Research Article

Self-Triggered Model Predictive Control for Linear Systems Based on Transmission of Control Input Sequences

Koichi Kobayashi

Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan

Correspondence should be addressed to Koichi Kobayashi; k-kobaya@ssi.ist.hokudai.ac.jp

A networked control system (NCS) is a control system where components such as plants and controllers are connected through communication networks. Self-triggered control is well known as one of the control methods in NCSs and is a control method that for sampled-data control systems both the control input and the aperiodic sampling interval (i.e., the transmission interval) are computed simultaneously. In this paper, a self-triggered model predictive control (MPC) method for discrete-time linear systems with disturbances is proposed. In the conventional MPC method, the first one of the control input sequence obtained by solving the finite-time optimal control problem is sent and applied to the plant. In the proposed method, the first some elements of the control input sequence obtained are sent to the plant, and each element is sequentially applied to the plant. The number of elements is decided according to the effect of disturbances. In other words, transmission intervals can be controlled. Finally, the effectiveness of the proposed method is shown by numerical simulations.

1. Introduction

In the last decade, theory of networked control systems (NCSs) has attracted considerable attention [1, 2]. An NCS, where components such as plants, sensors, controllers, and actuators are connected through communication networks (see also Figure 1), is closely related to IoT (Internet of Things) and CPS (Cyber-Physical Systems) technologies. To develop future smart community technologies, theory of NCSs is one of the fundamental tools. In design of NCSs, packet losses, transmission delays, and communication constraints are pointed out as technical issues. Since a technical issue is different depending on properties of communication networks used, it is desirable to consider a control method for each technical issue. From this viewpoint, several methods for NCSs have been studied so far (see, e.g., [3–6]).

In this paper, the periodic paradigm is focused as one of the technical issues in NCSs. In the periodic paradigm, the message about the control input is sent from the controller to the plant at a given unit of time. Similarly, the message about the measured output is sent from the plant to the controller at a given unit of time. However, continuous transmission is not necessarily required for improving the control performance. Communication should occur, only when there exists important information. Thus, a new aperiodic approach for design of NCSs is important.

As one of the aperiodic approaches, self-triggered control has been proposed so far (see, e.g., [7–12]). In self-triggered control, the next sampling time at which the control input is recomputed is computed. That is, both the sampling interval (i.e., the transmission interval) and the control input are computed simultaneously. In this paper, we focus on self-triggered control from the viewpoint of the optimal control problem, which is one of the fundamental problems in systems and control theory. From this viewpoint, for example, a design method based one-step finite horizon boundary has been recently proposed in [13, 14]. In this method, for the sampled-data optimal control problem for linear systems, the first sampling interval is optimized under the constraint that other sampling periods are given as a constant. However, this method has two weak points: (i) a nonlinear equation must be solved online; (ii) input constraints cannot be imposed. In [10, 15], the self-triggered model predictive control (MPC) method using Taylor series expansions has been proposed.
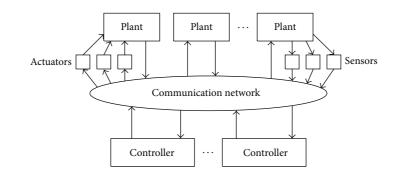
FIGURE 1: Illustration of networked control systems. Actuators and sensors may be located in a distributed way.

The MPC method is a control method that the control input is generated by solving the finite-time optimal control problem at each time (see, e.g., [16–18]).

In many existing methods, we suppose that the message about one control input, which is immediately applied to the plant, is sent from the controller to the plant. However, the message about a time sequence of the control input may be sent. From this viewpoint, we propose a new method of self-triggered MPC for NCSs with disturbances. In the proposed method, a time sequence of the control input is sent from the controller to the plant. The control input is chosen from this sequence and is sequentially applied to the plant. The length of time sequences is determined by evaluating the effect of disturbances for the cost function. In the control method proposed in [19], a time sequence of the control input is sent to the plant, but its length is uniform. In also [20, 21], self-triggered MPC has been studied for linear systems with state and input constraints. In this paper, we propose an easy and convenient method for linear systems with only input constraints. Then, the length of time sequences of the control input can be easily obtained by evaluating degradation of the optimality via disturbances. Furthermore, time delay by computation of the optimal control problem and communication of messages about the control input is discussed. Finally, numerical simulations are presented to show the effectiveness of the proposed method. The proposed method provides us with one of the fundamental results in theory of NCSs.

The main contribution of this paper is to propose an easy and convenient method of self-triggered MPC for linear systems with disturbances and input constraints.

This paper is organized as follows. In Section 2, some preliminaries are explained. In Section 3, the self-triggered MPC method is proposed. In Section 4, the improved version of the proposed method is presented. In Section 5, numerical simulations are presented. In Section 6, we conclude this paper.

Notation. Let $\mathscr{R}$ denote the set of real numbers. Let I_n and $0_{m \times n}$ denote the $n \times n$ identity matrix and the $m \times n$ zero matrix, respectively. For simplicity, we sometimes use the symbol 0 instead of $0_{m \times n}$, and the symbol I instead of I_n. For the matrix M, let $M^\top$ denote the transpose of M.

2. Preliminaries

As a preparation, we will explain the finite-time optimal control problem and the conventional model predictive control (MPC) method for constrained linear system.

2.1. Discrete-Time Constrained Linear Systems. As a plant, consider the following discrete-time constrained linear system:

$$x(k+1) = Ax(k) + B_1 u(k) + B_2 d(k), \tag{1}$$

where $x(k) \in \mathscr{R}^n$ is the state, $u(k) \in [u_{\min}, u_{\max}] \subset \mathscr{R}^m$ is the control input, and $d(k) \in [d_{\min}, d_{\max}] \subset \mathscr{R}^r$ is the disturbance. In this paper, the constraint is imposed for the control input. We assume that the upper and lower bounds of the disturbance are known.

When the finite-time optimal control problem is solved, we use the following model:

$$\hat{x}(k+1) = A\hat{x}(k) + B_1 u(k), \tag{2}$$

where $\hat{x}(k) \in \mathscr{R}^n$ is the predicted state under $\hat{x}(0) = x(0)$.

2.2. Finite-Time Optimal Control Problem. Consider the following finite-time optimal control problem (see, e.g., [16–18]).

Problem 1. Suppose that for prediction model (2) the initial state $\hat{x}(0) = x_0$ and the prediction horizon N are given. Then, find a control input sequence $\bar{u} := \begin{bmatrix} u^\top(0) & u^\top(1) & \cdots & u^\top(N-1) \end{bmatrix}^\top$ minimizing the following cost function:

$$J(x_0, \bar{u}) = \sum_{k=0}^{N} \hat{x}^\top(k) Q \hat{x}(k) + \sum_{k=0}^{N-1} u^\top(k) R u(k) \tag{3}$$

subject to the input constraint $u(k) \in [u_{\min}, u_{\max}] \subseteq \mathscr{R}^m$, where $Q = Q^\top$ is positive semidefinite and $R = R^\top$ is positive definite.

Consider rewriting this problem into a quadratic programming (QP) problem. From (2), we can obtain

$$\bar{x} = \bar{A} x_0 + \bar{B}_1 \bar{u}, \tag{4}$$

where

$$\bar{x} = \begin{bmatrix} x^\top(0) & x^\top(1) & \cdots & x^\top(N) \end{bmatrix}^\top,$$

$$\bar{A} = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix},$$

(5)

$$\bar{B}_1 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B_1 & 0 & \cdots & 0 \\ AB_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ A^{N-1}B_1 & \cdots & AB_1 & B_1 \end{bmatrix}.$$

Then, cost function (3) can be rewritten as

$$J = \bar{u}^T \left(\bar{R} + \bar{B}_1^T \bar{Q} \bar{B}_1 \right) \bar{u} + 2x_0^T \bar{A}^T \bar{Q} \bar{B}_1 \bar{u} + x_0^T \bar{A}^T \bar{Q} \bar{A} x_0, \quad (6)$$

where

$$\bar{Q} = \begin{bmatrix} Q & & 0 \\ & \ddots & \\ 0 & & Q \end{bmatrix},$$

$$\bar{R} = \begin{bmatrix} R & & 0 \\ & \ddots & \\ 0 & & R \end{bmatrix}.$$

(7)

The input constraint $u(k) \in [u_{\min}, u_{\max}]$ can be rewritten as $\bar{u}_{\min} \le \bar{u} \le \bar{u}_{\max}$; that is,

$$\begin{bmatrix} -I \\ I \end{bmatrix} \bar{u} \le \begin{bmatrix} -\bar{u}_{\min} \\ \bar{u}_{\max} \end{bmatrix}, \quad (8)$$

where $\bar{u}_{\min}$ and $\bar{u}_{\max}$ are defined by

$$\bar{u}_{\min} = \begin{bmatrix} u_{\min} \\ u_{\min} \\ \vdots \\ u_{\min} \end{bmatrix} \in \mathscr{R}^{mN},$$

$$\bar{u}_{\max} = \begin{bmatrix} u_{\max} \\ u_{\max} \\ \vdots \\ u_{\max} \end{bmatrix} \in \mathscr{R}^{mN},$$

(9)

respectively. Using the above results, Problem 1 is equivalent to the following QP problem.

Problem 2. Find $\bar{u}$ minimizing the cost function (6) subject to (8).

Problem 2 can be solved by using a suitable solver such as IBM ILOG CPLEX and Gurobi Optimizer. Hereafter, let J^* and $\bar{u}^* = \begin{bmatrix} (u^*(0))^\top & (u^*(1))^\top & \cdots & (u^*(N-1))^\top \end{bmatrix}^\top$ denote the optimal value of the cost function and the optimal sequence of the control input, respectively.

2.3. Conventional MPC Method. The procedure of the conventional MPC method in networked control systems (NCSs) is presented as follows (see, e.g., [16–18]).

Conventional Procedure of MPC in NCSs

Step 1. Set $k = 0$, and give the initial state $x(0) = x_0$.

Step 2. In the controller, solve Problem 1.

Step 3. Send the message about $u^*(0)$ from the controller to the plant.

Step 4. Apply $u^*(0)$ to the plant.

Step 5. Measure $x(1)$, and send the message about $x(1)$ from the plant to the controller.

Step 6. Update $k = 1$ to $k = 0$, and return to Step 2.

In the above procedure, there are some technical issues as follows.

(i) The message about only one control input $u^*(0)$ must be sent to the plant at each time.

(ii) Time delay by computation and communication occurred.

In order to solve (i), we consider that the message about multiple control inputs $u^*(0), u^*(1), \ldots, u^*(N^* - 1)$ is sent to the plant. Then, it is important to determine suitable $N^* (\le N)$. In Section 3, we will propose a new procedure of MPC in NCSs.

In order to solve (ii), we will propose an improved version of the procedure proposed in Section 3. See Section 4 for further details.

3. Proposed Self-Triggered MPC Method

In this section, a new MPC method called a self-triggered MPC method is proposed. The self-triggered control method is a control method that the control input and the sampling interval are computed simultaneously (see, e.g., [7–15]). In the conventional case, the control input is a given constant in each sampling interval. In the proposed self-triggered MPC method, the message about multiple control inputs $u^*(0), u^*(1), \ldots, u^*(N^* - 1)$ is sent from the controller to the plant. The sent control input is applied to the plant at time $0, 1, \ldots, N^* - 1$. When N^* is calculated, we focus on the effect of the disturbance.

First, we define the notation. Define

$$
\begin{aligned}
\overline{u}^*_{N-i} \\
:= \left[\left(u^*(0)\right)^\top \ \left(u^*(1)\right)^\top \ \cdots \ \left(u^*(N-i-1)\right)^\top \right]^\top,
\end{aligned} \tag{10}
$$

where $i = 0, 1, \ldots, N-1$, and $\overline{u}^*_N = \overline{u}^*$ holds. Let J^*_{N-i}, $i = 0, 1, \ldots, N-1$, denote the value of cost function (3) in the time interval $[0, N-i]$, where $\overline{u}^*_{N-i}$ is applied. That is,

$$
\begin{aligned}
J^*_{N-i}\left(x_0, \overline{u}^*_{N-i}\right) := \sum_{k=0}^{N-i} \widehat{x}^\top(k) Q \widehat{x}(k) \\
+ \sum_{k=0}^{N-i-1} \left(u^*(k)\right)^\top R u^*(k),
\end{aligned} \tag{11}
$$

where $\widehat{x}(k+1) = A\widehat{x}(k) + B_1 u^*(k)$.

Consider the effect of the disturbance $d(k)$ for the cost function. Let $\widetilde{J}$ denote the cost function in which $\widehat{x}$ is replaced with x. When $\overline{u}^*$ is used, $\widetilde{J}$ can be obtained as

$$
\begin{aligned}
\widetilde{J}\left(x_0, \overline{u}^*, \overline{d}\right) = \sum_{k=0}^{N} x^\top(k) Q x(k) + \sum_{k=0}^{N-1} u^\top(k) R u(k) \\
= J^* + \alpha\left(x_0, u^*(0), d(0)\right) \\
+ \beta\left(x_0, \overline{u}^*, \overline{d}\right),
\end{aligned} \tag{12}
$$

where $\overline{d} := \left[d^\top(0) \ d^\top(1) \ \cdots \ d^\top(N-1) \right]^\top$,

$$
\begin{aligned}
& \alpha\left(x_0, u^*(0), d(0)\right) \\
& = \begin{bmatrix} Ax_0 + B_1 u^*(0) \\ d(0) \end{bmatrix}^\top Q' \begin{bmatrix} Ax_0 + B_1 u^*(0) \\ d(0) \end{bmatrix}, \\
& Q' = \begin{bmatrix} I & B_2 \end{bmatrix}^\top Q \begin{bmatrix} I & B_2 \end{bmatrix}, \\
& \beta\left(x_0, \overline{u}^*, \overline{d}\right) \\
& = \begin{bmatrix} \overline{A}x_0 + \overline{B}_1 \overline{u}^* \\ \overline{d} \end{bmatrix}^\top \overline{Q}' \begin{bmatrix} \overline{A}x_0 + \overline{B}_1 \overline{u}^* \\ \overline{d} \end{bmatrix} \\
& \quad - \alpha\left(x_0, u^*(0), d(0)\right), \\
& \overline{Q}' = \begin{bmatrix} I & \overline{B}_2 \end{bmatrix}^\top \overline{Q} \begin{bmatrix} I & \overline{B}_2 \end{bmatrix}, \\
& \overline{B}_2 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B_2 & 0 & \cdots & 0 \\ AB_2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ A^{N-1}B_2 & \cdots & AB_2 & B_2 \end{bmatrix}.
\end{aligned} \tag{13}
$$

If N^* is chosen as a smaller value, then $\beta(x_0, \overline{u}^*, \overline{d})$ becomes smaller. In the case of $N^* = 1$, $\beta(x_0, \overline{u}^*, \overline{d}) = 0$ holds. On the other hand, even if $N^* = 1$ is chosen, $\alpha(x_0, u^*(0), d(0))$ is not zero generally. In other words, $\alpha(x_0, u^*(0), d(0))$ does not depend on N^*.

Consider deriving the upper bound of $\widetilde{J}(x_0, \overline{u}^*, \overline{d})$. We define

$$
\begin{aligned}
\overline{\alpha}\left(x_0, u^*(0)\right) := \max_{d(0)} \alpha\left(x_0, u^*(0), d(0)\right), \\
\overline{\beta}\left(x_0, \overline{u}^*\right) := \max_{\overline{d}} \beta\left(x_0, \overline{u}^*, \overline{d}\right) - \overline{\alpha}\left(x_0, u^*(0)\right).
\end{aligned} \tag{14}
$$

If x_0 and $\overline{u}^*$ are given, then $\overline{\alpha}(x_0, u^*(0))$ and $\overline{\beta}(x_0, \overline{u}^*)$ can be calculated. Then, we can obtain the following lemma immediately.

Lemma 1. *The following relation holds:*

$$
\widetilde{J}\left(x_0, \overline{u}^*, \overline{d}\right) \le J^* + \overline{\alpha}\left(x_0, u^*(0)\right) + \overline{\beta}\left(x_0, \overline{u}^*\right). \tag{15}
$$

From this lemma, we see that the upper bound of $\widetilde{J}(x_0, \overline{u}^*, \overline{d})$ can be obtained by the optimal value of the cost function J of (3), the initial state x_0, and the optimal input sequence $\overline{u}^*$. Finally, in a similar way to the definitions of $\overline{u}^*_{N-i}$ and $J^*_{N-i}(x_0, \overline{u}^*_{N-i})$, we define $\overline{\beta}_{N-i}(x_0, \overline{u}^*)$.

Using this result, we can obtain the procedure of the self-triggered MPC in NCSs. The proposed procedure is presented as follows.

Procedure of Self-Triggered MPC in NCSs

Step 1. Set $k = 0$, and give the initial state $x(0) = x_0$.

Step 2. In the controller, solve Problem 1.

Step 3. In the controller, find a maximum $N^* \in [1, N]$ satisfying the following condition:

$$
\begin{aligned}
J^*_{N^*} + \overline{\alpha}\left(x_0, \overline{u}^*(0)\right) + \overline{\beta}\left(x_0, \overline{u}^*_{N^*}\right) \\
\le \gamma\left(J^*_{N^*} + \overline{\alpha}\left(x_0, \overline{u}^*(0)\right)\right),
\end{aligned} \tag{16}
$$

where $\gamma \ge 1$ is a given constant.

Step 4. Send the message about $\overline{u}^*_{N^*}$ from the controller to the plant.

Step 5. Apply $u^*(0), u^*(1), \ldots, u^*(N^*-1)$ to the plant at time $0, 1, \ldots, N^*-1$, respectively.

Step 6. Measure $x(N^*)$, and send the message about $x(N^*)$ from the plant to the controller.

Step 7. Set $x_0 := x(N^*)$, update $k = N^*$ to $k = 0$, and return to Step 2.

In the above procedure, control performance is evaluated by (16), where γ is a given tuning parameter. In (16), the relation between the degradation of the performance and the optimal value of the cost function is relatively evaluated. Even if $N^* = 1$, then the effect of the disturbance is given by $\overline{\alpha}(x_0, \overline{u}^*(0))$. Hence, the right hand side of (16) is given by $\gamma(J^*_{N^*} + \overline{\alpha}(x_0, \overline{u}^*(0)))$. We remark that computation of Step 3 is easy. That is, for each $N^* = N, N-1, \ldots, 1$, condition (16) is checked.

The following lemma can be obtained as a property of the above procedure.

Lemma 2. *Assume that $d_{\min} \neq 0$ and $d_{\max} \neq 0$. In the case of $\gamma = 1$, the parameter N^* is always given by $N^* = 1$.*

Proof. From (16), the equality condition $\overline{\beta}(x_0, \overline{u}^*_{N^*}) = 0$ must hold in the case of $\gamma = 1$. This condition holds if and only if $N^* = 1$. $\qquad\square$

From this lemma, we see that if degradation of the optimality by the disturbance is not allowed, then the above procedure of self-triggered MPC does not work and is reduced to the conventional procedure in Section 2.3.

4. Consideration of Delay by Computation and Communication

Using the procedure proposed in Section 3, self-triggered control is realized. However, we do not consider time delay by computation of the optimal control problem and communication from the controller (plant) to the plant (controller). In order to consider time delay, we must use the predicted state in the procedure of self-triggered MPC.

Here, we assume that computation and communication can be executed within one discrete time. Then, the improved version of the procedure proposed in Section 3 is presented as follows.

Improved Procedure of Self-Triggered MPC in NCSs

Step 1. Set $k = 0$, and give the initial state $x(0) = x_0$.

Step 2. In the controller, solve Problem 1.

Step 3. In the controller, find a maximum $N^* \in [1, N]$ satisfying condition (16).

Step 4. Send the message about $\overline{u}^*_{N^*}$ from the controller to the plant.

Step 5. Apply $u^*(0), u^*(1), \ldots, u^*(N^* - 1)$ to the plant at time $0, 1, \ldots, N^* - 1$, respectively.

Step 6. Measure $x(N^* - 1)$, and send the message about $x(N^* - 1)$ from the plant to the controller.
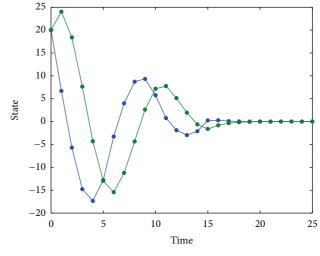
Step 7. In the controller, compute the predicted state $\widehat{x}(N^*)$ from $x(N^* - 1)$ and $u^*(N^* - 1)$.

Step 8. Set $x_0 := \widehat{x}(N^*)$, update $k = N^*$ to $k = 0$, and return to Step 2.

5. Numerical Simulations

As a plant, consider the following system:

$$x(k+1) = \begin{bmatrix} 1 & -0.6 \\ 0.6 & 0.6 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k)$$

$$+ \begin{bmatrix} 1 \\ 0 \end{bmatrix} d(k). \tag{17}$$

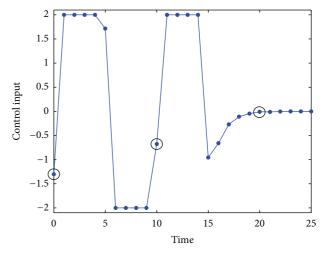FIGURE 2: State trajectory in case (i).

FIGURE 3: Control input in case (i). The circles imply the timing the controller received the message about the control input.

The input constraint is given by $u(t) \in [-2, 2]$. The upper and lower bounds of the disturbance are given by $d_{\min} = -1$ and $d_{\max} = 1$, respectively. Parameters in Problem 1 are given as follows: $N = 10$, $Q = 100I_2$, and $R = 1$.

We present the computation results. Here, we consider the following three cases:

(i) there is no disturbance and no delay (the ideal case);

(ii) there is the disturbance, and there is no delay;

(iii) there are the disturbance and delay.

In numerical simulations, the disturbance $d(k)$ is given by $d(k) = 1$.

First, case (i) is presented. Figures 2 and 3 show the state trajectory and the control input, respectively. In this case, N^* was always obtained by $N^* = 10$ (i.e., $N^* = N$). From these figures, we see that the state converges to the origin, and the number of transmissions from the controller to the plant is small.
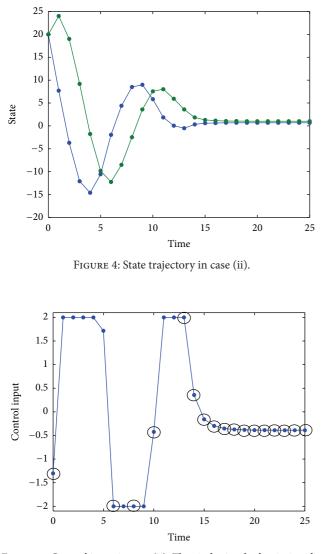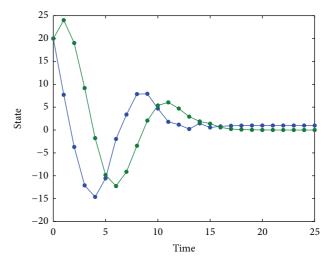
FIGURE 4: State trajectory in case (ii).

FIGURE 6: State trajectory in case (iii).

FIGURE 5: Control input in case (ii). The circles imply the timing the controller received the message about the control input.
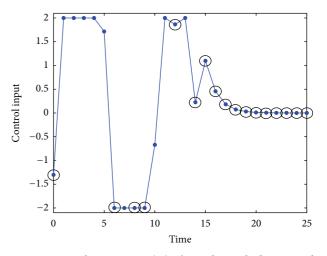

FIGURE 7: Control input in case (iii). The circles imply the timing the controller received the message about the control input.

Here, consider comparing the case of $N^* = 10$ with the case of $N^* = 1$. First, the difference in the control input is discussed. In the time interval $[0, 25]$, the maximum absolute error was 0.22. That is, in the case of $N^* = 10$, the control input at time 5 was 1.72, and in the case of $N^* = 1$ the control input at time 5 was 1.50. The mean absolute error in the time interval $[0, 25]$ was 0.02. Next, the difference in the state is discussed. In the time interval $[0, 25]$, the maximum absolute error of the first element of the state was 0.22. That is, in the case of $N^* = 10$, the first element of the state at time 6 was -3.27, and in the case of $N^* = 1$ the first element of the state at time 6 was -3.49. The mean absolute error in the time interval $[0, 25]$ was 0.03. In the time interval $[0, 25]$, the maximum absolute error of the second element of the state was 0.21. That is, in the case of $N^* = 10$, the second element of the state at time 8 was -4.34, and in the case of $N^* = 1$ the second element of the state at time 8 was -4.55. The mean absolute error in the time interval $[0, 25]$ was 0.03. Thus, the error was

very small, and under the ideal environment, solving Problem 1 at each time is not necessarily required.

Next, case (ii) is presented. Figures 4 and 5 show the state trajectory and the control input, respectively. In this case, N^* is obtained by $N^* = 6, 2, 2, 3, 1, 1, \ldots$. From Figure 4, we see that the state converges to the neighborhood of the origin (but the steady-state error appeared). From Figure 5, we see that the number of transmissions from the controller to the plant is small in the transient response. However, we see also that in the steady state the transmission occurred at each time. This is because the disturbance does not depend on the state, and the effect of the disturbance is relatively large in the steady state.

Third, case (iii) is presented. Figures 6 and 7 show the state trajectory and the control input, respectively. In this case, N^* is obtained by $N^* = 6, 2, 1, 3, 2, 1, 1, \ldots$. From Figure 6, we see that the steady-state error is larger than that in case (ii). From Figure 7, we see that in the steady state, the transmission

occurred at each time. This result is the same as that in case (ii).

Finally, we comment the computation time for solving Problem 1 and finding N^*. For each case, Problem 1 and the problem for finding N^* were solved 20 times, where we used Gurobi Optimizer 5.6.2 on the computer with the Intel Core i7-4770K processor and the 32 GB memory. The worst computation time was 0.0373 sec (case (i)), 0.0365 sec (case (ii)), and 0.0421 sec (case (iii)). The mean computation time was 0.0322 sec (case (i)), 0.0316 sec (case (ii)), and 0.0332 sec (case (iii)). Thus, these problems can be solved fast.

6. Conclusion

In this paper, we proposed the self-triggered model predictive control method of discrete-time linear systems with disturbances. In the proposed method, transmission intervals from the controller to the plant are computed depending on the effect of the disturbance. If the effect of the disturbance is relatively small, then transmission intervals become long. The effectiveness of the proposed method is presented by numerical simulations.

One of the future efforts is to consider state/input-dependent disturbances. Stability analysis of the closed-loop system is also important future work.

Competing Interests

The author declares that there are no competing interests regarding the publication of this paper.

Acknowledgments

References

[1] C. T. Abdallah and H. G. Tanner, "Complex networked control systems: introduction to the special section," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 30–32, 2007.

[2] P. Antsaklis and J. Baillieul, "Special issue on technology of networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 5–8, 2007.

[3] L.-S. Hu, T. Bai, P. Shi, and Z. Wu, "Sampled-data control of networked linear control systems," *Automatica*, vol. 43, no. 5, pp. 903–911, 2007.

[4] H. Ishii, "Stabilization under shared communication with message losses and its limitations," in *Proceedings of the 45th IEEE Conference on Decision and Control (CDC '06)*, pp. 4974–4979, December 2006.

[5] H. Ishii, "H_∞ control with limited communication and message losses," *Systems & Control Letters*, vol. 57, no. 4, pp. 322–331, 2008.

[6] K. Kobayashi and K. Hiraishi, "Optimal control of a class of networked systems based on MLD framework," in *Proceedings of the 18th IFAC World Congress*, pp. 66–71, Milano, Italy, 2011.

[7] A. Anta and P. Tabuada, "Self-triggered stabilization of homogeneous control systems," in *Proceedings of the American Control Conference (ACC '08)*, pp. 4129–4134, Seattle, Wash, USA, June 2008.

[8] A. Anta and P. Tabuada, "To sample or not to sample: self-triggered control for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2030–2042, 2010.

[9] A. Camacho, P. Martí, M. Velasco et al., "Self-triggered networked control systems: an experimental case study," in *Proceedings of the International Conference on Industrial Technology (ICIT '10)*, pp. 123–128, Vina del Mar, Chile, March 2010.

[10] K. Kobayashi and K. Hiraishi, "Self-triggered model predictive control with delay compensation for networked control systems," in *Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society (IECON '12)*, pp. 3200–3205, IEEE, Montreal, Canada, October 2012.

[11] M. Mazo Jr. and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," in *Proceedings of the 47th IEEE Conference on Decision and Control (CDC '08)*, pp. 435–440, Cancun, Mexico, December 2008.

[12] X. Wang and M. D. Lemmon, "Self-triggered feedback control systems with finite-gain $\mathscr{L}_2$ stability," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 452–467, 2009.

[13] M. Velasco, P. Martí, J. Yépez, F. J. Ruiz, J. M. Fuertes, and E. Bini, "Qualitative analysis of a one-step finite-horizon boundary for event-driven controllers," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC '11)*, pp. 1662–1667, IEEE, Orlando, Fla, USA, December 2011.

[14] J. Yépez, M. Velasco, P. Martí, E. X. Martín, and J. M. Fuertes, "One-step finite horizon boundary with varying control gain for event-driven Networked Control Systems," in *Proceedings of the 37th Annual Conference of IEEE Industrial Electronics (IECON '11)*, pp. 2606–2611, Melbourne, Australia, November 2011.

[15] K. Kobayashi and K. Hiraishi, "Self-triggered model predictive control using optimization with prediction horizon one," *Mathematical Problems in Engineering*, vol. 2013, Article ID 916040, 9 pages, 2013.

[16] E. F. Camacho and C. B. Alba, *Model Predictive Control*, Springer, Berlin, Germany, 2nd edition, 2007.

[17] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[18] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers and Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

[19] D. Lehmann, E. Henriksson, and K. H. Johansson, "Event-triggered model predictive control of discrete-time linear systems subject to disturbances," in *Proceedings of the 12th European Control Conference (ECC '13)*, pp. 1156–1161, Zurich, Switzerland, July 2013.

[20] E. Aydiner, F. D. Brunner, W. P. Heemels, and F. Allgower, "Robust self-triggered model predictive control for constrained discrete-time LTI systems based on homothetic tubes," in *Proceedings of the European Control Conference (ECC '15)*, pp. 1587–1593, IEEE, Linz, Austria, July 2015.

[21] F. D. Brunner, W. P. M. H. Heemels, and F. Allgöwer, "Robust self-triggered MPC for constrained linear systems," in *Proceedings of the 13th European Control Conference (ECC '14)*, pp. 472–477, Strasbourg, France, June 2014.