International Journal of Control, Automation, and Systems (2011) 9(5):952-957
DOI 10.1007/s12555-011-0516-5
http://www.springer.com/12555

A Constrained Consensus Problem Using MPC

Jinyoung Lee, Jung-Su Kim*, Hwachang Song, and Hyungbo Shim

Abstract: This paper presents an MPC (Model Predictive Control) based consensus algorithm which solves a consensus problem in which constraints are imposed on the increment of the state of each agent. After making an artificial consensus trajectory using a previously designed consensus algorithm, the MPC is used to make the agent track the consensus trajectory. Simulation results demonstrate the effectiveness of the proposed algorithm.

Keywords: Consensus problem, MPC tracking, multi-agent systems.

1. INTRODUCTION

Recently, the consensus problem among multi-agent systems (MAS) has attracted much attention in related literature, such as in [2,7-10,12] to name but a few. This is mainly because the consensus problem unifies many other problems in engineering, such as formation control, rendezvous problems, sensor networks and so on.

For the purpose of reaching consensus, the agents negotiate with each other and change their opinion. The consensus algorithm plays a pivotal role in the negotiation. This paper is concerned with the case where the agent cannot change its opinion more than a given limit in the middle of reaching consensus. For example, if the MAS consist of mobile robots, the mobile robot cannot significantly change its heading direction at one sampling instant because of its hardware limitations. Such limitations can be modeled as constraints on the increment of the state of the agent. Therefore, the heart of the consensus problem that this paper tries to tackle is how to handle the constraints in designing the consensus algorithm while maintaining its decentralized scheme.

To take such constraints into account, this paper proposes an MPC (Model Predictive Control) based consensus algorithm. To this end, an artificial consensus trajectory is first generated using any previously reported consensus algorithm. This artificial consensus trajectory does not have to satisfy the constraints. Then, each agent employs a tracking MPC in order to make its state follow the consensus trajectory fulfilling the constraints

Simulation results show that the proposed MPC based consensus algorithm indeed results in consensus of MAS fulfilling the constraints.

Manuscript received July 5, 2010; revised February 25, 2011; accepted May 11, 2011. Recommended by Editorial Board member Young Ik Son under the direction of Editor Young Il Lee. This research was supported by a Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (2010-0006131).

Jinyoung Lee and Hyungbo Shim are with ASRI, School of Electrical Engineering and Computer Science, Seoul National University, Kwanak, PO Box 34, Seoul 151-600, Korea (e-mails: jylee@cdsl.kr, hshim@snu.ac.kr).

Jung-Su Kim and Hwachang Song are with Dept. of Control and Instrumentation Eng. and Dept. of Electrical Eng., respectively, Seoul National University of Science and Technology, 172 Gongneung 2-dong, Nowon-gu, Seoul 139-743, Korea (e-mails: jungsu@seoultech.ac.kr, hcsong@seoultech.ac.kr).

* Corresponding author.

2. PROBLEM DEFINITION AND PRELIMINARIES

In this section, we introduce the definition of the problem under consideration and some preliminaries.

2.1. Problem definition

Consider the multi-agent system (MAS) consisting of N identical agents

$$\begin{aligned} x_1(k+1) &= Ax_1(k)+Bu_1(k),\;\; x_1(k_0)=x_1(0),\\ x_2(k+1) &= Ax_2(k)+Bu_2(k),\;\; x_2(k_0)=x_2(0),\\ &\vdots\\ x_N(k+1) &= Ax_N(k)+Bu_N(k),\;\; x_N(k_0)=x_N(0), \end{aligned} \tag{1}$$

where $x_i(k)\in\mathbb{R}^n$ is the state of the i th agent and $u_i(k)\in\mathbb{R}$ the input. In order to derive the main result, several assumptions are given.

Assumption 1: The pair (A, B) is stabilizable.

Assumption 2: The interconnection between the agents contains a spanning tree.

This paper aims at designing the consensus algorithm u_i, $i=1,\ldots,N$, solving *the constrained consensus problem* defined in the following.

Definition 1: The constrained consensus problem is said to be solved if the consensus algorithm is designed such that the MAS satisfy the two conditions

$C1$. $x_i(k)-x^*(k)\to 0$ for all $i\,(i=1,\ldots,N)$ and for some $x^*(k)$ as $k\to\infty$,

$C2$. $-\overline{x}\preceq x_i(k+1)-x_i(k)\preceq\overline{x}$, for all $i\,(i=1,\ldots,N)$ and $k>0$, and for some $\overline{x}\in R_+^n$,

where $x^*(k)$ is called the consensus trajectory, $\overline{x}$ is a constant vector denoting the constraint on the state increment of the agent, and the symbol $\preceq$ implies that the inequality holds componentwise. Note that each agent cannot know $x^*(k)$ a priori.

The condition $C1$ is used in the usual consensus problem [2,7,8,10,12]. On top of $C1$ defining consensus, the condition $C2$ reflects a more practical aspect. The condition $C2$ describes that the increment of the state is limited by some quantity and is the main theme of this paper. Therefore, the proposed consensus algorithm is designed in a way that the MAS reach consensus ($C1$) fulfilling the constraints ($C2$). This problem setup is quite parallel to that of MPC (Model Predictive Control) in that it stabilizes a system fulfilling constraints on the input or state. In Section 3, a consensus algorithm solving the constrained consensus problem is devised based on an MPC. Because of the condition $C2$, the following assumption is considered.

Assumption 3: The spectrum of matrix A belong to $O_{-} \cup 1$ where O_{-} denotes the open unit disk on the complex plane. In the case of the eigenvalue 1, it should be simple.

Assumptions 1 and 2 are basic ingredients to solve the usual consensus problem (satisfying only $C1$). On the other hand, Assumption 3 restricts the class of the system severely but is inevitably required in order to satisfy $C2$. This is because it is well known that the agreed trajectory (or consensus trajectory) in the usual consensus problem (without $C2$) is a solution of the unforced part of the agent model, $x_{agree}(k+1)=Ax_{agree}(k)$ with some $x_{agree}(0)$. In other words, if the consensus algorithm is designed properly, the solution of each agent $x_i(k+1)=Ax_i(k)+Bu_i(k)$, $i=1,2,\cdots,N$ converges to $x_{agree}(k)$ as $k\to\infty$. Note that it is impossible for each agent to know $x_{agree}(0)$ in advance. Therefore, if the agreed trajectory $x_{agree}(k)$ does not satisfy the constraint $\underline{x} \preceq x_{agree}(k+1)-x_{agree}(k) \preceq \overline{x}$ in the steady state, the problem is not solvable. Therefore, $x_{agree}(k)$ needs to satisfy the constraint $C2$ in order for the constrained consensus problem to make sense and this is why Assumption 3 is necessary.

2.2. Tracking MPC

MPC (Model Predictive Control) is a kind of finite horizon optimal control [5]. It solves a finite horizon optimal control and applies the first part of the resulting optimal control sequence to the plant under control at every sampling instant.

In this section, the general structure of the tracking MPC is briefly introduced [1,4]. Let a discrete-time linear system be described by

$$x(k+1)=Ax(k)+Bu(k),$$
$$y(k)=Cx(k)+Du(k),$$

where $x(k)\in\mathbb{R}^n$ is the current state of the system, $u(k)\in\mathbb{R}^m$ is the current input. This system is subjected to the mixed constraints on the state and input

$$\begin{aligned}&(x(k),u(k))\in Z:=\{x(k)\in\mathbb{R}^n,\\ &u(k)\in\mathbb{R}^m \,|\, M_1x(k)+M_2u(k)\preceq M_3\}\ \forall k\ge 0,\end{aligned} \tag{2}$$

where M_1, M_2, and M_3 are appropriately defined matrices describing constraints on the input and state, and set Z is assumed to be a non-empty compact convex polyhedron containing the origin in its interior. For instance, if there are only input constraints such as $-\overline{u}\preceq u(k)\preceq\overline{u}$ with a finite constant $\overline{u}\succeq 0$, this is equivalent to $[I_m\ {-I_m}]^T u(k) \preceq [1_{\mathbf{m}}{}^T 1_{\mathbf{m}}{}^T]^T\overline{u}$ where I_m is the $m\times m$ identity matrix and $1_{\mathbf{m}}$ a vector with all its elements being 1. Therefore, $M_1=0$, $M_2=[I_m\ {-I_m}]^T$, and $M_3=[1_{\mathbf{m}}{}^T 1_{\mathbf{m}}{}^T]^T\overline{u}$ in this case. The constrained tracking MPC problem is to design an MPC to drive the output $y(k)$ to the piecewise constant reference r as $k\to\infty$ satisfying the constraints on the state and/or input. For the purpose of addressing the given tracking problem, similarly to other tracking problems, the steady state of the state and input are calculated by solving

$$\begin{bmatrix}0\\ r\end{bmatrix}=\begin{bmatrix}A-I & B\\ C & D\end{bmatrix}\begin{bmatrix}x_s\\ u_s\end{bmatrix},$$

where x_s and u_s denote the steady state value of the state and input with the piecewise constant r. With this steady state in mind, the optimization problem associated with the tracking MPC is solved at every sampling instant as follows

$$\begin{aligned}V_M^*(x,r)&=\min_u V_M(x,r,\mathbf{u})\\ \text{s.t. }\ & x(0)=x\\ & x(j+1)=Ax(j)+Bu(j)\\ & (x(j),u(j))\in Z,\ j=0,1,\cdots,M-1\\ & x(M)\in X_f,\end{aligned}$$

where M is the prediction horizon, $u:=\{u(0),\cdots,u(M-1)\}$ the control sequence over the horizon, r a piecewise constant reference, X_f the terminal constraint set for closed-loop stability, $V_M(x,r,\mathbf{u})$ the cost function defined by

$$\begin{aligned}V_M(x,r,\mathbf{u})&=\sum_{i=0}^{M-1}\left[\|x(i)-x_s\|_Q^2+\|u(i)-u_s\|_R^2\right]\\ &\quad+\|x(M)-x_s\|_P^2\end{aligned} \tag{3}$$

with Q being positive semidefinite, and R and P positive definite. Note that $\|x\|_Q$ denotes a weighted Euclidian norm, i.e., $\|x\|_Q=\sqrt{x^TQx}$ where x is a vector and Q is a weighting matrix. Fig. 1 describes the structure of the general tracking MPC.

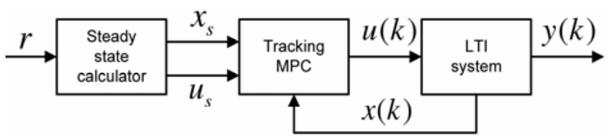


Fig. 1. General structure of a tracking MPC.

Remark 1: Depending on the given reference, it can happen that $(x_s, u_s) \notin Z$. In this case, the cost function needs to be modified in a way that the tracking MPC steers the state to the value as close as possible to x_s fulfilling the constraints. For details, see [1,4]. In theory, any stabilizing tracking MPC can be employed in the proposed MPC-based consensus algorithm. The tracking MPC in [4] is used in this paper.

3. CONSTRAINED CONSENSUS PROBLEM USING MPC

In this section, an MPC based consensus algorithm solving the constrained consensus problem is proposed. To this end, at first it is shown that the constrained consensus problem can be reformulated into a constrained tracking problem which can be dealt with using the tracking MPC.

The following lemma shows that in fact the condition $C2$ denotes mixed constraints.

Lemma 1: The constraint in $C2$ is equivalent to the mixed constraint

$$-\bar{x} \preceq Mx_i(k) + Nu_i(k) \preceq \bar{x}, \tag{4}$$

where $M=A-I$ and $N=B$.

Proof: It is shown by rewriting the constraint $C2$ as

$$\begin{aligned} x_i(k+1) - x_i(k) &= Ax_i(k) + Bu_i(k) - x_i(k) \\ &= (A-I)x_i(k) + Bu_i(k). \end{aligned}$$

With the mixed constraints in mind, if it is possible to know the consensus trajectory $x^*(k)$ a priori, we can design the consensus algorithm such that it makes x_i track the known consensus trajectory fulfilling the mixed constraints (4). In other words, the constrained consensus problem can be reformulated into a constrained tracking problem if the consensus trajectory is known. In what follows, it is presented how to identify the consensus trajectory in advance. For the purpose of generating the consensus trajectory, information MAS (iMAS) is defined

$$\begin{aligned} &z_1(k+1) = Az_1(k) + Bv_1(k),\ z_1(0) := x_1(0), \\ &z_2(k+1) = Az_2(k) + Bv_2(k),\ z_2(0) := x_2(0), \\ &\quad\vdots \\ &z_N(k+1) = Az_N(k) + Bv_N(k),\ z_N(0) := x_N(0). \end{aligned} \tag{5}$$

As seen in the equations, the dynamics of iMAS is exactly the same as that of the MAS and their initial conditions are as well. However, note that the input v_i to the iMAS is different from u_i (the input of MAS) and that iMAS is a part of the proposed consensus algorithm just as a state observer in output feedback controller. In other words, z_i dynamics is embedded in the proposed consensus algorithm u_i.

It is assumed that the input v_i to the iMAS is defined as follows

$$v_i(k) = F \sum_{j \in N_i} a_{ij}(z_j(k) - z_i(k)), \tag{6}$$

where N_i denotes the index set (which contains the set of neighbor agents) of the i th agent, $F \in \mathbb{R}^{1\times n}$ the gain, and a_{ij} the connection between the i th agent and j th agent. If the information flows from the j th agent to i th agent, then a_{ij}=1. Otherwise, a_{ij}=0. If the gain F is designed according to the methods in ([2,7,10-12]), the iMAS reaches consensus (i.e., $z_i(k)-z^*(k)\to 0$ for all i, i=1,..., N and for some $z^*(k)$, as $k\to\infty$). Previous results ([6,7,12]) report that the consensus trajectory $z^*(k)$ of the MAS (5) is the solution of $z^*(k+1) = Az^*(k)$ with $z^*(0)= \sum_{i=1}^N c_i x_i(0)$ where c_i are constants such that $\sum_{i=1}^N c_i = 1$. Note that the constants c_i are determined by the given communication graph, in particular the roots in the graph. Therefore, the consensus trajectory of iMAS is exactly the same as that of the MAS (i.e., $z^*(k)= x^*(k)$) as far as only the condition $C1$ is concerned. Note that z_i does not necessarily meet the constraints in $C2$ because the constraints are imposed only on x_i not on z_i. Thanks to this iMAS, u_i can identify its consensus trajectory in advance by exchanging z_i with its neighbors and applying v_i to iMAS.

Since the consensus trajectory z_i is known, if the tracking MPC described in Section 2.2 is applied to design u_i such that it makes x_i follow z_i fulfilling the mixed constraints (4), the constrained consensus problem is solved. Fig. 2 depicts the proposed MPC-based consensus algorithm. The proposed consensus algorithm sends and receives the state of iMAS and gives u_i using the tracking MPC. In Fig. 2, the blocks in the dashed box indicate the proposed MPC-based consensus algorithm.

One missing part not yet explained is a piecewise constant reference employed in the tracking MPC. Although z_i acts as the reference trajectory for the tracking MPC, since the tracking MPC in Section 2.2 works only for a piecewise constant reference, it is necessary to convert the z_i trajectory into a piecewise constant trajectory whose steady state is the same as that of the z_i trajectory. To this end, a function denoted by $\Xi()$ is defined as follows

$$r_i(k) = \Xi(z_i(k)) = \begin{cases} z_i(k) & \text{if } \left[\dfrac{k}{\sigma}\right] = 0 \\ r_i(k-1) & \text{otherwise,} \end{cases} \tag{7}$$

where $\left[\dfrac{a}{b}\right]$ is the remainder of $\dfrac{a}{b}$ and σ in the definition is a positive integer large than 2 and can be heuristically determined. From the definition, it is easily seen that the input to the function $\Xi()$ is z_i trajectory, the output is a piecewise constant signal r_i, and the function $\Xi()$ just samples and holds the input $z_i(k)$ in order to convert it to the corresponding piecewise constant signal. Note that $z_i(k)$ converges to a constant due to Assumption 3 and that it follows that

$$\lim_{k\to\infty} r_i(k) = \lim_{k\to\infty} z_i(k).$$

The following algorithm describes the proposed MPC based consensus algorithm.

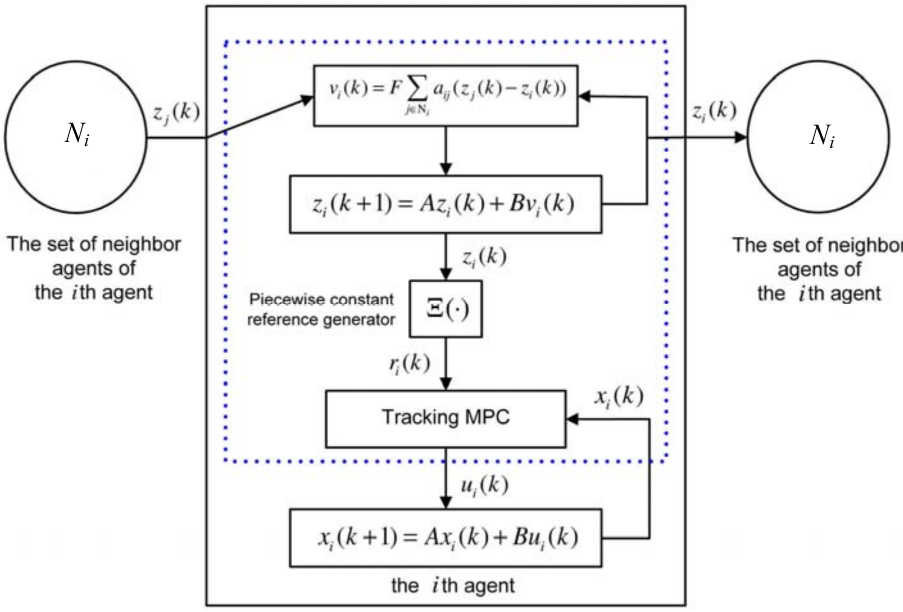


Fig. 2. The proposed MPC-based consensus algorithm.

Algorithm 1: MPC based consensus algorithm u_i of the i th agent $(\forall i \in \{1,2,\ldots,N\})$

Step 1: Initialize. $z_i(0) = x_i(0)$.

Step 2: Send z_i to its neighbors and receive $z_j, j \in N_i$ from its neighbors.

Step 3: Update z_i using $z_i(k+1) = Az_i(k) + Bv_i(k)$ and $v_i(k)$ in (6).

Step 4: Generate the piecewise constant signal $r_i(k)$ using $r_i(k) = \Xi(z_i(k))$.

Step 5: Compute u_i using the tracking MPC and apply it.

Step 6: $k := k+1$, and go to Step 2.

Now we are ready to present the main result of the paper.

Theorem 1: Suppose that Assumptions 1-3 hold true and that the optimization problems associated with the tracking MPC employed in the proposed consensus algorithm u_i are all initially feasible. In addition, the gain F in $v_i(k)$ of the iMAS is designed such that iMAS satisfies only $C1$. Then, the MPC-based consensus algorithm described in Algorithm 1 solves the constrained consensus problem.

Proof: The first important point is that iMAS reaches consensus using v_i because of Assumptions 1 and 2, and the main result in [2,7,8,10,12]. Since the function $\Xi()$ converts the z_i trajectory into a piecewise constant trajectory r_i, the tracking MPC proposed in [4] can be employed. If the optimization problem associated with the tracking MPC is initially feasible, the optimization problem is always feasible thereafter [4]. The employed tracking MPC in Section 2.2 renders the state x_i of the i th agent to track r_i (which means $C1$ is met) fulfilling the constraint $C2$. Note that Assumption 3 leads to $\lim_{k\to\infty} z_i(k) = \lim_{k\to\infty} r_i(k)$. Hence the constrained consensus problem is solved by the proposed MPC based consensus algorithm.

What the proposed MPC based consensus algorithm pursues is that it first tries to make a consensus trajectory using iMAS (which does not necessarily satisfy the constraints in $C2$) and then drive the state to the consensus trajectory fulfilling the constraints by employing a tracking MPC.

4. SIMULATION STUDY

In order to illustrate the performance of the proposed consensus algorithm, a simulation study is done in this section. For this, we consider the MAS consisting of four agents (N=4) with

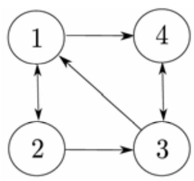
$$A = \begin{bmatrix} 1 & 1 \\ 0 & 0.9 \end{bmatrix} \text{ and } B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

and state increment constraints

$$-\begin{bmatrix} 15 \\ 15 \end{bmatrix} \le x_i(k+1) - x_i(k) \le \begin{bmatrix} 15 \\ 15 \end{bmatrix}, \quad (8)$$

and the communication graph and the corresponding Laplacian L are given in Fig. 3. To implement the proposed algorithm, a consensus algorithm proposed in [3] and the tracking MPC in [4] are employed. The coupling gain for the associated iMAS is

$$F = [0.1206 \quad 0.9278]$$



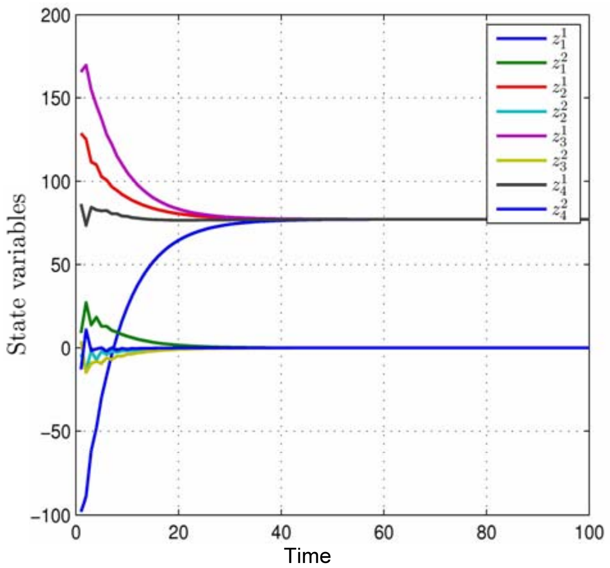
$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$$

Fig. 3. The communication graph and the Laplacian matrix.

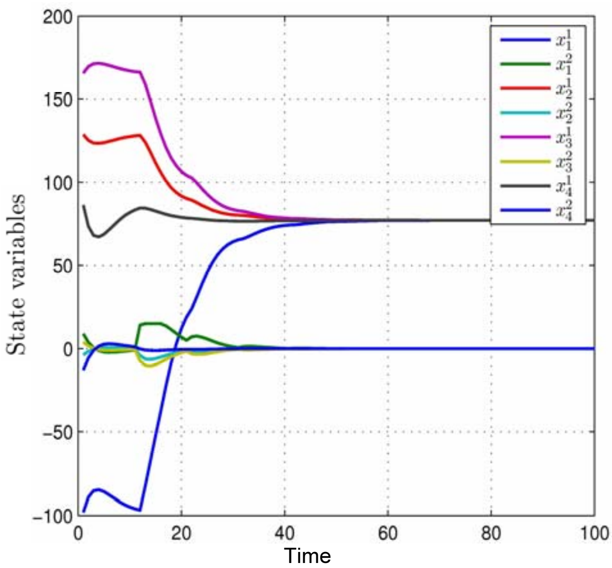
and the parameters used in the cost function (3) are

$$M = 7, \quad P = I_2, \quad Q = I_2, \quad \text{and,} \quad R = 1.$$

The proposed consensus algorithm is implemented according to Algorithm 1 in the previous section. The simulation result is depicted in Fig. 4 in which it is shown that the consensus of both iMAS (z_i) and MAS (x_i) are achieved. In Fig. 5, the red horizontal lines denote the constraints on the state increment. As shown in Fig. 5(a), the constraint on the state increments is violated in the case of the iMAS. This is natural because the consensus algorithm v_i does not take the constraints into account. On the contrary, consensus of the MAS is reached fulfilling the constraints in Fig. 5(b) owing to the proposed MPC based consensus algorithm. Fig. 6 shows the resulting z_i trajectory and corresponding r_i and x_i trajectories.

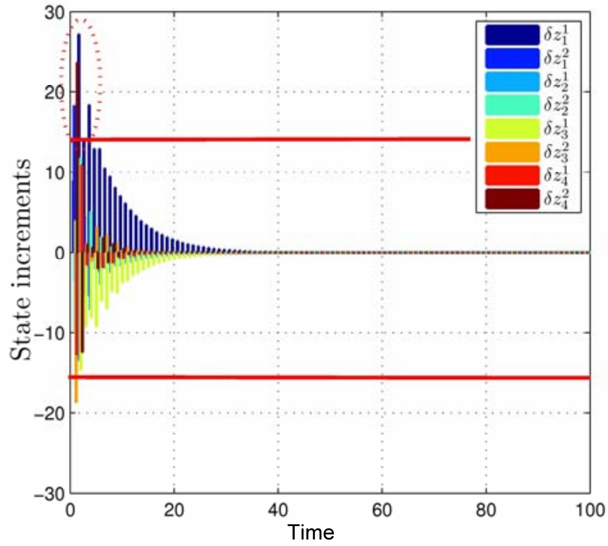


(a) Consensus of iMAS.

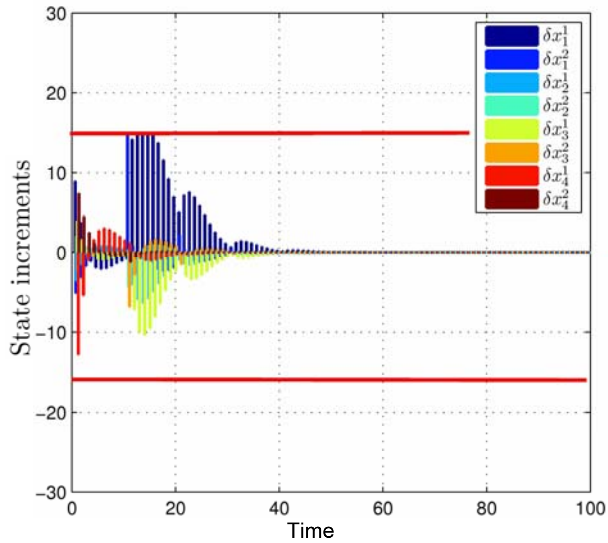


(b) Consensus of MAS.

Fig. 4. State variables of iMAS and MAS.



(a) State increments of iMAS.



(b) State increments of MAS.

Fig. 5. State increments of iMAS and MAS with the constraints (red line).

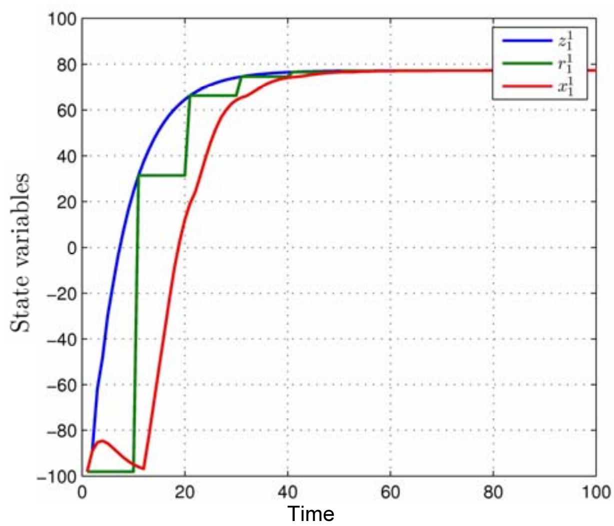


Fig. 6. Trajectories of z_i, r_i, and x_i.

5. CONCLUSION

This paper proposed an MPC-based consensus algorithm for a constrained consensus problem which is motivated from practical problems. To this end, a conventional consensus algorithm is used for a copy of MAS and the resulting trajectories are employed as a reference in tracking MPC which makes the state of the agent reach consensus while fulfilling the constraints.

REFERENCES

[1] L. Chisci and G. Zappa, "Dual mode predictive tracking of piecewise constant references for constrained linear systems," *International Journal of Control*, vol. 76, pp. 61-72, 2003.

[2] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1465-1476, September 2004.

[3] J. Lee, J.-S. Kim, and H. Shim, "Disc margins of the discrete-time LQR and its application to consensus problem," *International Journal of Systems Science*, in press, 2011.

[4] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, "MPC for tracking piecewise constant references for constrained linear systems," *Automatica*, vol. 44, pp. 2382-2387, 2008.

[5] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. N. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789-814, 2000.

[6] R. Olfati-Saber and R. M. Murray, "Consensus problem in networks of agents with switching topology and time-delays," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1520-1533, September 2004.

[7] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71-82, 2007.

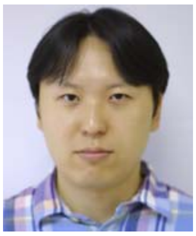
[8] J. H. Seo, H. Shim, and J. Back, "Consensus of high-order linear systems using dynamic output feedback compensator: low gain approach," *Automatica*, vol. 11, pp. 2659-2664, 2009.

[9] J. H. Seo, H. Shim, and J. Back, "Reduced-order consensus controllers for output-coupled SISO linear systems," *International Journal of Control, Automation, and Systems*, vol. 8, no. 6, pp. 1356-1363, 2010.

[10] S. E. Tuna, "LQR-based coupling gain for synchronization of linear systems," Arxiv:0801.3390 [math.OC], February 2008.

[11] J. Wang, D. Cheng, and X. Hu, "Consensus of multi-agent linear dynamic systems," *Asian Journal of Control*, vol. 10, pp. 144-155, 2008.

[12] P. Wieland, J.-S. Kim, and F. Allgöwer, "On topology and dynamics of consensus among linear high-order agents," *International Journal of Systems Science*, vol. 42, no. 10, pp. 1831-1842, 2011.



Jinyoung Lee received his B.S. degree in Electrical Engineering from Korea University in 2008, and his M.S. degree from Seoul National University. Currently, he works in Samsung's Semiconductor Research Center. His research interests include multi-agent systems, optimization and robotics.

Jung-Su Kim received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from Korea University. He was a post-doc at CDSL, Seoul National University, in 2005, at IST, University of Stuttgart, Germany from 2006-2007, and Systems Biology Laboratory, University of Leicester, UK, in 2008. Since 2009, he has been with the Department of Control and Instrumentation Engineering, Seoul National University of Science and Technology, as an assistant professor. His research interests include adaptive and predictive control, nonlinear control and systems biology.

Hwachang Song received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from Korea University in 1997, 1999 and 2003, respectively. He was a postdoctoral visiting scholar at Iowa State University from 2003 to 2004. He was an assistant professor in the School of Electronic and Information Eng. at Kunsan National University from 2005 to 2008. Currently he is an associate professor in the Dept. of Electrical Eng. at Seoul Nat'l Univ. of Science and Tech. His main research interests are on power system stability, application of optimization and energy storage and renewable generation.

Hyungbo Shim received his B.S., M.S., and Ph.D. degrees from Seoul National University, Korea, in 1993, 1995 and 2000, respectively. From 2000 to 2002 he had a position as a post-doctoral fellow for the Center for Control Engineering and Computation at the University of California, Santa Barbara. In 2002, he joined the faculty of the Division of Electrical and Computer Engineering, Hanyang University, Seoul, Korea, where he was an assistant professor until 2003. Since then, he has been with the School of Electrical Engineering at Seoul National University, Korea, where he is currently an associate professor. His research interests include analysis and control of nonlinear systems.