Design of poles of controller in strongly stable generalized predictive control using symbolic computation software

Akira Inoue[1] and Mingcong Deng[2]

***Abstract*— This paper proposes a method to specify straightforwardly poles of controllers of strongly stable generalized predictive control(GPC) systems using symbolic computation software. Strongly stable control system is defined as a system having both of stable poles of closed loop systems and stable poles of controllers. That is, the system is stable even when feedback loop is cut by an accident. The authors have already derived a strongly stable controller for GPC systems by extending the controllers to two degree of freedom compensators and using coprime factorization approach. To design the strongly stable systems, we need to specify both of the poles of closed-loop and poles of controllers. So far we do not have a method to deign directly the poles and we need to use trial-and-error method to specify the controller's poles. To specify the poles directly, we need to calculate design parameters from given desirable poles analytically, not numerically. To calculate poles analytically, symbolic computation software is useful. So far symbolic computation software requires large amount of computer resources and until recently, the software was not practically available. But, recently computer technology is developed so fast that the software becomes practically usable. Hence this paper proposes to use the symbolic computation software in design of GPC controller poles.**

I. INTRODUCTION

In industry, safety is the most important issue. As a safe model predictive control(MPC), strongly stable MPC is proposed[1]. The strongly stable control system is defined as a system having both of stable poles of closed loop systems and stable poles of controllers[2]. That is, the system is stable even when feedback loop is cut by an accident. The proposed strongly stable GPC is obtained by extending the controllers to two degree of freedom compensators and using coprime factorization approach and have applied the strongly stable MPC to experimental systems[3], [4]. Also studies exist to extend the strongly stable MPC to a continuous-time system and to apply to a real plant[5].

To design the strongly stable systems, we need to specify both of the poles of closed-loop and poles of controllers. In applying the controller to real plants, plant dynamics changes frequently and when the change occurs, the controllers should be redesigned to follow the change. Hence a design method for design parameters to be determined directly is required. So far we do not have such design method and we depend on trial-and-error methods. That is, first, give design parameter candidates, then calculate numerically poles of controllers, if the poles are not desirable, then try other parameter values. This procedures are repeated until the desirable poles are obtained.

There exists a paper using two degree of freedom controller in GPC[6]. Their design method is to insure stability and robustness and to use an optimization programming. Also exists a paper to propose a computational method for strongly stable GPC[7]. The paper is concerned to the closed-loop poles not controller poles.

To avoid these trial-and-error procedures, we need to calculate design parameters from given desirable poles analytically, not numerically. Since several matrix equations to be solved and a matrix inversion are included, to obtain the analytical expressions in GPC, these calculations are impossible by hand. To calculate such expressions analytically, symbolic computation software are useful. Once the analytical expressions are obtained, then from the desirable poles the design parameters are determined. So far symbolic computation software requires large amount of computer resources and until recently, the software was not practically available. But, recently computer technology is developed so fast that software becomes practically usable. Hence this paper proposes to use the symbolic computation software in design of GPC controller poles.

Already some studies have tried to use symbolic computation software in controller design[8], [9], but there does not exist researches to apply symbolic computation software to the design of GPC and this paper is the first one trying the application.

This paper is organized as follows. Section II gives the problem setting. Section III reviews strongly stable GPC. In section IV proposes a design procedure to use symbolic computation in the design of poles of controller in strongly stable GPC. Section V is for simulation to show the design procedure. Finally, Section VI is the conclusion of this paper.

Notation z^{-1} denotes time-delay, that is, $z^{-1}y(t) = y(t-1)$. Δ denotes as $\Delta = 1 - z^{-1}$. Polynomials of z^{-1} are written as $A[z^{-1}]$, whereas rational functions of z^{-1} are as $A(z^{-1})$.

II. PROBLEM SETTING

The plant to be considered is given by the discrete-time single-input single-output system described by the following equations.

$$A[z^{-1}]y(t) = z^{-k_m}B[z^{-1}]u(t) + C[z^{-1}]\frac{\xi(t)}{\Delta} \quad (1)$$

$$A[z^{-1}] = 1 + a_1z^{-1} + a_2z^{-2} + \cdots + +a_nz^{-n} \quad (2)$$

$$B[z^{-1}] = b_0 + b_1z^{-1} + b_2z^{-2} + \cdots + +b_mz^{-m} \quad (3)$$

[1]Akira Inoue is Professor Emeritus, Okayama University. Japan inoue-09@t.okadai.jp

[2]Mingcong Deng is with the Department of Electrical and Electronics Engineering, Tokyo University of Agriculture and Technology, Tokyo, Japan deng@cc.tuat.ac.jp

$$C[z^{-1}] = 1 + c_1 z^{-1} + \cdots c_l z^{-l} \tag{4}$$

where $u(t)$ is scalar input and $y(t)$ is scalar output, $k_m \geq 1$ is delay-time and $\xi(t)$ is uniform random measurement noise. $A[z^{-1}]$, $B[z^{-1}]$ are polynomials with n, mth-order, coprime with each other. Polynomial $C[z^{-1}]$ is of order l. For simplicity, it is assumed that $k_m = 1$.

The control objective is for output $y(t)$ to follow reference $w(t)$ and to have desirable response. To attain the objective, in GPC, the following generalized index J is to be minimized.

$$\begin{aligned} J &= E\left[\sum_{j=1}^{N_2}\{y(t+j) - y_m(t+j)\}^2 \right. \\ &\quad \left. + \sum_{j=1}^{N_3} \lambda \{\Delta u(t+j-1)\}^2 \right] \end{aligned} \tag{5}$$

where N_2 and N_3 are output horizon and control horizon and for simplicity N_3 is set as $N_3 = N_2$. λ are weighting coefficients and the poles of the closed-loop are determined by these coefficients. The expectation of the index J is averaged over random noise $\xi(t)$, $\xi(t-1)$, $\cdots$. $y_m(t+j)$ are the outputs of the following reference model.

$$\begin{aligned} y_m(t) &= y(t) \\ y_m(t+j) &= \alpha y_m(t+j-1) + (1-\alpha) w(t) \end{aligned} \tag{6}$$

$w(t)$ is reference input, α is a design parameter to determine transient responsee and $0 \leq \alpha \leq 1$.

III. Strongly Stable Model Predictive Control

This section summaries the design procedure of the strongly stable GPC[1]. The strong stability includes two stability as shown in Fig.1, that is,
(i) the closed-loop is stable,
(ii) also controller itself is stable so that the control input will not run explosively when feedback loop is breakdown.

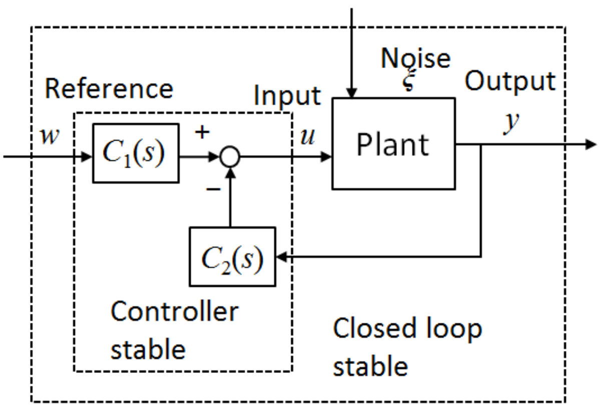

Fig. 1. Strongly stable control system

The design procedure has the following 7 steps[1]:
Step 1. For $j = 1, \cdots, N_2$, obtain $(j-1)$th-order polynomial $E_j[z^{-1}]$ and nth-order polynomial $F_j[z^{-1}]$ satisfying the following Diopahtine equation

$$\begin{aligned} C[z^{-1}] &= \Delta A[z^{-1}] E_j[z^{-1}] + z^{-j} F_j[z^{-1}] \quad (7) \\ E_j[z^{-1}] &= 1 + e_1 z^{-1} + \cdots + e_{j-1} z^{-(j-1)} \quad (8) \\ F_j[z^{-1}] &= f_0^j + f_1^j z^{-1} + \cdots + f_n^j z^{-n} \quad (9) \end{aligned}$$

where coefficients $e_1, \cdots, e_{N_2}$ of $E_j[z^{-1}]$ are determined independently to j.
Step 2. For $j = 1, \cdots, N_2$, obtain $(j-1)$th-order polynomial $R_j[z^{-1}]$ and n_3th-order polynomial $S_j[z^{-1}]$ by decomposing the following equation, where $n_3 = \max\{m, l\} - 1$,

$$\begin{aligned} E_j[z^{-1}] B[z^{-1}] &= C[z^{-1}] R_j[z^{-1}] + z^{-j} S_j[z^{-1}] \quad (10) \\ R_j[z^{-1}] &= r_0 + r_1 z^{-1} + \cdots + r_{j-1} z^{-(j-1)} \quad (11) \\ S_j[z^{-1}] &= s_0^j + s_1^j z^{-1} + \cdots + s_{n_3}^j z^{-n_3} \quad (12) \end{aligned}$$

where coefficients $r_0, r_1, \cdots, r_{N_2}$ of $R_j[z^{-1}]$ are determined independently to j.
Step 3. Obtain coefficients $p_1, \cdots, p_{N_2}$ using the following equation and define nth-order polynomial $F_p[z^{-1}]$, n_3th-order polynomial $S_p[z^{-1}]$ and (N_2-1)th-order polynomial $P[z^{-1}]$.

$$[p_1, \cdots, p_{N_2}] = [1, 0, \cdots, 0](R^T R + \Lambda)^{-1} R^T \tag{13}$$

$$R = \begin{bmatrix} r_0 & 0 & \cdots & 0 \\ r_1 & r_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ r_{N_2-1} & r_{N_2-2} & \cdots & r_0 \end{bmatrix} \tag{14}$$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_{N_2} \end{bmatrix} \tag{15}$$

$$F_p[z^{-1}] = \sum_{j=1}^{N_2} p_j F_j[z^{-1}] \tag{16}$$

$$S_p[z^{-1}] = \sum_{j=1}^{N_2} p_j S_j[z^{-1}] \tag{17}$$

$$P[z^{-1}] = \sum_{j=1}^{N_2} p_{N_2-j+1} z^{j-1} \tag{18}$$

Step 4. For $j = 1, \cdots, N_2$, obtain $(n+m)$th-order polynomial $D_j[z^{-1}]$ and $D_p[z^{-1}]$ using the following equations

$$D_j[z^{-1}] = \Delta A[z^{-1}] S_j[z^{-1}] + B[z^{-1}] F_j[z^{-1}] \tag{19}$$

$$D_P[z^{-1}] = \sum_{j=1}^{N_2} p_j D_j[z^{-1}] \tag{20}$$

Then calculate $(n+1)$th-order polynomial $T[z^{-1}]$ satisfying the following equation

$$C[z^{-1}] T[z^{-1}] = \Delta A[z^{-1}] C[z^{-1}] + z^{-1} D_p[z^{-1}] \tag{21}$$

Step 5. Using polynomial $T[z\ -1]$, obtain coprime rational expression $G(s)$ of system (1) by the following equations[2].

$$G(z^{-1}) = N_G(z^{-1})/D_G(z^{-1}) \quad (22)$$
$$N_G(z^{-1}) = z^{-k_m}B[z^{-1}]/T[z^{-1}] \quad (23)$$
$$D_G(z^{-1}) = A[z^{-1}]/T[z^{-1}] \quad (24)$$

Then $N(z^{-1})$ and $D(z^{-1})$ satisfy the following Bezout identity

$$X(z^{-1})N_G(z^{-1}) + Y(z^{-1})D_G(z^{-1}) = 1 \quad (25)$$
$$X(z^{-1}) = F_p[z^{-1}]/C[z^{-1}] \quad (26)$$
$$Y(z^{-1}) = (C[z^{-1}] + z^{-1}S_p[z^{-1}])\Delta/C[z^{-1}] \quad (27)$$

and $N_G(z^{-1}),\ D_G(z^{-1}) \in RH_\infty$ are coprime.

Step 6. Introducing new design parameter rational functions $U(z^{-1}), K(z^{-1}) \in RH_\infty$, two degree of freedom stabilizing controller for $G(z^{-1})$ is given by the following Youla Parameterization[2].

$$u(t) = C_1(z^{-1})w(t) - C_2(z^{-1})y(t) \quad (28)$$
$$C_1(z^{-1}) = (Y(z^{-1}) - U(z^{-1})N(z^{-1}))^{-1}K(z^{-1}) \quad (29)$$
$$C_2(z^{-1}) = (Y(z^{-1}) - U(z^{-1})N(z^{-1}))^{-1} (X(z^{-1}) + U(z^{-1})D(z^{-1})) \quad (30)$$

Using new parameter polynomials $U_d[z^{-1}]$ and $U_n[z^{-1}]$, parameter $U(z^{-1})$ is defined as

$$U(z^{-1}) = \frac{U_n[z^{-1}]}{U_d[z^{-1}]}T[z^{-1}] \quad (31)$$

where $U_d[z^{-1}]$ is chosen as stable polynomial, Then controller (28) described by rational function is expressed in polynomial form as

$$\begin{aligned}&\{U_d[z^{-1}](C[z^{-1}]+z^{-1}S_p[z^{-1}]\Delta \\ &\quad -U_n[z^{-1}]z^{-1}C[z^{-1}]B[z^{-1}]\}u(t) \\ &= U_d[z^{-1}]C[z^{-1}]P[z^{-1}]y_M(t+N_2) \\ &\quad -(U_d[z^{-1}]F_p[z^{-1}] + U_n[z^{-1}]C[z^{-1}]A[z^{-1}])y(t) \quad (32)\end{aligned}$$

Step 7. Using the controller (32), the poles of the closed-loop system are given by the zeroes of polynomial $T[z^{-1}]$ and the poles of the controller are zeroes of the following equation

$$T_c[z^{-1}] = U_d[z^{-1}](C[z^{-1}] + z^{-1}S_p[z^{-1}])\Delta - U_n[z^{-1}]z^{-1}C[z^{-1}]B[z^{-1}] \quad (33)$$

Since zeros of $T[z^{-1}]$ are independent to parameters $U_d[z^{-1}]$ and $U_n[z^{-1}]$, to design a strongly stable GPC, it is necessary to select polynomials $U_d[z^{-1}]$ and $U_n[z^{-1}]$ so that zeroes of $T_C[z^{-1}]$ are stable. But the selecting procedure relies on trial-and-error method as shown in "conventional design procedure" of Fig.2.

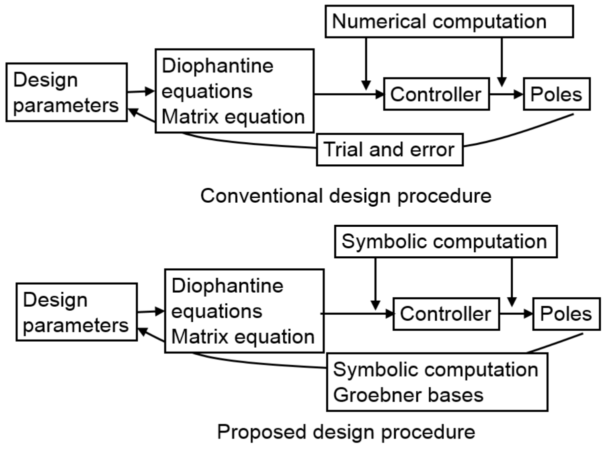

Fig. 2. Design procedures

IV. Design using symbolic computation software

In this section, a design procedure to specify the controller poles to given $p_1, \cdots, p_{n_c}$ is proposed.

First define the orders of parameters $U_n[z^{-1}]$ and $U_d[z^{-1}]$ as n_n and n_d. Then $U_n[z^{-1}]$ and $U_d[z^{-1}]$ are given by the following polynomials

$$U_n[z^{-1}] = u_{n0} + u_{n1}z^{-1} + \cdots + u_{nn_d}z^{-n_n} \quad (34)$$
$$U_d[z^{-1}] = 1 + u_{d1}z^{-1} + \cdots + u_{dn_d}z^{-n_d} \quad (35)$$

And the coefficients $u_{n0}, \cdots, u_{nn_n}$ and $u_{d1}, \cdots, u_{dn_d}$ of $U_n[z^{-1}]$ and $U_d[z^{-1}]$ are determined so that zeroes of polynomial $T_c[z^{-1}]$ of (33) are equal to $p_1, \cdots, p_{n_c}$. The order n_c of $T_c[z^{-1}]$ is

$$n_c = \max\{n_d + l + 1, n_d + \max\{m, l\}, n_n + 1 + l + m\} \quad (36)$$

Then $T_c[z^{-1}]$ is polynomial of z^{-1}

$$T_c[z^{-1}] \equiv l_1 + l_2z^{-1} + \cdots + l_{n_c}z^{-n_c} \quad (37)$$

and its coefficients $l_1, l_2, \cdots, l_{n_c}$ are polynomials of $u_{n0}, \cdots, u_{nn_n}, u_{d1}, \cdots, u_{dn_d}$.

n_c poles of controller should be specified to $p_1, \cdots, p_{n_c}$, then polynomials $T_c[z^{-1}]$ should be

$$T_c[z^{-1}] = (1 - p_1z^{-1}) \cdots (1 - p_{n_c}z^{-1}) \quad (38)$$
$$= 1 + q_1z^{-1} + \cdots + q_{n_c}z^{-n_c} \quad (39)$$

where coefficients $q_1, \cdots, q_{n_c}$ are polynomials of $p_1, \cdots, p_{n_c}$. Hence equations (37) and (39) should be equal to each other, then the coefficients $U_n[z^{-1}]$, $U_d[z^{-1}]$ should be determined so that the coefficients of the two equations are equal.

$$l_1 = q_1, \cdots, l_{n_c} = q_{n_c} \quad (40)$$

Equations (40) are simultaneous algebraic equations with unknown variables $u_{n0}, \cdots, u_{nn_n}, u_{d1}, \cdots, u_{dn_d}$ including the coefficients $p_1, p_2, \cdots, p_{n_c}$ and coefficients of polynomials $C[z^{-1}]$, $B[z^{-1}]$ and $S_p[z^{-1}]$ as parameters. These simultaneous algebraic equations are calculated analytically using symbolic computation software and solved. Then substituting the desirable poles $p_1, p_2, \cdots, p_{n_c}$ into the solved solutions of the simultaneous algebraic equations (40), the design parameters $u_{n0}, \cdots, u_{nn_n}, u_{d1}, \cdots, u_{dn_d}$ are obtained directly, without using trial-and-error method as is shown in "proposed design procedure" in Fig. 2.

V. Simulation Example

Suppose polynomials $A[z^{-1}]$, $B[z^{-1}]$ and $C[z^{-1}]$ of the plant and the output horizon N_2 are given as

$$\begin{aligned} A[z^{-1}] &= 1 + a_1 z^{-1} + a_2 z^{-2} = 1 + 0.6z^{-1} + 0.7z^{-2}, \\ B[z^{-1}] &= b_0 + b_1 z^{-1} = 0.5 - 1.5z^{-1}, \\ C[z^{-1}] &= 1, \quad N_2 = 5, \quad \lambda = 1, \quad \alpha = 0 \end{aligned} \tag{41}$$

The reference signal is rectangular wave with amplitude 1, period 100 sampling times and since $\alpha = 0$, the output of reference model is $y_m(t) = w(t)$. Then polynomial $S_p[z^{-1}]$ of (17) is

$$S_p[z^{-1}] = s_{p0} = 0.095793 \tag{42}$$

Let the design parameters $U_n[z^{-1}]$ and $U_d[z^{-1}]$ be as

$$U_n[z^{-1}] = u_{n0} + u_{n1} z^{-1}, U_d[z^{-1}] = 1 + u_{d1} z^{-1} \tag{43}$$

Then design parameters are u_{n0}, u_{n1} and u_{d1}. Let the poles to be specified be denoted as p_1, p_2 and p_3, Then equations to be solved are

$$\begin{aligned} &-u_{n1} b1 - u_{d1} s_{p0} = -p_3 p_2 p_1, \\ &-u_{n1} b0 - u_{n0} b1 + (u_{d1} - 1) s_{p0} - u_{d1} \\ &\qquad = -((-p_2 - p_3) p_1 - p_3 p_2), \\ &-u_{n0} b0 + s_{p0} + u_{d1} = -(p_1 + p_2 + p_3 - 1) \end{aligned} \tag{44}$$

Using symbolic computation software, these equations are solved as

$$u_{d1} = -u_{d1n}/u_{d1d} \tag{45}$$

$$\begin{aligned} u_{d1n} &= (p_3 p_2 p_1) b_0^2 + (s_{p0} + (p_2 + p_3) p_1 + p_3 p_2) b_1 b_0 \\ &\quad + (s_{p0} + p_1 + p_2 + p_3 - 1) b_1^2 \\ u_{d1d} &= (-s_{p0} b_0^2 - s_{p0} b_1 b_0 + b_1 b_0 + b_1^2) \end{aligned}$$

$$u_{n1} = -u_{n1n}/u_{n1d} \tag{46}$$

$$\begin{aligned} u_{n1n} &= +(-s_{p0}^2 + (((p_3 - 1) p_2 - p_3) p_1 - p_3 p_2) s_{p0} \\ &\quad - p_3 p_2 p_1) b_0 \\ &\quad + (-s_{p0}^2 + (-p_1 - p_2 - p_3 + 1) s_{p0} - p_3 p_2 p_1) b_1 \\ u_{n1d} &= (-s_{p0} b_0^2 + (-s_{p0} + 1) b_1 b_0 + b_1^2) \end{aligned}$$

$$u_{n0} = -u_{n0n}/u_{n0d} \tag{47}$$

$$\begin{aligned} u_{n0n} &= ((-s_{p0}^2 + (-p_1 - p_2 - p_3 + 1) s_{p0} + (-p_3 p_2 p_1) b_0 \\ &\quad + (-s_{p0}^2 + (-p_1 - p_2 - p_3 + 1) s_{p0} \\ &\quad + (-p_2 - p_3 + 1) p_1 + (-p_3 + 1) p_2 + p_3 - 1) b_1) \\ u_{n0d} &= (s_{p0} b_0^2 + (s_{p0} - 1) b_1 b_0 - b_1^2) \end{aligned}$$

When desirable poles p_1, p_2 and p_3 are given, then substituting these poles into these equations, parameters u_{d1}, u_{n1} and u_{n0} to specify the poles of controllers are directly calculated.

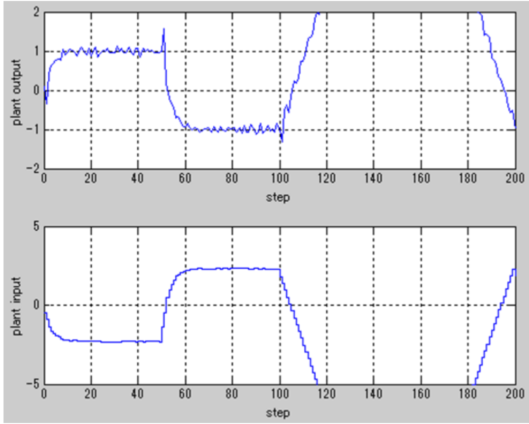

Fig. 3. Simulation Run #1

That is, once the desirable poles p_1, p_2 and p_3 are selected, then the design parameters are determined.

To show usefulness of these equations, three simulation runs are conducted. In the simulations, to show effectiveness of strongly stable controllers, feedback is cut at 100th sampling time. Also. uniform random noise ξ with amplitude ± 0.1 is added.

Simulation #1: Non-extended model predictive control, that is, u_{d1}, u_{n1} and u_{n0} are selected equal to 0. Since controller has unstable pole, $z^{-1} = 1$, after the feedback is cut, the controller output is divergent.

Simulation #2: This case is strongly stable. Parameters are selected by trial-and-error method[1], poles of controller include complex numbers and after the feedback is cut, the response is oscillatory.

Simulation #3: The extended controller is designed using the method proposed in this paper, that is, first, desirable poles are selected as $p_1 = 0.7. p_2 = 0.6, p_3 = 0.5$. Then the design parameters are determined using equations (45) $\sim$ (47). Simulation run shows that the response is stable after feedback is cut and also not oscillatory.

Simulation results are summarized in Table 1.

TABLE I
SIMULATION

No,	Design	u_{n0}, u_{n1}, u_{d1}	Poles of controller	Fig.
1	Not strongly stable	0, 0, 0	1.0, -0.0958	Fig.3
2	Strongly stable trial-and-error	0.4, 0, 0	$0.552 \pm 0.447i$	Fig.4
3	Strongly stable proposed	0.264, -0.189 -0.774	0.7, 0.6 0.5	Fig.5

VI. Conclusion

In this paper, a design procedure is proposed for strongly stable generalized model predictive control. So far, the poles of controller are selected by trial-and-error method. In the design procedure given in this paper, the design parameters are decided from the given desirable poles straightforwardly. To calculate the parameters, the expressions to

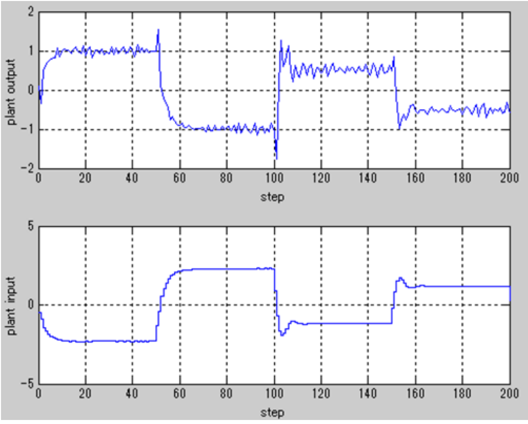

Fig. 4. Simulation Run #2

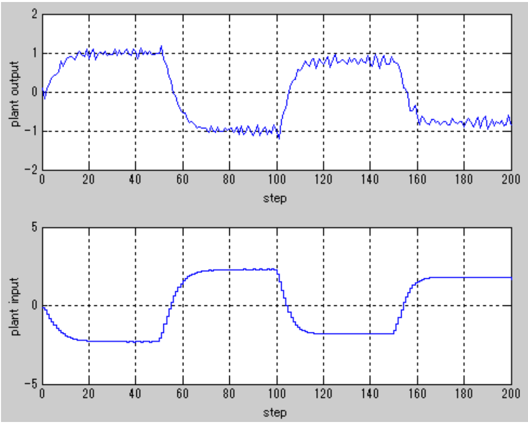

Fig. 5. Simulation Run #3

show the relations between the poles and the parameters are first obtained, then substituting the desired poles into the relations, the values of parameters are obtained. To calculate the expressions, symbolic computation software is used. Simulation runs show that better output responses are obtained in the case that controller is designed by the method of this paper than the case that the parameters are obtained by trial-and-error method in previous paper.

To apply the proposed deign method to an exprimental plant or a practical process plant is a future work.

REFERENCES

[1] Inoue, A. Yonou, A. and Hirashima, Y. : 'A Design of a Strongly Stable Generalized Predictive Control Using Coprime Factorization Approach', *Proc. American Control Conference*, June, 1999, pp.652–656.1999

[2] Vidyasagar M.: 'Control System Synthesis, A Factorization Approach', MIT Press. 1985

[3] Deng, M., Inoue, A., Yanou, A. and Okazaki, S.: 'Stable Anti-windup Continuous-time Generalized Predictive Control to a Process Control Experimental Ssystem', *Measurement and Control: The Journal of the Inst. of Measurement and Control, UK*, Vol. 40, No. 4, pp.120–123. 2007

[4] Deng, M., Inoue, A., Ishibashi, N. and Yanou, A. : 'A Multivariable Continuous-time Aanti-windup Generalized Predictive Control for an Aluminum Plate Thermal Process',2007 *International Journal of Modeling, Identification, and Control*, Vol. 2, No. 2, pp.130–137.

[5] Ishibashi, N., Deng, M. and Inoue, A. : 'Robust Temperature Control of a Reformer by Using Stable Continuous-time Generalized Predictive Control', *ICE Trans. on Industrial Application*, Vol. 7, No. 7, pp.48–52(In Japanese).2008

[6] J. R. Gossper, B. Kouvaritakis and J. A. Rossiter: Cautious Stable Predictive Control: a Guaranteed Stable Predictive Control Algorithm with Law Input Activity and Good Robustness, Int. J. Control, Vol.67, No.5, pp.675-697, 1997

[7] A. Yanou, M. Minami and T. Matsuno: Extended Self-Tuning Generalized Predictive Control with Computation Reduction Focused on Closed-Loop Characteristics, 11th IFAC Int. workshop on ALCOSP, pp.51-56, 2013

[8] Park, H. and Regensburger, G.: 'Greobner Bases in Control Theory and Signal Processing', *Walter de Gruyter*,2007

[9] Shin, H.S. and Lall, S.: 'Optimal Decentralized Control of Linear Systems via Groebenr Bases and Variable Elimination', *Proc. American Control Conference*, June 30-July 2, pp5608–5613, 2010