
$$A_{c1} := A - B_1 K_1 \quad (11)$$

instead of (4).

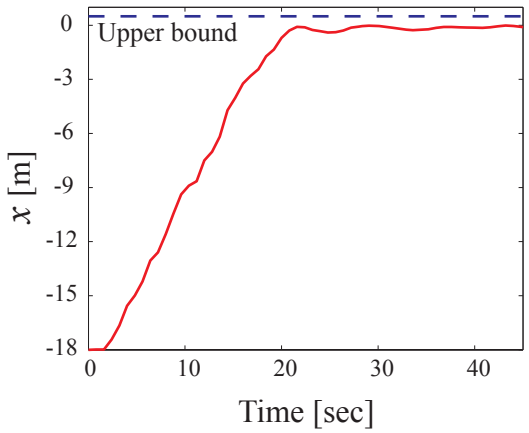

Fig. 3 Time plot of x for $x_0 = -18$

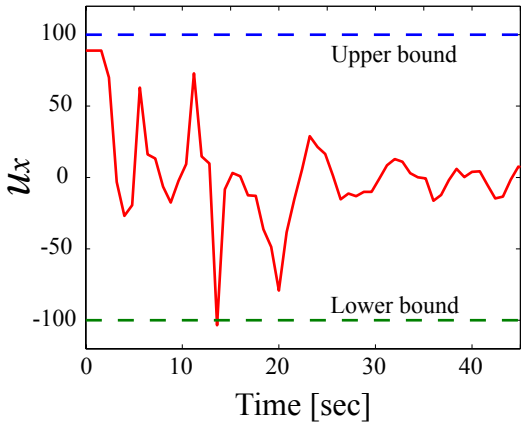

Fig. 4 Time plot of u_x for $x_0 = -18$

MPC methods typically determine control input based on finite horizon open-loop control optimization problems. Our optimization problem at k to determine $v_x[k]$ in (10) is

$$\min_{v_\tau} \sum_{\tau=0}^{N-1} v_\tau^2 \quad (12)$$

subject to

$$\xi_{\tau+1} := A_{c1}\xi_\tau + B_1 v_\tau, \quad \xi_0 := \xi_x[k] \quad (13)$$
$$\underline{\xi} \le \xi_\tau \le \overline{\xi}, \quad \underline{u} \le -K_1\xi_\tau + v_\tau \le \overline{u}, \quad (14)$$

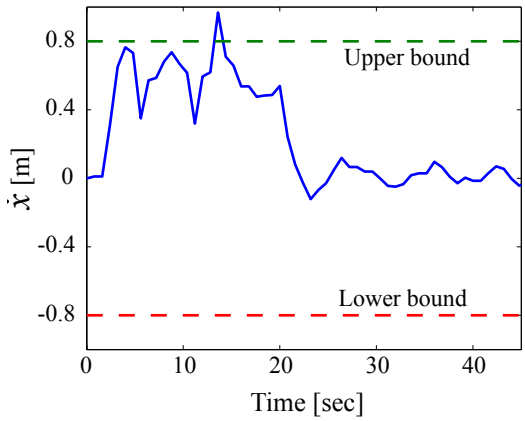

Fig. 5 Time plot of $\dot{x}$ for $x_0 = -18$

where $\underline{\xi}, \overline{\xi}, \underline{u}, \overline{u}$ are the upper and lower bounds given in (7)-(8) and (3). The first element v_0 of the optimal solution is applied at each time step k. While traditional MPC methods solve online the optimization problem as described above, recent methods[11][12] can solve offline the problem above as the following piecewise affine feedback low:

$$v_x[k] = F_r \xi_x[k] + G_r, \quad if \quad \xi_x[k] \in \mathcal{P}_r \tag{15}$$

$$\mathcal{P}_r = \{\xi \in \mathbf{R}^2 | \ H_r \xi \leq E_r\}, \quad r = 1, \cdots, N_p, \tag{16}$$

where N_p is the number of the polytope regions. In order to take account of disturbances and modeling errors, we adopt a robust MPC approach[4], which deals with the additive uncertainty $w_x[\tau] \in \mathcal{W}$ as

$$\xi_x[\tau + 1] = A_{c1} \xi_x[\tau] + B_1 v_x[\tau] + w_x[\tau]. \tag{17}$$

The polytope $\mathcal{W}$ is simply chosen as

$$\mathcal{W} := \{w \in \mathbf{R}^2 | \ |w| \leq B_1 \eta\} \tag{18}$$

to consider input disturbances, where η is a design parameter. See [13]-[15] for more details on robust MPC approaches.

Fig. 2 shows partitions $\mathcal{P}_r$ for $\eta = 25$ in (18) and a trajectory of ξ_x of the double integrator model for $x_0 = -18$, which is obtained using Multi-Parametric Toolbox for MATLAB[16]. From the figure, it can be seen that the given constraints are satisfied with some margin, since the controller takes account of the disturbance in (17).

The MPC controller for $\eta = 25$ mentioned above is applied to the blimp system. Solid lines in Fig. 3-5 show respectively the trajectories of x, u_x and an estimate of $\dot{x}$. Fig. 3-4 show that the MPC controller satisfies the given constraints. Although the estimate of $\dot{x}$ slightly violates the constraint because it is considerably noisy as shown in Fig. 5, it can be seen that the maximum velocity is kept around the given value 0.8. Note that although MPC is applied to the control of y and z in the same way, MPC is not essential in control of y and z in this example, since the constraints are not tight for small initial deviations.

3. MULTI-VEHICLE FORMATION

3.1 Problem formulation

We consider a group of n unicycles indexed by $i = 1, \cdots, n$:

$$\dot{x}_i = v_i \cos\theta_i, \quad \dot{y}_i = v_i \sin\theta_i, \quad \dot{\theta}_i = \omega_i, \tag{19}$$

where v_i and ω_i are the linear and angular velocities of the vehicle i respectively, and (x_i, y_i, θ_i) denotes the measurable coordinate with respect to a global frame (see Fig. 6).

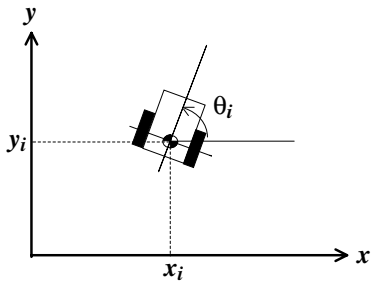

Fig. 6 Wheeled vehicle

We also define a leader vehicle described as:

$$\dot{x}_r = v_r \cos\theta_r, \quad \dot{y}_r = v_r \sin\theta_r, \quad \dot{\theta}_r = \omega_r, \tag{20}$$

where v_r and ω_r are the linear and angular velocities respectively, and (x_r, y_r, θ_r) denotes the measurable coordinate with respect to the global frame.

The reference position of the vehicle i in (19) is given as a constant vector (r_i, l_i) in a local frame on the leader vehicle in (20) (see Fig. 7). In other words, the reference trajectory for the vehicle i is given with respect to the global frame as

$$z_i^d := \begin{bmatrix} x_r + r_i \sin\theta_r + l_i \cos\theta_r \\ y_r - r_i \cos\theta_r + l_i \sin\theta_r \end{bmatrix}. \tag{21}$$

We refer to the vehicle $i (= 1, \cdots, n)$ in (19) as the "follower i", and the leader vehicle in (20) as the "leader".

Our goal is to control the each follower's position with a given offset d, defined as

$$z_i := \begin{bmatrix} x_{vi} \\ y_{vi} \end{bmatrix} = \begin{bmatrix} x_i + d\cos\theta_i \\ y_i + d\sin\theta_i \end{bmatrix}, \tag{22}$$

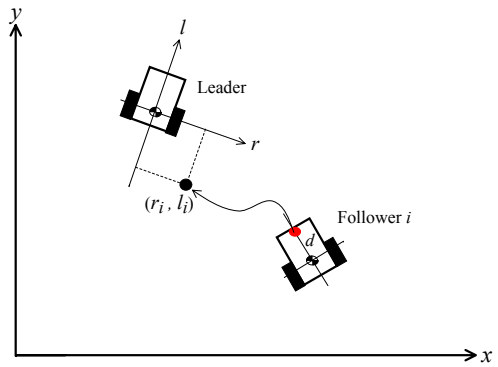

Fig. 7 Leader and follower

to the reference trajectory z_i^d in (21) without collisions. We assume that a sufficient condition for collision avoidance between the followers i and j is given from the size of the vehicles as follows:

$$\|z_i - z_j\|_\infty \geq \psi, \quad \forall j \neq i. \tag{23}$$

Note that it is known in the literature that the collision avoidance constraint in (23) can be written as the following linear constraints[8]

$$\begin{aligned} x_{vi} - x_{vj} &\leq \Psi\kappa_{ij1} - \psi \\ y_{vi} - y_{vj} &\leq \Psi\kappa_{ij2} - \psi \\ -x_{vi} + x_{vj} &\leq \Psi\kappa_{ij3} - \psi \\ -y_{vi} + y_{vj} &\leq \Psi\kappa_{ij4} - \psi \\ \sum_{p=1}^{4} \kappa_{ijp} &\leq 3 \end{aligned} \tag{24}$$

including binary variables κ_{ijp} (= 0 or 1), where Ψ is a positive number much larger than the possible values of z_i. In the same way, it is possible to take account of the collisions with the leader. However, we do not incorporate the collision avoidance with the leader into the control problem, since the leader could be given only virtually in some applications.

In this paper, we propose an algorithm for the followers to determine (v_i, ω_i) steering z_i to z_i^d without collisions. This method first stabilizes the system by using feedback linearization as described in Section 3.2and then a collision avoidance method based on a distributed MPC in Section 3.3 is applied to the linearized system. For implementation of this method, we assume that the vehicles can communicate necessary information on the future trajectories predicted in their optimization problems, as in the existing works [8][7]. Note that we also need to assume that the followers know the future trajectory of $\theta_r(t)$ $(t \leq \tau \leq t+T)$ for a given constant T, when the collision avoidance method is applied.

3.2 Feedback linearization

In this section, we describe the feedback linearization method. The basic idea is similar to the one in [6]. However, we need to modify the method in [6] to take account of the collision avoidance between the followers.

From (20) and (21), we have

$$\dot{z}_i^d = F_i u_r, , \quad u_r := \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \tag{25}$$

$$F_i := \begin{bmatrix} \cos\theta_r & r_i\cos\theta_r - l_i\sin\theta_r \\ \sin\theta_r & r_i\sin\theta_r + l_i\cos\theta_r \end{bmatrix}. \tag{26}$$

It is also seen from (19) and (22)

$$\dot{z}_i = G_i u_i, \tag{27}$$

where

$$G_i := \begin{bmatrix} \cos\theta_i & -d\sin\theta_i \\ \sin\theta_i & d\cos\theta_i \end{bmatrix}, \quad u_i := \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}. \tag{28}$$

The tracking error $e_i := z_i - z_i^d$ is now described as

$$\dot{e}_i = G_i u_i - F_i u_r. \tag{29}$$

By applying

$$u_i = G_i^{-1}(-\lambda e_i + F_i u_r + \alpha_i) \tag{30}$$

to (29) we have

$$\dot{e}_i = -\lambda e_i + \alpha_i, \tag{31}$$

where $\lambda > 0$ is a design parameter. Note that G_i is always invertible for $d > 0$, since

$$\det G_i = d\cos^2\theta_i + d\sin^2\theta_i = d.$$

3.3 Model predictive collision avoidance

In the proposed method, each follower sequentially solves an optimal control problem at every update interval δ. More precisely, the follower i solves the optimal control problem at $t = k_i\delta$ for $k_i := sn + i - 1$ $(s = 0, 1, \cdots)$, and apply the optimal control trajectory α_i^* until its next update time $t = (k_i+n)\delta$. The predicted value of $e_i(\tau)$ at $t = k\delta$ is defined as $\widehat{e}_i(\tau|k)$, which is transmitted to all other followers as soon as the optimal control problem at $t = k\delta$ is solved. The vehicle i uses the predicted trajectories $\widehat{e}_j$ transmitted from other vehicles j $(j \neq i)$ to take account of collision avoidance.

The algorithm for the vehicle i is described as follows:

Algorithm for the follower i:

Step 0: At the initial time $t = 0$, define

$$k := 0 \tag{32}$$

$$\widehat{\alpha}_i(\tau|0) := \lambda e_i(0), \qquad 0 \leq \tau \leq T \tag{33}$$

$$\widehat{e}_j(\tau|0) := e_j(0), \qquad 0 \leq \tau \leq T, \tag{34}$$

for each $j \neq i$.

Step 1: At $t = k\delta$,

- If $k = i - 1 \pmod n$, solve the optimization described below, update $\widehat{\alpha}_i$ and $\widehat{e}_i$ as

$$\widehat{\alpha}_i(\tau|k) = \alpha_i^*(\tau|k) \tag{35}$$

$$\widehat{e}_i(\tau|k) = e_i^*(\tau|k) \qquad t \leq \tau \leq t+T, \tag{36}$$

  and transmit $\widehat{e}_i(\tau|k)$, where $\alpha_i^*(\tau|k)$ and $e_i^*(\tau|k)$ denote the optimal trajectories obtained from the optimization below.
- Otherwise, receive $\widehat{e}_p(\tau|k)$ from the vehicle p, where $p = k \pmod n$.

Step 2: Apply $u_i(\tau)$ in (30) for $t \leq \tau \leq t+\delta$ using

$$\alpha_i(\tau) = \widehat{\alpha}_i(\tau|k). \tag{37}$$

Step 3: Update k, $\widehat{\alpha}_i$, $\widehat{e}_j$ as

$$k = k + 1 \tag{38}$$

$$\widehat{\alpha}_i(\tau|k) = \widehat{\alpha}_i(\tau|k-1) \tag{39}$$

$$\widehat{e}_j(\tau|k) = \widehat{e}_j(\tau|k-1), \quad t+\delta \leq \tau \leq t+T \tag{40}$$

$$\widehat{e}_j(\tau|k) = \exp(\lambda(t+T-\tau))\ \widehat{e}_j(\tau|k-1), \qquad t+T \leq \tau \leq t+T+\delta \tag{41}$$

and go to Step 1.

Note that, in (41), the predicted value of e_j is computed by assuming

$$\alpha_j(\tau) = 0, \quad t+T \leq \tau \leq t+T+\delta, \tag{42}$$

since the vehicle j does not have the optimal control trajectory for $\alpha_j(\tau)$ at $t+T \le \tau \le t+T+\delta$. It is also important to note that the prediction horizon T needs to be chosen such that $T \ge n\delta$ to achieve the collision avoidance.

The optimal control problem for the vehicle i at $k = i-1 \pmod n$ is described as follows:

Optimization at $t = k\delta$:

$$\min_{\widehat{\alpha}_i} \int_t^{t+T} \widehat{\alpha}_i(\tau|k)^T R \widehat{\alpha}_i(\tau|k) d\tau \tag{43}$$

subject to

$$\dot{\widehat{e}}_i = -\lambda \widehat{e}_i + \widehat{\alpha}_i, \quad \widehat{e}_i(t|k) = e_i(t) \tag{44}$$

$$\|\widehat{e}_i(\tau|k) - \widehat{e}_j(\tau|k) + \mu_i - \mu_j\|_\infty \ge \psi \tag{45}$$

$$\| -\lambda \widehat{e}_i(\tau|k) + \widehat{\alpha}_i(\tau|k)\|_\infty \le \eta \tag{46}$$

$$\|\widehat{e}_i(t+T|k)\|_\infty \le \gamma_i \tag{47}$$

$$\forall j \ne i, \quad t \le \tau \le t+T,$$

where

$$\mu_i := \begin{bmatrix} \sin\theta_r & \cos\theta_r \\ -\cos\theta_r & \sin\theta_r \end{bmatrix} \begin{bmatrix} r_i \\ l_i \end{bmatrix} \tag{48}$$

and R is a given symmetric positive definite matrix. Note that this optimization needs to be approximated by a discretized problem for implementation.

This problem determines $\widehat{e}_i(\tau|k)$ and $\widehat{\alpha}_i(\tau|k)$ as mentioned in Step 1 above, while $\widehat{e}_j(\tau|k)$ is given in advance in Step 3. The equality constraint in (44) is a prediction model for $\widehat{e}_i$ based on (31). The inequality in (45) is the collision avoidance constraint in (23).

The inequality constraint in (46) is introduced to constrain the control input u_i in (30), where η is a design parameter chosen as a positive number. The terminal constraint in (47) is introduced to guarantee the feasibility of the optimal control problem at each time and the asymptotic stability of the closed-loop system. Conditions, which γ_i needs to satisfy for the feasibility and stability, are given in the next section.

3.4 Experiments

The formation control method using MPC is applied to the experimental vehicle systems developed based on Tamiya radio-controlled model tanks (see Fig. 8). The coordinates of the vehicles (x_i, y_i, θ_i) are measured by a camera located on the ceiling. The control inputs for all vehicles are computed by one PC (CPU: Intel Pentium III 1.0GHz, RAM: 512MB) to reduce the cost and size of the experimental system. The collision avoidance control is implemented using a MATLAB algorithm[17] for solving Mixed Integer Quadratic Programs and MATLAB compiler 4[18].

The offset d in (22), the distance bound ψ, and a large number Ψ in (24) are given as $d = 0.15$, $\psi = 0.45$ and $\Psi = 30$, respectively. The prediction horizon T and update time δ in the control algorithm are given as $T = 3$ and $\delta = 0.25$. We set $\lambda = 0.4$ in (30), $\gamma_i = 1.25$ in (47) and $\eta = 0.5$ in (46).

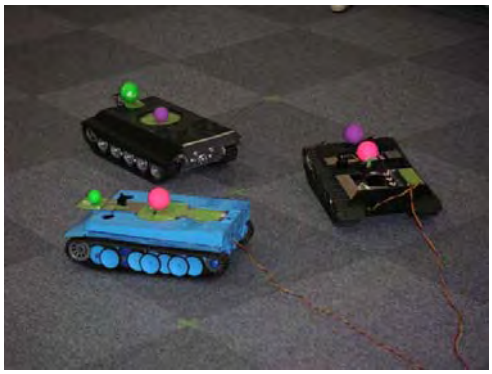

Fig. 8 Vehicles

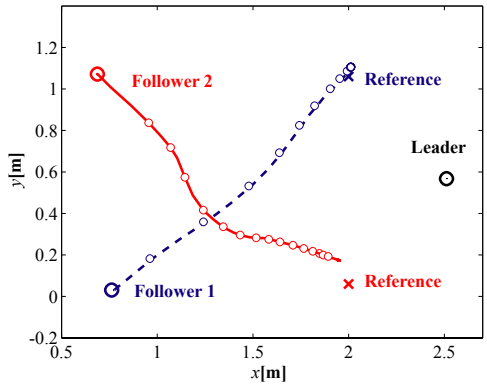

Fig. 9 x-y plot of followers

The initial coordinates of the follower 1 and 2 are $(0.5, -1.75, 3\pi/4)$ and $(-0.5, -1.75, \pi/4)$, while the reference positions are $(-0.5, -0.5)$ and $(0.5, -0.5)$ in the global frame. The leader does not move in this example due to the restriction of the space for experiments. The follower 1 and 2 collide in this situation, if the collision avoidance method in Section 3.3 is not applied (*i.e.* $\alpha = 0$).

We apply the collision avoidance method by discretizing the problem in (43) with sampling interval 1.0[sec]. Fig. 9 and Fig. 10 show the x-y plots of the followers' trajectories in the global frame and the minimum distance between the followers, *i.e.* $\min_{i,j} \|z_i - z_j\|_\infty$, respectively. From these figures, it is seen that the vehicles track the reference positions without violating collision avoidance constraints.

Note that the constraint in (46) plays a significant role in this example, since the vehicle system has an input constraint $|\omega_i| \le 1$. Fig. 11 and Fig. 12 show the x-y plots of the vehicles without applying the constraints in (46) and the time plot of a input ω_2 for the follower 2, respectively. These figures show that the follower 2 has an erratic trajectory without the constraint in (46), since the constraint $|\omega_i| \le 1$ is largely violated.

REFERENCES

[1] J.M. Maciejowski : Predictive Control with Constraints; Prentice Hall (2002)

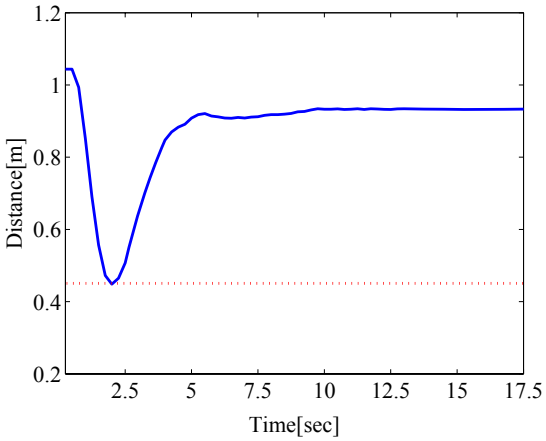

Fig. 10 Minimum distance between followers

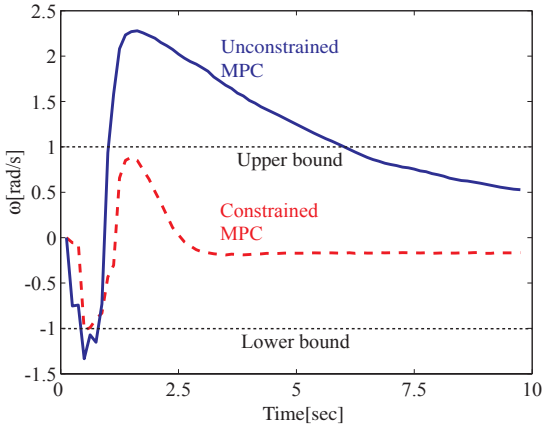

Fig. 12 ω of the follower 2

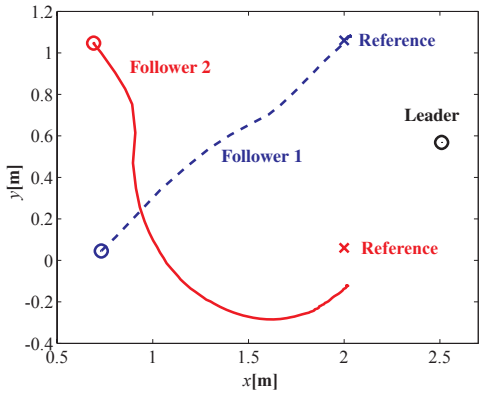

Fig. 11 x-y plot without input constraints

[2] S. Lacroix and I-K. Jung: High Resolution Terrain Mapping with an Autonomous Blimp; *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2002)

[3] H. Zhang J.P. Ostrowski: Visual Servoing with Dynamics: Control of an Unmanned Blimp; *Proc. of the IEEE Int. Conf. on Robotics and Automation* (1999)

[4] P. Grieder, P.A. Parrilo and M. Morari: Robust Receding Horizon Control - Analysis & Synthesis; *Proc. IEEE Conf. on Deci. and Contr.* (2003)

[5] H. Fukushima, R. Saito, F. Matsuno, Y. Hada, K. Kawabata and H. Asama: Model Predictive Control of an Autonomous Blimp with Input and Output Constraints; *Proc. of IEEE Int. Conf. on Control Applications* (2006)

[6] A. K. Das, R. Fierro and V. Kumar: A Vision-Based Formation Control Framework, *IEEE Trans. on Robotics and Automation*,18(5), pp.813-825 (2002)

[7] W. B. Dunbar and R. M. Murray : Distributed receding horizon control for multi-vehicle formation stabilization; *Automatica*, Vol. 42, No. 4, pp. 549-558 (2006)

[8] A. Richards and J. How : A Decentralized Algorithm for Robust Constrained Model Predictive Control; *Proc. of American Control Conference*, (2004)

[9] H. Fukushima, K. Kon and F. Matsuno : Distributed Model Predictive Control for Multi-Vehicle Formation with Collision Avoidance Constraints; *Proc. IEEE Conf. on Deci. and Contr.* (2005)

[10] Y. Hada, K. Kawabata, H. Kaetsu, H. Asama: Autonomous Blimp System for Aerial Infrastructure; *Proc. of the 2nd Int. Conf. on Ubiquitious Robuts and Intelligence Ambient Intelligence*, (2005)

[11] A. Bemporad, M. Morari, V. Dua and E. N. Piskopoulos : The explicit linear quadratic regulator for constrained systems; *Automatica*, Vol. 38, pp. 3–20 (2002)

[12] P. Grieder and M. Morari: Complexity Reduction of Receding Horizon Control; *Proc. of the 42rd IEEE Conference on Decision and Control* (2003)

[13] H. Fukushima and R. Bitmead : Robust Constrained Predictive Control using Comparison Model; *Automatica*, Vol. 41, No. 1, pp. 97–106 (2005)

[14] Y. I. Lee and B. Kouvaritakis : Robust receding horizon predictive control for systems with uncertain dynamics and input saturation; *Automatica*, Vol. 36, pp. 1497–1504 (2000)

[15] M. V. Kothare, V. Balakrishnan and M. Morari : Robust constrained model predictive control using linear matrix inequalities; *Automatica*, Vol. 32, No. 10, pp. 1361–1379 (1996)

[16] M. Kvasnica, P. Grieder, M. Baotić and M. Morari : Multi Parametric Toolbox (MPT), http://control.ee.ethz.ch/~mpt (2004)

[17] MIQP.m;http://www.dii.unisi.it/hybrid

[18] MATLAB Compiler User's Guide;The MathWorks, Inc.