Model Predictive Control of Multivariable Plants Using Interactor and Solving Procedure of Matrix Polynomial Diophantine Equations

Akira Inoue *, Mingcong Deng **, Takao Sato ***, Akira Yanou ****, and Tomohiro Henmi ****

Abstract : This paper proposes a design method of model predictive control (MPC) for multi-input multi-output (MIMO) plants with time-delay by using an interactor matrix and a sequential procedure to solve the matrix polynomial Diophantine equations required to be solved in the design. The equations are of matrix polynomials, and matrix calculations are not commutative; hence it is not easy to solve the equations, and it is necessary to obtain a sequential solving procedure. Also, the difficulty in the design of MPC of MIMO plants comes from the fact that a plant transfer function is a matrix, which is not commutative in multiplication. This paper avoids this difficulty by deriving a plant transfer function with a scalar polynomial denominator. And to handle the time-delay in MIMO plants, an interactor matrix is used to shift the outputs by time-delay steps. Then the design problem with time-delay is reduced to a problem without time-delay. There exist designs of MPC for time-delay plants by using a longer horizon than the time-delay steps. In this paper, it is shown by simulations that MPC having a long horizon is sensitive to disturbances and that the proposed MPC is less sensitive.

Key Words : model predictive control (MPC), multi-input multi-output (MIMO), time-delay, interactor matrix, matrix polynomial Diophantine equation.

1. Introduction

Chemical plants in industry include time-delays in their input-output relations in many cases. Since the effects of control inputs appear after the delay time passed, the controls of plants with time-delay are difficult, and many papers are published on controls of time-delay plants.

Model Predictive Control (MPC) [1] is effective for time-delay plants because MPC can design control laws using a longer horizon than time-delay steps to cover time-delay. This fact is written in a textbook [2] written by engineers in industry. Also, a control engineer [3] in industry said he designs MPC using longer horizons to handle time-delay. Shah and others [4] discussed a relation between multi-input multi-output (MIMO) MPC and interactor [5], which represents time-delay in MIMO plants, and stated that output horizon is set to be longer than (control horizon) + (time-delay) − 1. But these methods result in long horizons. MPC is designed to minimize an index which is the sum of squared output errors and squares of weighted inputs during some time-interval, that is, horizon. Hence when disturbances exist in input or output, a longer horizon causes the disturbances to affect the index in the longer time-interval and to deteriorate the control results. This is confirmed by a simulation shown in this paper.

On the other hand, if the horizon is shorter than the time-delay, then the effects of control inputs do not appear in the outputs in the index, and the inputs to minimize the index are not obtained. Hence, this paper proposes to shift the control horizon to the end of time-delay instead of using a longer horizon.

If the followings are assumed, (1) the plant model and the number of time-delay steps are known, (2) the future values of the reference inputs during time-delay steps are known, then the shifting method is possible. If this method is applied to MIMO plants, time-delays of MIMO plants should be known. But MIMO plants have different time-delays from each input to output. Hence time-delay in MIMO plants was extended by Wolovich [5] to the interactor. The interactor is extensively discussed in relation to the designs of multivariable adaptive control systems [6]–[8]. Later the interactor is used in various areas of control designs, such as single LQ output feedback [9], output error learning control [10], and iterative learning control [11].

Multivariable MPC was first proposed by Koivo [12] in 1980, but it supposed that the numerator of the transfer matrix of the plant has a nonsingular leading coefficient matrix, that is, the plant has no time-delay. Also, Camacho and Bordox [1] gave a design of MPC of MIMO plants without time-delay.

The difficulty in designing multivariable MPC comes from the facts that plant is expressed by matrices and matrix multiplication is not commutative. To avoid this difficulty, Koivo [12] and Camacho and Bordox [1] introduced new many auxiliary matrices, and their design procedures are very complicated. The authors [13] already proposed design of MIMO MPC using interactor. In the design, to avoid the difficulty of being not commutative of matrix multiplication, a transfer function with a scalar polynomial denominator is derived from matrix transfer functions. This causes no need for additional matrices. And the designing procedure becomes very simple. To handle the time-delay, this paper uses an interactor matrix, and defines a plant model without time-delay by shifting the outputs and the
numerator matrix. Then the design problem with time-delay reduces to a problem without time-delay. Finally, the problem without time-delay is solved by using solutions of matrix polynomial Diophantine equations.

The matrix polynomial Diophantine equations include so many unknown elements in the unknown matrices. Further, the matrix equations include non-commutative matrix calculation. Hence it is difficult to solve the matrix polynomial Diophantine equations numerically, and it requires some solving procedures. Koivo [12] just mentioned the Euclidean algorithm to solve the equations and did not give any solving procedures at all. Later Camacho and Bordons [1] gave a recursive solving procedure, but their procedure needs to calculate intermediate matrices. This paper gives a sequential solving procedure which does not need intermediate matrices.

In the next section, the problem statement and a plant model are given. Then in Section 3, an interacto is defined. And design of MPC for MIMO plants is shown. Then a sequential solving procedure is given. Section 4 gives simulation results. Finally, Section 5 is concluding remarks.

Notation Time-delay is denoted by $z^{-1}$, as $z^{-1}y(k) = y(k-1)$. Polynomials and rational functions of $z^{-1}$ are written as $A(z^{-1})$ and $A(z^{-1})^\dagger$. Matrices are denoted by bold letters, that is, $A(z^{-1})$ and $A(z^{-1})^\dagger$ are matrices of polynomials and rational functions.

2. Problem Statement and Plant Model

The plant considered in this paper has $m$ inputs $u(k)$ and $m$ outputs $y(k)$ and is described as

$$A(z^{-1})y(k) = B(z^{-1})(u(k) + d(k)),$$

$$A(z^{-1}) = I + a_1 z^{-1} + \cdots + a_m z^{-m},$$

(1)

$$B(z^{-1}) = \begin{bmatrix} B_0 & B_1 & \cdots & B_m \end{bmatrix} z^{-m},$$

(2)

where $A(z^{-1})$ is a scalar polynomial, $B(z^{-1})$ is an $m \times m$ polynomial matrix, and $d(k)$ is an unknown disturbance. This model which has a scalar polynomial denominator $A(z^{-1})$, not a matrix polynomial, is derived by one of the following methods:

1. If the original plant model is given by a transfer matrix, then the model (1) is derived by the following way. For simplicity, 2-input 2-output case ($m = 2$) is explained. Let a given transfer matrix $T(z^{-1})$ be

$$T(z^{-1}) = \begin{bmatrix} T_{11}(z^{-1}) & T_{12}(z^{-1}) \\ T_{21}(z^{-1}) & T_{22}(z^{-1}) \end{bmatrix},$$

(3)

and each $T_{ij}(z^{-1})$, $i, j = 1, 2$ is described by its denominator and numerator

$$T_{ij}(z^{-1}) = \frac{T_{ij0}(z^{-1})}{Td_{ij}(z^{-1})}.$$

(4)

Then the polynomial $A(z^{-1})$ of (2) and the polynomial matrix $B(z^{-1})$ of (3) are given by

$$A = Td_{11}Td_{12}Td_{21}Td_{22},$$

$$B = \begin{bmatrix} T_{11}Td_{12}Td_{21}Td_{22} \\ T_{21}Td_{12}Td_{21}Td_{22} \\ T_{11}Td_{12}Td_{21}Td_{22} \\ T_{11}Td_{12}Td_{21}Td_{22} \end{bmatrix},$$

(5)

where variable $z^{-1}$ is not shown to save text space in each polynomial of the form $T_{ij}$, $Td_{ij}$ and $Tn_{ij}$.

2. If the plant model is given by a state equation

$$x(k+1) = A_d x(k) + B_d u(k), y(k) = C_d x(k),$$

(6)

then the transfer matrix is

$$T(z^{-1}) = \frac{1}{A(z^{-1})} B(z^{-1}) = C_d(I_d - A_d z^{-1})^{-1} B_d.$$  

(7)

Hence $A(z^{-1})$ of (2) and $B(z^{-1})$ of (3) are

$$A(z^{-1}) = \text{Det}(I_d - A_d z^{-1}),$$

$$B(z^{-1}) = C_d \text{Adj}(I_d - A_d z^{-1}) z^{-1} B_d,$$

(8)

where $I_d$ is the identity matrix of size $n_d$, which is the dimension of the state vector $x$ of (8), Det() is the determinant of a matrix, and Adj() is the adjutant matrix.

(3) The matrices $B_0, B_1, \ldots, B_m$ are calculated sequentially from (8) by Fadeeva’s method [14].

Let the reference input be $\tilde{w}(k)$. Then for the plant (1), the control objective is for the output $y(k)$ to follow $\tilde{w}(k)$.

3. Controller for Multi-Input Multi-Output (MIMO) Plant

3.1 Interactor Matrix

Time-delay of MIMO plant is converted to a plant without time-delay using the interactor matrix [5]. Interactor is defined by the matrix $L(z)$ satisfying

$$\lim_{z \to \infty} L_k(z) T_k(z) = \lim_{z \to \infty} L(z) A(z^{-1})^{-1} B(z^{-1})$$

$$\equiv B_0 : \text{constant non-singular matrix},$$

(9)

$$l_m \equiv \text{the highest degree of } z \text{ in the } L(z).$$

(10)

The interactor matrix for the plant (1) is obtained by the method using singular value decomposition [15]. If the plant model (1) is derived from the transfer function matrix $T(z)$, then the interactor of the plant (1) is the same to the interactor of the transfer function matrix $T(z^{-1})$, and methods to derive the interactor matrix of transfer matrices are already proposed by several authors [5]–[8].

Remark The interactor matrix is not unique.

3.2 Derivation for MPC to MIMO Plants

Using the interactor matrix $L(z)$, define reference $\tilde{w}(k)$, output $\hat{y}(k)$, and numerator $B(z^{-1})$ without time-delay.

$$\hat{y}(k) = L[z]y(k), \quad \tilde{w}(z^{-1}) = L[z]B(z^{-1}), \quad \tilde{w}(k) = L[z]w(k),$$

(11)

and index $J$,

$$J = \sum_{j=1}^{N_2} \left[ \tilde{w}(j + k + j) - \tilde{y}(j + k + j) \right] = \sum_{j=1}^{N_2} \lambda \left[ \Delta u(k + j - 1) \right]^T \left[ \Delta u(k + j - 1) \right],$$

(12)

where $N_2$ is the horizon. The estimate $\tilde{y}(k + j)$ is the predicted value of $\hat{y}(k + j)$, and $\lambda$ is a weighing coefficient. Control input is derived so that $J$ is minimized.

For the plant (1) and the interactor $L(z)$ of (12), the followings are assumed:

1. The polynomial $A(z^{-1})$, the polynomial matrix $B(z^{-1})$, and the interactor matrix $L(z)$ are known. Then, $B(z^{-1})$ is known.

2. At time $k$, the values of references $w(j)$ of future steps from $j = k + l_m + 1$ to $j = k + l_m + N_2$ are known. That is, $\tilde{w}(j)$, $j = k, \ldots, k + N_2$ are known.

85
As the denominator $A[z^{-1}]$ of (2) is a scalar polynomial, multiplication with matrix $L[z]$ is commutative,

$$A[z^{-1}]L[z]=L[z]A[z^{-1}].$$  
(16)

Then the plant (1) is rewritten as a plant without time-delay,

$$A[z^{-1}][\dot{y}(k)]=\hat{B}[z^{-1}]u(k).$$  
(17)

Then MPC for the plant (17) and the index (15) is designed by a standard MPC method [1] for plants without time-delay as follows.

For $j=1,\ldots,N_2$, obtain $(j-1)$th order polynomial matrices $E_j[z^{-1}]$ and $n$th order polynomial matrices $F_j[z^{-1}]$ satisfying the Diophantine matrix equation

$$\Delta E_j[z^{-1}]A[z^{-1}]+z^{-j}F_j[z^{-1}]=I_n, \quad j=0,1,\ldots,N_1, \quad j\geq 1,$$

(18)

$$E_j[z^{-1}]=E_0+E_1z^{-1}+\cdots+E_{j-1}z^{-j+1}, \quad F_j[z^{-1}]=F_0+F_1z^{-1}+\cdots+F_{m_j}z^{-n}, \quad j=0,1,\ldots,N_1,$$

(19)

$$\Delta \approx 1-z^{-1}$$

and $I_m$ is the $m\times m$ identity matrix. Then multiplying the plant (17) by $\Delta E_j[z^{-1}]$ from left side and using (18), the estimate $\hat{y}(k+j)$ is obtained as

$$\hat{y}(k+j)=F_j[z^{-1}]\hat{y}(k)+E_j[z^{-1}]\hat{B}[z^{-1}]\Delta u(k+j).$$  
(21)

Separate the polynomial matrix $E_j[z^{-1}]\hat{B}[z^{-1}]$ into the $(j-1)$th order polynomial matrix $R_j[z^{-1}]$ and the $n$th order polynomial matrix $S_j[z^{-1}]$ as

$$E_j[z^{-1}]\hat{B}[z^{-1}]=R_j[z^{-1}]+z^{-j}S_j[z^{-1}], \quad R_j[z^{-1}]=R_0z^{-j}+z^{-j+1}S_j[z^{-1}],$$

(22)

$$S_j[z^{-1}]=S_0+R_1z^{-j+1}+\cdots+S_{m_j}z^{-n}.$$  
(23)

Then the estimate $\hat{y}(k+j)$ is given by

$$\hat{y}(k+j)=F_j[z^{-1}]\hat{y}(k)+S_j[z^{-1}]\Delta u(k)+R_j[z^{-1}]\Delta u(k+j).$$  
(25)

To combine the $N_2$ equations (25) for $j=1,\ldots,N_2$, into one equation, define

$$h(k)=[h_0(k) \ h_1(k) \ \cdots \ h_{N_2}(k)],$$

(26)

$$h_j(k)=F_j[z^{-1}]\hat{y}(k)+S_j[z^{-1}]\Delta u(k-1), \quad j=1,\ldots,N_2, \quad j\geq 1,$$

$$\hat{y}=[\hat{y}(k+1) \ \hat{y}(k+2) \ \cdots \ \hat{y}(k+N_2)],$$

(28)

$$\Delta u=[\Delta u(k) \ \Delta u(k+1) \ \cdots \ \Delta u(k+N_2-1)],$$

$$\tilde{w}=[\tilde{w}(k) \ \tilde{w}(k+1) \ \cdots \ \tilde{w}(k+N_2)],$$

(29)

$$R=\begin{bmatrix} R_0 & 0 & \cdots & 0 \\ R_1 & R_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ R_{N_2-1} & R_{N_2-2} & \cdots & R_0 \end{bmatrix},$$

(31)

$$A=\begin{bmatrix} A_{m_1} & 0 & \cdots & 0 \\ 0 & A_{m_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{m_1} \end{bmatrix}.$$  
(32)

Then the $N_2$ equations (25) for $j=1,\ldots,N_2$ are expressed by one equation $\hat{y}=RA\tilde{u}+\hat{u}$, and the index $J$ of (15) is

$$J=(\hat{y}-\tilde{w})^T(\hat{y}-\tilde{w})+\Delta \hat{u}^T\Delta \hat{u}=\left(R\Delta \hat{u}+\hat{u} \right)^T \left(R\Delta \hat{u}+\hat{u} \right)+\Delta \hat{u}^T\Delta \hat{u}. \quad (33)$$

Then the optimal $\Delta \hat{u}$ is obtained by $\partial J/\partial \Delta \hat{u}=0$. And the optimal input $\Delta u$ at step $k$ is the first element of the optimal $\Delta \hat{u}$, that is

$$\Delta u(k)=P[z^{-1}]\tilde{w}(k+N_2)-F_p[z^{-1}]\hat{y}(k)-S_p[z^{-1}]\Delta u(k-1), \quad (35)$$

$$[P_1, P_2, \ldots, P_{N_2}] = [I_m, O_{m_1}, \ldots, O_{m_1}], \quad (36)$$

$$O_m=m\times m \text{ zero matrix},$$

$$P[z^{-1}]=P_{N_2}+P_{N_2-1}z^{-1}+\cdots+P_1z^{N_2-1}, \quad (37)$$

$$F_p[z^{-1}]=P_1F_1[z^{-1}]+P_2F_2[z^{-1}]+\cdots+P_{N_2}F_{N_2}[z^{-1}],$$

$$S_p[z^{-1}]=P_1S_1[z^{-1}]+P_2S_2[z^{-1}]+\cdots+P_{N_2}S_{N_2}[z^{-1}].$$

This control law minimizes the index (15) for the plant (17). Since the plant (17) is equivalent to the plant (1) with the relation of (14), and since the output $\hat{y}(k)$ follows the reference $\hat{w}(k)$, the output $y(k)$ follows the reference $\hat{w}(k)$.

But to calculate the law (35), the values of $F_p[z^{-1}]\hat{y}(k)$ are necessary. That is, the values of the outputs $\hat{y}(j)$ from $j=k-l_m+1$ to $k$ are necessary. These values include the future values of $\hat{y}(k)$. It is assumed that the plant (1), that is, the plant (17) is known. Hence, the values of outputs $\hat{y}(k-N_2), \ldots, \hat{y}(k)$ are calculated by using the plant equation (17) and the law (35). That is, first, calculate $\hat{y}(k-l_m-n+1), \ldots, \hat{y}(k-l_m)$ using $y(k-l_m-n+1), \ldots, y(k)$ and the relation (14). Then calculate $\hat{y}(k-l_m+1), \ldots, \hat{y}(k)$ using $\hat{y}(k-l_m-n+1), \ldots, \hat{y}(k-n), u(k-1), u(k-2), \ldots, u(k-l_m-n)$ and the plant equation (17).

3.3 Sequential Algorithm to Solve Matrix Polynomial Equations

To calculate the control input (35), the solutions $F_j[z^{-1}]$ and $S_j[z^{-1}]$ for $j=1,\ldots,N_2$ of the matrix polynomial equations (18) and (22) are required. In these equations, the unknown matrices include so many unknown variables, that is, $E_j[z^{-1}]$ have $m\times m \times N_2$ unknown elements, $F_j[z^{-1}]$ $m\times m \times N \times N_2$, $R_j[z^{-1}]$ $m\times m \times N_2$, and $S_j[z^{-1}]$ $m\times m \times N_2$. Also, the equations are of matrix, and matrix operations are not commutative; hence it should be carefully calculated. So it is difficult to solve the equations directly and numerically, and a sequential algorithm to solve the equations is required.

Let $A_0[z^{-1}]=\Delta A[z^{-1}]$ as

$$A_0[z^{-1}]=[1+a_0, z^{-1}+\cdots+a_{n_0}z^{-n_0}]$$

$$=1+a_{d_0}z^{-1}+\cdots+a_{d_0+n_0}z^{-n_0}. \quad (38)$$

Substitute $A_0[z^{-1}]$ of (38), $E_j[z^{-1}]$ of (19), and $F_j[z^{-1}]$ of (20) into (18) and set the coefficient matrices of the terms $z^{-j}, \ldots, z^{-j-n}$ equal to zero, then the next calculating equations are obtained, where $N_2\leq n$ is supposed for simplicity.

$$E_0=I_m,$$

(39)

$$E_1=-E_0a_{d_0},$$

$$\vdots$$

$$E_{N_2-1}=-E_0a_{d_0(N_2-1)}+\cdots+E_{N_2-1}a_{d_0},$$

$$F_{j0}=-E_0a_{d_0+j}+\cdots+E_{j-1}a_{d_0},$$

(40)

$$\vdots$$

$$F_{j0+1-j}=-E_0a_{d_0+j}+\cdots+E_{j-1}a_{d_0+1-j},$$

$$F_{j0+2-j}=-E_0a_{d_0+j}+\cdots+E_{j-1}a_{d_0+2-j},$$

$$\vdots$$

$$F_{j0+n_0-j}=-E_0a_{d_0+j}+\cdots+E_{j-1}a_{d_0+n_0-j}.$$


Using $l_m$ as the highest degree of $z$ in $L[z],$
\[ L[z] = L_0 + L_1 z + \cdots + L_{l_m} z^{l_m}, \tag{41} \]
the condition (13) is expressed equivalently as [4]
\[ L_0 B_0 = 0, \tag{42} \]
\[ L_0 B_1 + L_1 B_0 = 0, \]
\[ \vdots \]
\[ L_{l_m} B_0 + \cdots + L_0 B_{l_m} = \tilde{B}_0 \text{ (non-singular)}. \]
Using these conditions, supposing $n_0 \geq l_m$ and $m = n_0 - l_m,$
$B[z^{-1}]$ is expressed by
\[ \tilde{B}[z^{-1}] = \tilde{B}_0 + \tilde{B}_1 z^{-1} + \cdots + \tilde{B}_{l_m} z^{l_m}. \tag{43} \]
Substitute $\tilde{B}[z^{-1}]$ of (43), $E_i[z^{-1}]$ of (19), $R_i[z^{-1}]$ of (23), and
$S_j[z^{-1}]$ of (24) into (22) and set the coefficient matrices of the terms
$z^0, z^{-1}, \ldots, z^{-1+m}$ equal to zero, then the next calculating equations are
obtained for $k = 0, \ldots, n,$ supposing $N_2 \leq n.$
\[ R_0 = E_0 \tilde{B}_0, \tag{44} \]
\[ R_1 = E_0 \tilde{B}_1 + E_1 \tilde{B}_0, \]
\[ \vdots \]
\[ R_{N_2-1} = E_0 \tilde{B}_{N_2-1} + \cdots + E_{N_2-1} \tilde{B}_0, \]
\[ S_p = E_0 \tilde{B}_j + \cdots + E_{j-k} \tilde{B}_0, \tag{45} \]
where $E_k = 0$ for $k > N_2,$ and $B_k = 0$ for $k > m.$

4. Simulations

The plant is selected to be unstable, and time-delay steps are
different from each input to output, so that the effect of the
interactor and the proposed control can be clearly shown.
The plant considered is the two-input two-output plant with
the same transfer matrix to the plant used in simulations in [13].
\[ T(z) = \begin{bmatrix} \frac{b_1}{z^2(z + a_1)} & \frac{b_2}{z^3(z + a_2)} \\ \frac{b_3}{z^4(z + a_3)} & \frac{b_4}{z^5(z + a_4)} \end{bmatrix}, \tag{46} \]
\[ a_1 = -1.05, \quad a_2 = -1.03, \quad a_3 = 1.05, \quad a_4 = -1.03, \tag{47} \]
\[ b_1 = 1.0, \quad b_2 = 2.0, \quad b_3 = 3.0, \quad b_4 = 4.0. \tag{48} \]
This plant is transformed to the form of (1), whose denominator
is scalar polynomial $A[z^{-1}],$
\[ A[z^{-1}] = 1 - 3.11 z^{-1} + 4.274 z^{-2} - 4.379 z^{-3} \]
\[ + 3.85 z^{-4} - 1.17 z^{-5}, \tag{49} \]
\[ B[z^{-1}] = B_0 + B_1 z^{-1} + \cdots + B_5 z^{-5}, \tag{50} \]
\[ B_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \]
\[ B_3 = \begin{bmatrix} -2.06 & 2.0 \\ 0 & 0 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 2.111 & -4.16 \\ 3.0 & 4.0 \end{bmatrix}, \]
\[ B_5 = \begin{bmatrix} -2.16 & 4.264 \\ -9.33 & -8.32 \end{bmatrix}, \quad B_6 = \begin{bmatrix} 1.114 & -4.37 \\ 9.67 & 8.52 \end{bmatrix}, \]
\[ B_7 = \begin{bmatrix} 0 & 2.27 \\ -3.34 & -8.74 \end{bmatrix}, \quad B_8 = \begin{bmatrix} 0 & 0 \\ 0 & 4.54 \end{bmatrix}. \]
The parameter of the index (15) is $\lambda = 0.1.$ Reference inputs
$w(k)$ and disturbances $d(k)$ are shown in Fig. 1, and from Fig. 2
to Fig. 5, the solid lines show plant outputs, and the dashed lines are reference outputs.
Simulations of six cases are conducted. In Case 1 to Case 4, the horizons are $N_2 = 5.$ In Case 5 and Case 6, the horizons are
longer than time-delays; $N_2 = 8$ in Case 5 and $N_2 = 9$ in Case 6.

Case 1. Time-delay is not considered, that is, $L[z^{-1}]$ is
\[ L = \begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix}, \quad \lim_{z \to \infty} L(z) T(z) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{51} \]
Hence $L[z^{-1}]$ is not an interactor matrix. The horizon is $N_2 = 5$
and shorter than time-delays. Outputs are divergent fast, and
their graphs are the same as Fig. 2 in [13].

Case 2. Time-delay is considered as $k_m = 3.$ A matrix $L[z^{-1}]$
is determined as $L[z^{-1}] = z I,$ so that the numerator is shifted
by two time-steps, and the first two terms, $B_0$ and $B_1$ are
removed in $\tilde{B}[z^{-1}].$
\[ L z^2 I = \begin{bmatrix} z^3 & 0 \\ 0 & z^3 \end{bmatrix}, \quad \lim_{z \to \infty} L(z) T(z) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{52} \]
This $L[z^{-1}]$ is also not an interactor matrix. Outputs are divergent.
Their graphs are Fig. 2 and are different from Fig. 3
in [13] since the weight $\lambda = 0.1$ is different from $\lambda = 0.14$
in [13].

Case 3. The following interactor $L[z]$ is used.
\[ L = \begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix}, \quad \lim_{z \to \infty} L(z) T(z) = \begin{bmatrix} 1 & 0 \\ 3 & 4 \end{bmatrix} = \tilde{B}_0 \text{(non-singular)}. \tag{53} \]
The interactor and the numerator polynomial matrix $\tilde{B}[z^{-1}]$ are
the same as the Case 3 in [13]. Since the weight $\lambda$ is different
from one in [13], the parameters of the controller (35) are different and are given as
\[ P_1 = \begin{bmatrix} 0.324 & 0.0522 \\ -0.230 & 0.182 \end{bmatrix}, \quad P_2 = \begin{bmatrix} -0.0638 & -0.0608 \\ 0.0990 & 0.0380 \end{bmatrix}, \]
\[ P_3 = \begin{bmatrix} 0.133 & -0.0686 \\ -0.0796 & 0.0493 \end{bmatrix}, \quad P_5 = \begin{bmatrix} 0.152 & -0.0052 \\ -0.115 & 0.0114 \end{bmatrix}, \]
\[ P_5 = \begin{bmatrix} -0.0628 & 0.0196 \\ 0.0351 & -0.0148 \end{bmatrix}, \tag{54} \]
\[ F_p[z^{-1}] = F_{p0} + F_{p1} z^{-1} + \cdots + F_{p5} z^{-5}, \tag{55} \]
\[ F_{p0} = \begin{bmatrix} 4.57 & -0.806 \\ -3.11 & 1.62 \end{bmatrix}, \quad F_{p1} = \begin{bmatrix} -10.3 & 1.86 \\ 7.16 & -3.31 \end{bmatrix}, \]
\[ F_{p2} = \begin{bmatrix} 12.9 & -2.46 \\ -9.01 & 4.11 \end{bmatrix}, \quad F_{p3} = \begin{bmatrix} -13.0 & 2.53 \\ 9.14 & -4.00 \end{bmatrix}, \]
\[ F_{p4} = \begin{bmatrix} 8.79 & -1.63 \\ -6.28 & 2.51 \end{bmatrix}, \quad F_{p5} = \begin{bmatrix} -2.51 & 0.433 \\ 1.83 & -0.673 \end{bmatrix}, \]
\[ S_p[z^{-1}] = S_{p0} + S_{p1} z^{-1} + \cdots + S_{p5} z^{-5}, \tag{56} \]
\[ S_{p0} = \begin{bmatrix} -0.480 & 3.96 \\ -2.11 & -4.98 \end{bmatrix}, \quad S_{p1} = \begin{bmatrix} -0.514 & -8.78 \\ 3.26 & 8.38 \end{bmatrix}, \]
\[ S_{p2} = \begin{bmatrix} -2.47 & 8.86 \\ 0.662 & -8.65 \end{bmatrix}, \quad S_{p3} = \begin{bmatrix} 2.39 & -9.15 \\ -1.74 & 7.82 \end{bmatrix}, \]
\[ S_{p4} = \begin{bmatrix} 0 & 4.87 \\ 0 & -3.55 \end{bmatrix}. \]
Control results are shown in Fig. 3. Outputs follow the references quite well, and the effects of disturbances are not large.

**Case 4.** The following interactor $L[z]$ is used.

\[
L[z] = \begin{bmatrix}
z^3 & 0 \\
-0.75z^3 & 0.25z^3
\end{bmatrix}, \tag{57}
\]

\[
\lim_{z \to \infty} L[z]T(z) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \bar{B}_0 \text{ (non-singular)}.
\]

The interactor and the numerator polynomial matrix $\bar{B}[z^{-1}]$ are the same as Case 4 in [13]. The numerical values of parameters of the controller (35), $P_1, \ldots, P_{N_2}$, $F_p[z^{-1}]$, and $S_p[z^{-1}]$ are omitted to save text space. The control results using the controller with the interactor (57) are shown in Fig. 4. The output $y_1$ follows the reference well, and $y_2$ has spikes when the reference changes. The effects of disturbances are not large.

Comparing Fig. 4 and Fig. 5, it is shown that for the same plant, if the interactor is different, then the controller even designed in the same procedure gives different results.

**Case 5.** As in Case 1, time-delay is not considered. The horizon is set $N_2 = 8$, which is longer than time-delay steps. The simulation result is the same as Case 2 and outputs are divergent as shown in Fig. 2.

**Case 6.** As in Cases 1 and 5, time-delay is not considered. The horizon is set $N_2 = 9$, which is longer than Case 5. The simulation result is given in Fig. 5. When disturbances do not exist, control results are well, but when disturbances exist, their effects are large since a longer horizon brings the inputs of the longer time interval in the index and the outputs in the index are affected by disturbances in the longer time interval.

**5. Conclusion**

This paper proposed a design method of MPC for MIMO plants with time-delay by using an interactor matrix and a sequential procedure to solve the matrix polynomial Diophantine equations required to be solved in the design. The equations are of matrix polynomials, and matrix calculations are not commutative; hence it is not easy to solve the equations, and it is necessary to obtain a sequential solving procedure.

Also, difficulty in the design of MPC of MIMO plants comes from the fact that a plant transfer function is a matrix which is
not commutative in multiplication. This paper avoided this difficulty by obtaining a plant transfer function with a scalar polynomial denominator. And to handle the time-delay in MIMO plants, an interactor matrix was used to shift the outputs by time-delay steps. Then the design problem with time-delay was reduced to a problem without time-delay.

There exist designs of MPC for time-delay plants by using a longer horizon than the time-delay steps. In this paper, it was shown by simulations that MPC having a long horizon was sensitive to disturbances and that the proposed MPC was less sensitive. The controller uses the future values of plant outputs, and they are to be calculated by using the plant model and the control law. This fact means that control results are strongly sensitive to the uncertainty of the plant model. To enhance the robustness to the uncertainty of the plant model is also future work.

Acknowledgments

The authors would like to thank the financial support by JSPS KAKENHI Grant Number 17K06225.

References

[1] E.F. Camacho and C. Bordux, Model Predictive Control, Chapter 6, Springer, 1998.
[2] T. Itoh: Control in Chemical Industry, Chapter 1, Corona Publishing, 2002 (in Japanese).
[3] Private communication with a control engineer in a chemical company in a technical meeting at October 21, 2019, Kurashiki, Japan
[4] S.L. Shah, C. Mohtadi, and D.W. Clarke: Multivariable adaptive control without a prior knowledge of the delay matrix, Systems & Control Letters, Vol. 9, No. 4, pp. 295–306, 1987.
[5] W.A. Wolovich and P.L. Falb: Invariants and canonical forms under dynamic compensation, SIAM J. Control and Optimization, Vol. 14, No. 6, pp. 996–1008, 1976.
[6] H. Elliott and W.A. Wolovich: Parametrization issues in multivariable adaptive control, Automatica, Vol. 20, No. 5, pp. 533–545, 1984.
[7] Y. Mutoh: Interactor matrix and structure at infinity in linear multivariable control systems, Journal of SICE, Vol. 32, No. 3, pp. 237–252, 1993 (in Japanese).
[8] T. Tamura: Designs of model reference adaptive controllers for multivariable systems, 21st Seminar XXI of Technical Committee on Adaptive Control, Report No. 63–2, 1988 (in Japanese).
[9] B. Huang and S.L. Shah: The role of the unitary interactor matrix in the explicit solution of the singular LQ output feedback control problem, Automatica, Vol. 32, No. 11, pp. 2071–2075, 1997.
[10] A. Basel, K. Hirata, and K. Sugimoto: Generalization of feedback error learning (FEL) to MIMO systems, Trans. SICE, Vol. 43 No. 4, pp. 293–302, 2007.
[11] W. Kase: Iterative learning control systems based on inverse systems and interactor matrix, Trans. SICE, Vol. 45, No. 10, pp. 538–540, 2009 (in Japanese).
[12] H.N. Koivo: A multivariable self-tuning controller, Automatica, Vol. 16, No. 4, pp. 351–366, 1980.
[13] A. Inoue, M. Deng, T. Sato, A. Yanou, and T. Henmi: Design of model predictive control of multi-variable time-delay plants using interactor matrix, Proceedings of SICE Annual Conference, pp. 1535–1540, 2019.
[14] H. Maeda: Linear Systems, pp. 53–54, Asakura Publishing, 2001 (in Japanese).
[15] H. Katoh and Y. Funahashi: New method to derive generalized interactor matrices, Trans. SICE, Vol. 33, No. 1, pp. 63–65, 1997 (in Japanese).