2016 IEEE 55th Conference on Decision and Control (CDC)
ARIA Resort & Casino
December 12-14, 2016, Las Vegas, USA

Tracking control method for a plant with continuous time unstable zeros: Finite preactuation based on state trajectory regeneration by using redundant order polynomial

Wataru Ohnishi[1], Hiroshi Fujimoto[1]

***Abstract*— A plant with unstable zeros is considered to be difficult to control because of initial undershoot of step response and unstable poles of its inversion system. A plant may have unstable zeros in discrete time domain because of following reasons: 1) non-collocation of actuators and sensors and 2) discretization by zero-order hold. We proposed a solution for these problems by using a multirate feedforward control with state trajectory generation based on time axis reversal. However, this method requires preactuation for negative infinite time. This paper proposes a state trajectory regeneration method via redundant order polynomial for the negative finite time. Although this method abandons perfect tracking during preactuation, it guarantees perfect tracking for a positive time domain. Moreover, the tracking error during finite time preactuation is reduced by the regenerated state trajectory obtained by the optimized redundant order polynomial. The validity of the proposed method is demonstrated through simulations.**

I. Introduction

A plant with unstable zeros is known to be difficult to control because of unstable poles of its inversion system and initial undershoot of step response, as shown in Fig. 1. The zeros of the discretized transfer function can be classified as follows [1][2]: 1) intrinsic zeros and 2) discretization zeros [3]. Intrinsic zeros correspond to zeros of the continuous time transfer function. The others are called discretization zeros. Discretization zeros are unstable when the relative order of the continuous time plant is greater than two even without continuous time unstable zeros [3].

To design a stable feedforward controller for a plant with unstable zeros, approximated inversion-based feedforward controllers are proposed: for example, nonminimum-phase zeros ignore (NPZI) [4], zero-phase-error tracking controller (ZPETC) [5], and zero-magnitude-error tracking controller (ZMETC) [6]. These controllers handle aforementioned problems 1) and 2) simultaneously because they are designed using discretized transfer functions.

Compensation methods have been proposed for unstable intrinsic and discretization zeros through preactuation and preview [7][8]. In addition, these methods compensate for intrinsic and discretization zeros simultaneously. A continuous time domain approach was proposed in [9]. This method solves the differential equation in a continuous time domain; however, the reference trajectory must be defined by an equation in the positive time domain.

[1]The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba, 277-8561, Japan, ohnishi@hflab.k.u-tokyo.ac.jp, fujimoto@k.u-tokyo.ac.jp

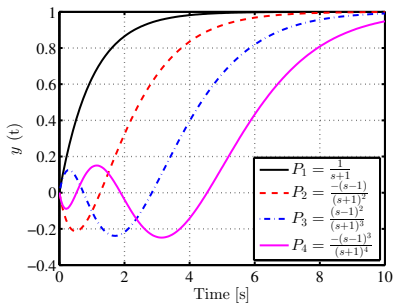


Fig. 1. Step response comparison. P_1 is a first order transfer function without an unstable zero. P_2, P_3, and P_4 have one, two, and three unstable zero(s), respectively. Step responses of the system with unstable zero(s) creates undershoot.

Our previous paper [10] proposes a method to solve problems 1) and 2) separately. The unstable zeros in the continuous time transfer function are managed through a state trajectory generation based on time axis reversal and preactuation commands. This method can be applied for any reference trajectory, given its $n-1$ th derivative. Here, n denotes the order of the plant in the continuous time transfer function. Next, the plant discretization problem is solved through the multirate feedforward control [11]. However, this method requires preactuation of negative infinite time.

This paper proposes a finite time preactuation method based on state trajectory regeneration by using a redundant polynomial in the negative time domain. By using the state trajectory regeneration and the finite time preactuation with multirate feedforward, the initial state variable of the plant can be matched with the desired initial state. Although this method abandons perfect tracking for the reference trajectory r during preactuation, it guarantees perfect tracking for the positive time domain. The tracking error during preactuation is reduced by the regenerated state trajectory obtained through the optimized redundant order polynomial. The validity of the proposed method is demonstrated through simulations.

II. Preactuation Perfect Tracking Control [10]

We proposed a preactuation perfect tracking control (PPTC) method in [10] to design a stable inversion feedforward controller for plants with unstable intrinsic and discretization zeros. This method solves the unstable zeros inversion problem in two steps. The stable inversion for unstable intrinsic zeros generated through continuous time

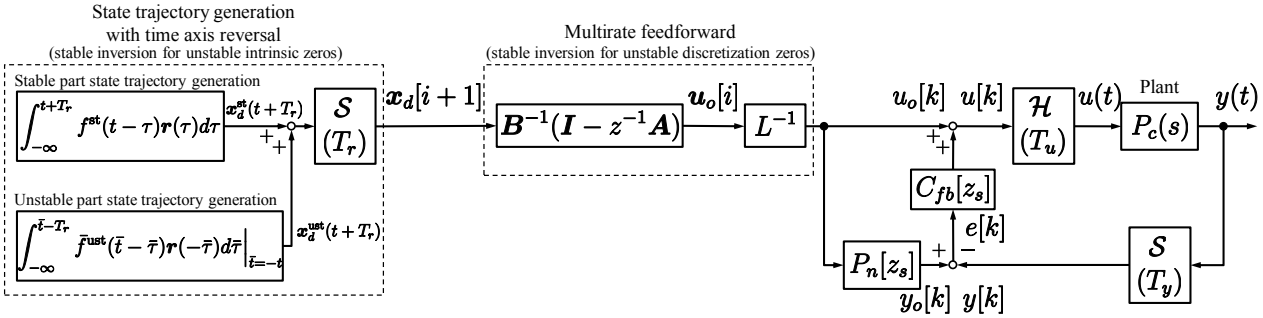


Fig. 2. Multirate feedforward control with state trajectory generation based on time axis reversal [10]. $\mathcal{S}$, $\mathcal{H}$, and L denote a sampler, holder, and lifting operator [12], respectively. z and z_s denote e^{sT_r} and e^{sT_u}, respectively.

unstable zeros are calculated using a time axis reversal in a continuous time domain. The stable inversion for unstable discretization zeros are calculated using a multirate feedforward proposed in [11].

A. Plant definition

A nominal plant in a continuous time domain is defined as a controllable canonical form:

$$P_c(s) = \frac{B(s)}{A(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_0}{s^n + a_{n-1}s^{n-1}\cdots + a_0}, \quad (1)$$

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_c\boldsymbol{x}(t) + \boldsymbol{b}_c u(t), \quad (2)$$

$$y(t) = \boldsymbol{c}_c\boldsymbol{x}(t), \quad (3)$$

where

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \boldsymbol{A}_c = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ & & & \ddots & \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \quad (4)$$

$$\boldsymbol{b}_c = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}^{\mathrm{T}}$$

$$\boldsymbol{c}_c = \begin{bmatrix} b_0 & b_1 & \cdots & b_m & 0 & \cdots & 0 \end{bmatrix}.$$

$B(s)$ and $A(s)$ are respectively the numerator and denominator of $P_c(s)$. n and $m\,(<n)$ denote the nominal plant order and number of the zeros, respectively. The discretized plant by zero-order hold with sampling time T_u is defined as

$$\boldsymbol{x}[k+1] = \boldsymbol{A}_s\boldsymbol{x}[k] + \boldsymbol{b}_s u[k], \;\; y[k] = \boldsymbol{c}_s\boldsymbol{x}[k] \quad (5)$$

$$\boldsymbol{A}_s = e^{\boldsymbol{A}_c T_u}, \;\; \boldsymbol{b}_s = \int_0^{T_u} e^{\boldsymbol{A}_c\tau}\boldsymbol{b}_c d\tau, \;\; \boldsymbol{c}_s = \boldsymbol{c}_c. \quad (6)$$

B. State trajectory $\boldsymbol{x}_d$ generation

According to Eq. (3), to track the reference position trajectory $r(t)$, the desired state trajectory $\boldsymbol{x}_d$ should satisfy

$$r(t) = \boldsymbol{c}_c\boldsymbol{x}_d(t). \quad (7)$$

Here, the reference trajectory vector is prepared as follows:

$$\boldsymbol{r}(t) = \begin{bmatrix} r_1(t) & r_2(t) & \cdots & r_n(t) \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 1 & \frac{d}{dt} & \cdots & \frac{d^{n-1}}{dt^{n-1}} \end{bmatrix}^{\mathrm{T}} r(t) \quad (8)$$

Next, the state trajectory $\boldsymbol{x}_d(t)$ is generated from $\boldsymbol{r}(t)$ by using controllable canonical form characteristics. To prevent the diversion of the state trajectory $\boldsymbol{r}(t)$, the stable-unstable decomposition and time axis reversal technique is used.

1) Stable-unstable decomposition: $B(s)^{-1}$ is decomposed into a stable part $F^{\mathrm{st}}(s)$ and an unstable part $F^{\mathrm{ust}}(s)$ as follows:

$$B(s)^{-1} = F^{\mathrm{st}}(s) + F^{\mathrm{ust}}(s) \quad (9)$$

$$f^{\mathrm{st}}(t) = \bar{\mathcal{L}}^{-1}\left[F^{\mathrm{st}}(s)\right], \bar{f}^{\mathrm{ust}}(t) = \bar{\mathcal{L}}^{-1}\left[(-1)^l F^{\mathrm{ust}}(-s)\right], \quad (10)$$

where l denotes the order of $F^{\mathrm{ust}}(s)$. Note that $F^{\mathrm{ust}}(-s)$ is stable.

2) Stable part state trajectory generation: The desired state trajectory $\boldsymbol{x}_d^{\mathrm{st}}(t)$ for the stable part is forwardly generated as follows.

$$\boldsymbol{x}_d^{\mathrm{st}}(t) = \begin{bmatrix} x_{1d}^{\mathrm{st}}(t) & x_{2d}^{\mathrm{st}}(t) & \cdots & x_{nd}^{\mathrm{st}}(t) \end{bmatrix}^{\mathrm{T}} = \int_{-\infty}^{t} f^{\mathrm{st}}(t-\tau)\boldsymbol{r}(\tau)d\tau \quad (11)$$

3) Unstable part state trajectory generation: The desired state trajectory $\boldsymbol{x}_d^{\mathrm{ust}}(t)$ for unstable part is generated by

$$\boldsymbol{x}_d^{\mathrm{ust}}(t) = \begin{bmatrix} x_{1d}^{\mathrm{ust}}(t) & x_{2d}^{\mathrm{ust}}(t) & \cdots & x_{nd}^{\mathrm{ust}}(t) \end{bmatrix}^{\mathrm{T}} = \int_{-\infty}^{\bar{t}} \bar{f}^{\mathrm{ust}}(\bar{t}-\bar{\tau})\boldsymbol{r}(-\bar{\tau})d\bar{\tau}\Big|_{\bar{t}=-t}. \quad (12)$$

$\boldsymbol{x}_d^{\mathrm{ust}}(t)$ is calculated as follows. First, a convolution of the time reversed reference position trajectory $\boldsymbol{r}(-\bar{t})$ and the stable signal $\bar{f}^{\mathrm{ust}}(\bar{t})$ is calculated. Next, the time axis is reversed. The mathematical proof is provided in [13][14].

4) State trajectory generation: The state trajectory $\boldsymbol{x}_d(t)$ is obtained by

$$\boldsymbol{x}_d(t) = \boldsymbol{x}_d^{\mathrm{st}}(t) + \boldsymbol{x}_d^{\mathrm{ust}}(t). \quad (13)$$

C. Feedforward output u_o generation from $\boldsymbol{x}_d$

The effect of unstable discretization zeros can be avoided using the multirate feedforward control [11]. Fig. 2 shows that there are three time periods T_y, T_u, and T_r denoting the periods for $y(t)$, $u(t)$, and $r(t)$, respectively. These periods are set as $T_r = nT_u = nT_y$.

The multirate system of (5) is given as

$$\boldsymbol{x}[i+1] = \boldsymbol{A}\boldsymbol{x}[i] + \boldsymbol{B}\boldsymbol{u}[i], \;\; y[i] = \boldsymbol{c}\boldsymbol{x}[i], \quad (14)$$

where

$$\begin{aligned} &\boldsymbol{A} = \boldsymbol{A}_s^n,\ \boldsymbol{B} = \begin{bmatrix} \boldsymbol{A}_s^{n-1}\boldsymbol{b}_s & \boldsymbol{A}_s^{n-2}\boldsymbol{b}_s & \cdots & \boldsymbol{A}_s\boldsymbol{b}_s & \boldsymbol{b}_s \end{bmatrix} \\ &\boldsymbol{c} = \boldsymbol{c}_s,\ \boldsymbol{x}[i] = \boldsymbol{x}(iT_r) \end{aligned} \tag{15}$$

by calculating the state transition from $t = iT_r = kT_u$ to $t = (i+1)T_r = (k+n)T_u$. Here, the input vector $\boldsymbol{u}[i]$ is defined in the lifting form

$$\begin{aligned} \boldsymbol{u}[i] &= \begin{bmatrix} u_1[i] & u_2[i] & \cdots & u_n[i] \end{bmatrix}^{\mathrm{T}} \\ &= \begin{bmatrix} u(kT_u) & u((k+1)T_u) & \cdots & u((k+n-1)T_u) \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{16}$$

According to Eq. (14), the feedforward output $\boldsymbol{u}_o[i]$ is obtained from the previewed state trajectory $\boldsymbol{x}_d[i+1]$ as follows:

$$\boldsymbol{u}_o[i] = \boldsymbol{B}^{-1}(\boldsymbol{I} - z^{-1}\boldsymbol{A})\boldsymbol{x}_d[i+1], \tag{17}$$

where $z = e^{sT_r}$.

III. Preactuation truncation problem

The method introduced in section II requires preactuation for $-\infty < t \leq 0$. In practice, infinite preactuation is impossible. In this section, a simple preactuation truncation is considered.

$$u_o[k] = \begin{cases} 0 & (k < -\frac{t_{pa}}{T_u}) \\ u_o[k] & (\text{otherwise}) \end{cases}, \tag{18}$$

where t_{pa} denotes the length of the preactuation time.

However, this method cannot achieve a perfect tracking for not only $-t_{pa} < t < 0$ but also $0 \leq t$ because the actual initial state variable $\boldsymbol{x}[0]$ does not match the initial state trajectory $\boldsymbol{x}_d[0]$. This result will be shown in section V-B and in Fig. 5 and Fig. 6.

IV. Finite preactuation method by state trajectory regeneration

This paper proposes a state trajectory regeneration method via redundant order polynomial in the negative time domain. Although this method abandons perfect tracking for $-t_{pa} < t < 0$, it guarantees perfect tracking for $0 \leq t$ because it can match $\boldsymbol{x}_d(t)$ and $\boldsymbol{x}(t)$. The tracking error of $-t_{pa} < t < 0$ is reduced by the regenerated state trajectory obtained using the optimized redundant order polynomial.

As shown in section II-A, the plant is realized through the controllable canonical form

$$\begin{aligned} \boldsymbol{x}_d^{\mathrm{reg}}(t) &= \begin{bmatrix} x_{1d}^{\mathrm{reg}}(t) & x_{2d}^{\mathrm{reg}}(t) & \cdots & x_{jd}^{\mathrm{reg}}(t) \end{bmatrix}^{\mathrm{T}}, \\ x_{jd}^{\mathrm{reg}}(t) &= \frac{d^{j-1}}{dt^{j-1}} x_{1d}^{\mathrm{reg}}(t) \end{aligned} \tag{19}$$

where $\boldsymbol{x}_d^{\mathrm{reg}}$ denotes the regenerated state trajectory.

The proposed method consists of the following steps.

A. Boundary condition calculation

The boundary condition for $-t_{pa} \leq t \leq 0$ is given as

$$\boldsymbol{x}_d^{\mathrm{reg}}(-t_{pa}) = \boldsymbol{O}, \quad \boldsymbol{x}_d^{\mathrm{reg}}(0) = \boldsymbol{x}_d(0), \tag{20}$$

where $\boldsymbol{x}_d^{\mathrm{ust}}(0)$ is calculated using Eq. (13). $\boldsymbol{x}(t) = \boldsymbol{O}\,(t < -t_{pa})$ is assumed without loss of generality.

B. State trajectory $x_{1d}^{\mathrm{reg}}(t)$ definition

$\boldsymbol{x}_d^{\mathrm{reg}}(t)$ $(-t_{pa} \leq t \leq 0)$ is defined using the n_tth order polynomial as

$$x_{1d}^{\mathrm{reg}}(t) = \boldsymbol{V}\boldsymbol{T}_v + \boldsymbol{W}\boldsymbol{T}_w \tag{21}$$

$$\begin{cases} \boldsymbol{T}_v &= \begin{bmatrix} 1 & t & t^2 & \cdots & t^{2n-1} \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{T}_w &= \begin{bmatrix} t^{2n} & t^{2n+1} & \cdots & t^{2n-1+n_r} \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{V} &= \begin{bmatrix} v_0 & v_1 & \cdots & v_{2n-1} \end{bmatrix} \\ \boldsymbol{W} &= \begin{bmatrix} w_1 & w_2 & \cdots & w_{n_r} \end{bmatrix} \end{cases}, \tag{22}$$

where n and n_r $(0 \leq n_r \in \mathbb{Z})$ denote the orders of the plant and generated trajectory redundancy, respectively. The order of the trajectory n_t is expressed using $n_t = 2n-1+n_r$ because the number of the boundary condition is $2n$ considering the plant order is n.

C. Evaluation function definition

By simultaneously solving the equations (20) and (21), $x_{1d}(t)$ can be expressed through only $\boldsymbol{W}$ and t. Eq. (4) shows that the desired output $y_d(t, \boldsymbol{W})$ is calculated as

$$\begin{aligned} y_d(t, \boldsymbol{W}) &= \boldsymbol{c}_c \boldsymbol{x}_d^{\mathrm{reg}}(t, \boldsymbol{W}) \\ &= \boldsymbol{c}_c \begin{bmatrix} x_{1d}^{\mathrm{reg}}(t, \boldsymbol{W}) \\ \frac{d}{dt} x_{1d}^{\mathrm{reg}}(t, \boldsymbol{W}) \\ \frac{d^2}{dt^2} x_{1d}^{\mathrm{reg}}(t, \boldsymbol{W}) \\ \vdots \\ \frac{d^{n-1}}{dt^{n-1}} x_{1d}^{\mathrm{reg}}(t, \boldsymbol{W}) \end{bmatrix} \end{aligned} \tag{23}$$

Thus, the tracking error $e_d(t, \boldsymbol{W})$ is calculated as

$$e_d(t, \boldsymbol{W}) = r(t) - y_d(t, \boldsymbol{W}) \tag{24}$$

In this paper, Eq. (25) is defined as an evaluation function:

$$g(\boldsymbol{W}) = \int_{-t_{pa}}^{0} \left[(e_d(t, \boldsymbol{W}))^2 + k_{\mathrm{acc}} (\ddot{y}_d(t, \boldsymbol{W}))^2 \right] dt, \tag{25}$$

where k_{acc} (> 0) denotes a weighting coefficient to smoothen $y_d(t, \boldsymbol{W})$.

D. Unconstrained optimization

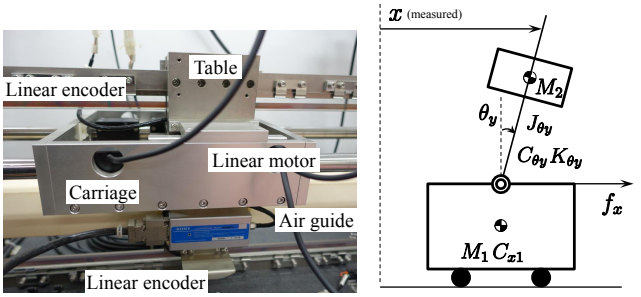
The optimal coefficient is obtained by solving the following unconstrained optimization problem:

$$\nabla g(\boldsymbol{W}) = 0,\ \nabla^2 g(\boldsymbol{W}) > 0 \tag{26}$$

E. Negative time state trajectory calculation

$x_{1d}^{\mathrm{reg}}(t)$ is expressed by the $n_t = 2n - 1 + n_r$ th polynomial with optimized coefficients. $\boldsymbol{x}_d^{\mathrm{reg}}(t)$ is calculated using Eq. (27) by considering Eq. (19).

$$\boldsymbol{x}_d^{\mathrm{reg}}(t) = \begin{bmatrix} x_{1d}^{\mathrm{reg}}(t) \\ x_{2d}^{\mathrm{reg}}(t) \\ x_{3d}^{\mathrm{reg}}(t) \\ \vdots \\ x_{(n-1)d}^{\mathrm{reg}}(t) \end{bmatrix} = \begin{bmatrix} x_{1d}^{\mathrm{reg}}(t) \\ \frac{d}{dt} x_{1d}^{\mathrm{reg}}(t) \\ \frac{d^2}{dt^2} x_{1d}^{\mathrm{reg}}(t) \\ \vdots \\ \frac{d^{n-1}}{dt^{n-1}} x_{1d}^{\mathrm{reg}}(t) \end{bmatrix} \tag{27}$$



(a) High-precision stage. (b) Model of (a).

Fig. 3. Experimental high-precision stage and its model for simulation[15][16][17].

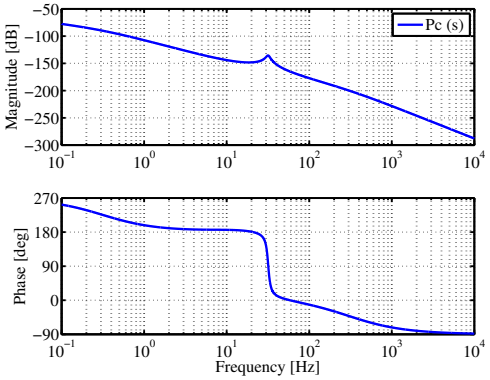


Fig. 4. Bode diagram of $P_c(s)$.

F. State trajectory generation

As abovementioned the state trajectory is obtained by

$$\boldsymbol{x}_d(t) = \begin{cases} \boldsymbol{O} & (t < -t_{pa}) \\ \boldsymbol{x}_d^{\mathrm{reg}}(t) & (-t_{pa} \le t \le 0) \\ \boldsymbol{x}_d^{\mathrm{st}}(t) + \boldsymbol{x}_d^{\mathrm{ust}}(t) & (0 < t) \end{cases}. \tag{28}$$

V. Simulation results

A. Simulation condition

This section describes the simulations performed using the model illustrated in Fig. 3(b). This model assumes a high-precision stage shown in Fig. 3(a). Here, the continuous time domain transfer function is defined as

$$P_c(s) = \frac{-(s-140)(s+100)}{s(s+2000)(s+2)(s^2+20s+40000)} \tag{29}$$

assuming that the transfer function from the current reference of the x-axis actuator which generates force f_x to the measured stage position x. τ is defined as the time constant of the unstable zero as

$$\tau = \frac{1}{140} \simeq 7.2\ [\mathrm{ms}]. \tag{30}$$

The discretized transfer function of Eq. (29) with zero-order hold is obtained as

$$P_s[z_s] = \frac{K(z_s+3.547)(z_s-1.014)(z_s-0.9900)(z_s+0.2543)}{(z_s-1)(z_s-0.9998)(z_s-0.8187)(z_s^2-1.998z_s+0.998)}, \tag{31}$$

with the sampling period $T_u = 100\ [\mu\mathrm{s}]$. z_s denotes e^{sT_u}. Fig. 4 shows the Bode diagram of $P_c(s)$. In the continuous time domain, $P_c(s)$ has a stable and an unstable zero. In the discrete time domain, $P_s[z_s]$ has intrinsic zeros at $z_s = +0.99$ and $z_s = +1.014$, and discretized zeros at $z_s = -3.547$ and $z_s = -0.2543$. Here, $P_s[z_s]$ has an unstable intrinsic zero and an unstable discretized zero.

Fig. 5(a) shows the step target trajectory $r(t)$ designed using 9th-order polynomial during step motion. The step time is set at 0.02 [s]. The weighting coefficient k_{acc} is set at 1.0×10^{-11}.

B. Preactuation truncation problem

Fig. 5, 6, and 10(a) show the simulation results of preactuation truncation for $t_{pa} = 70\tau$, 5τ, 4τ, and 3τ. When $t_{pa} = 70\tau$, the output $y(t)$ tracks the reference trajectory $r(t)$ because $t_{pa} = 70\tau \gg \tau$. However, for $t_{pa} = 5\tau$, 4τ, and 3τ cases, the output $y(t)$ cannot track the reference for not only $-t_{pa} < t < 0$ but also $0 \le t$. This is because of the mismatch between the initial value of the state trajectory $\boldsymbol{x}(0)$ and $\boldsymbol{x}_d(0)$, as shown in Fig. 6 and 10(a).

C. Simulation results of the proposed method

Fig. 7–9, 10(b), and 11 show simulation results. Fig. 7(b), 8(b), and 9(b) show that perfect tracking is achieved in the positive time domain in contrast to Fig. 5(b). Fig. 10(b) shows proposed method can match the state variable to the desired initial value at $t = 0$. Fig. 11 shows peak-to-peak tracking error for $-t_{pa} < t < 0$ is reduced by the increasing order of the regenerated state trajectory.

These figures indicate that the length of the preactuation t_{pa} has a considerable impact on the reduction of tracking error $e(t)$ and peak of control input $u(t)$ during preactuation.

D. Comparison with approximated inverse methods

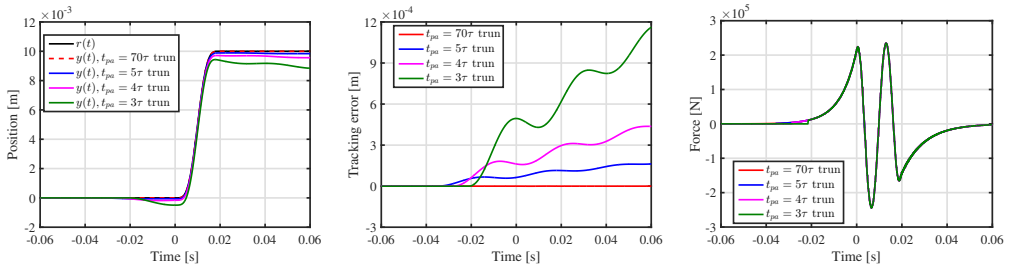
Simulation results of NPZI method [4], ZPETC method [5], and ZMETC method [6] are shown in Fig. 12. These three methods are designed by sampling period T_u. The ZPETC method uses a preview to achieve the zero-phase-error characteristics.

These methods create an undershoot and/or overshoot to compensate for the unstable intrinsic zero ($z_s = +1.014$) and unstable discretization zero ($z_s = -3.547$). Therefore, without preactuation, trade-offs exist between the undershoot and/or overshoot amplitude and the settling time.

VI. Conclusion

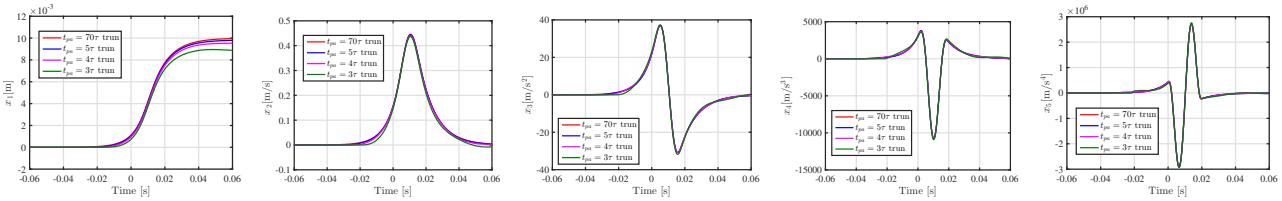
In the discretized domain, a plant has two types of zeros: 1) intrinsic zeros, which have counterparts in the continuous time domain, and 2) discretization zeros generated through discretization. The feedforward control is thus difficult because of unstable pole(s) of its inversion system. According to [10], the unstable intrinsic and discretization zeros can be managed separately through the combination of multirate feedforward and state trajectory generation with time axis reversal. However, this method needs infinite time preactuation.

This paper proposes a finite time preactuation method based on state trajectory regeneration using a redundant



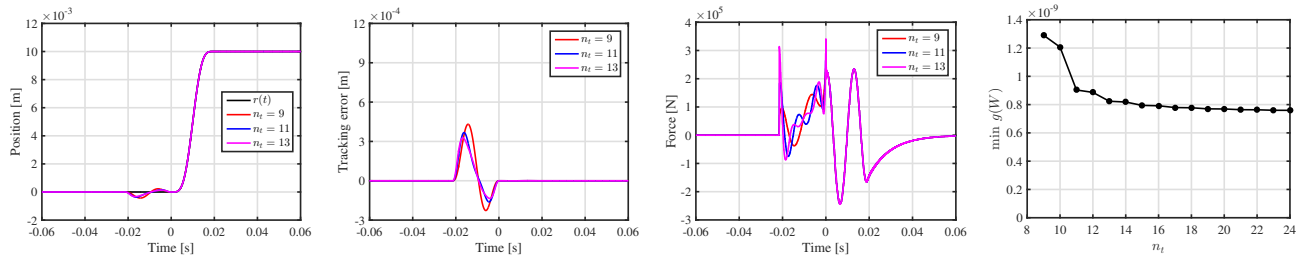
(a) $r(t), y(t)$. (b) $e(t)$. (c) $u(t)$.

Fig. 5. Preactuation truncation case. Output $y(t)$ cannot track the reference $r(t)$.



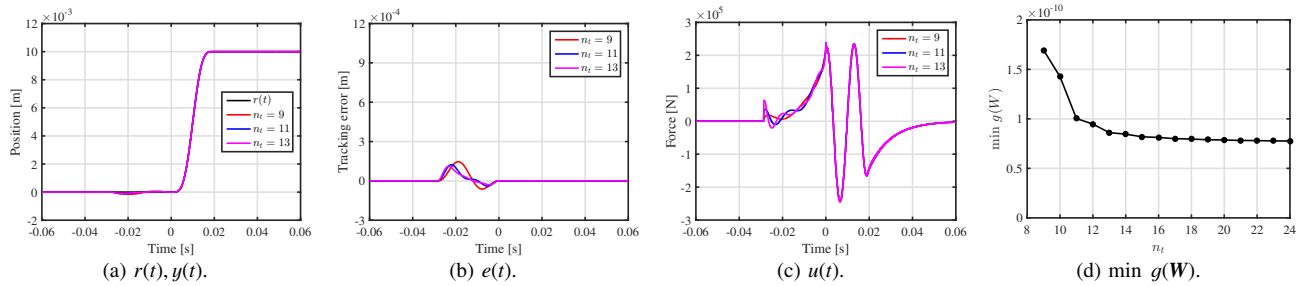
(a) $x_1(t)$. (b) $x_2(t)$. (c) $x_3(t)$. (d) $x_4(t)$. (e) $x_5(t)$.

Fig. 6. State trajectory output $\boldsymbol{x}(t)$ of the preactuation truncation case. Mismatches exist between $t_{pa} = 70\tau$ case and the other cases for not only $-t_{pa} < t < 0$ but also $0 \le t$. Therefore, perfect tracking in $0 < t$ is not achievable, as shown in Fig. 5. Here, $t_{pa} = 70\tau \gg \tau$ is assumed.



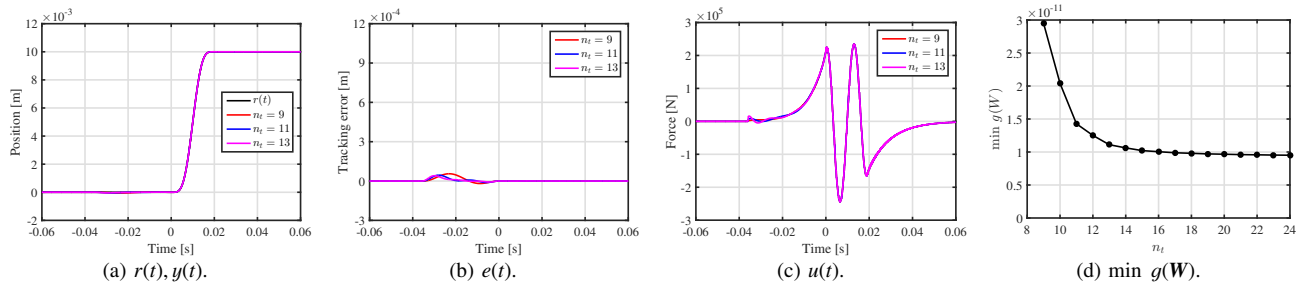
(a) $r(t), y(t)$. (b) $e(t)$. (c) $u(t)$. (d) min $g(\boldsymbol{W})$.

Fig. 7. Finite preactuation method with redundant order polynomial trajectory ($t_{pa} = 3\tau$).



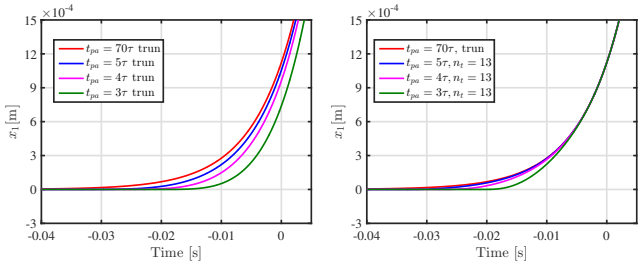
(a) $r(t), y(t)$. (b) $e(t)$. (c) $u(t)$. (d) min $g(\boldsymbol{W})$.

Fig. 8. Finite preactuation method with redundant order polynomial trajectory ($t_{pa} = 4\tau$).



(a) $r(t), y(t)$. (b) $e(t)$. (c) $u(t)$. (d) min $g(\boldsymbol{W})$.

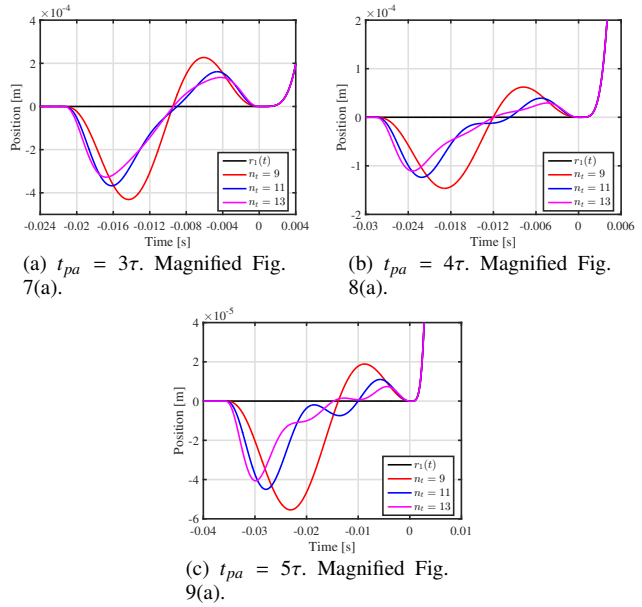
Fig. 9. Finite preactuation method with redundant order polynomial trajectory ($t_{pa} = 5\tau$).



(a) Preactuation truncation case. Magnified Fig. 6(a).

(b) State trajectory regeneration case.

Fig. 10. State trajectory output $x_1(t)$. State trajectory regeneration case can match the state variable $\boldsymbol{x}(t)$ to the desired value at $t = 0$.



(a) $t_{pa} = 3\tau$. Magnified Fig. 7(a).

(b) $t_{pa} = 4\tau$. Magnified Fig. 8(a).

(c) $t_{pa} = 5\tau$. Magnified Fig. 9(a).

Fig. 11. Magnified figures of $r(t), y(t)$. Higher order trajectory and longer t_{pa} reduces peak-to-peak tracking error for $-t_{pa} < t < 0$.

polynomial. The proposed method can achieve perfect tracking in the positive time domain. The tracking error in the negative time domain is reduced by optimizing the coefficient of the redundant order polynomial. Owing to the multirate feedforward and controllable canonical form realization, the proposed method can formulate the state trajectory during preactuation as a polynomial. The polynomial defined trajectory enables us to apply the optimization algorithm to reduce tracking error. The effectiveness of the proposed method is demonstrated through simulations.

Acknowledgment

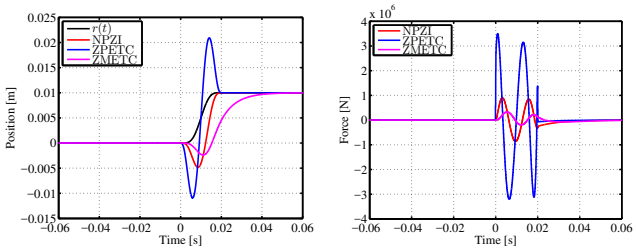
This work is supported by JSPS KAKENHI Grant Number 15J08488.

References

[1] T. Hagiwara, T. Yuasa, and M. Araki, "Stability of the limiting zeros of sampled-data systems with zero-and first-order holds," *International Journal of Control*, vol. 58, no. 6, pp. 1325–1346, 1993.

[2] T. Hagiwara, "Analytic study on the intrinsic zeros of sampled-data systems," *IEEE Transactions on Automatic Control*, vol. 41, no. 2, pp. 261–263, 1996.

[3] K. Åström, P. Hagander, and J. Sternby, "Zeros of sampled systems," *Automatica*, vol. 20, no. 1, pp. 31–38, 1984.



(a) $r(t), y(t)$.

(b) $u_o(t)$.

Fig. 12. Feedforward simulation results. The reference position trajectory $r(t)$, output position $y(t)$, and feedforward force u of the NPZI, ZPETC, and ZMETC methods.

[4] J. A. Butterworth, L. Y. Pao, and D. Y. Abramovitch, "Analysis and comparison of three discrete-time feedforward model-inverse control techniques for nonminimum-phase systems," *Mechatronics*, vol. 22, no. 5, pp. 577–587, 2012.

[5] M. Tomizuka, "Zero phase error tracking algorithm for digital control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, pp. 65–68, 1987.

[6] J. Wen and B. Potsaid, "An experimental study of a high performance motion control system," in *American Control Conference*, vol. 6, pp. 5158–5163, 2004.

[7] B. Rigney, L. Y. Pao, and D. Lawrence, "Nonminimum phase dynamic inversion for settle time applications," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 989–1005, 2009.

[8] L. Marconi, G. Marro, and C. Melchiorri, "A solution technique for almost perfect tracking of non-minimum-phase, discrete-time linear systems," *International Journal of Control*, vol. 74, no. 5, pp. 496–506, 2001.

[9] T. Shiraishi and H. Fujimoto, "A Reference Trajectory Generation for System with Unstable Zeros Considering Negative-time Domain Analysis," in *IEEJ International Workshop on Sensing, Actuation, and Motion Control*, 2015.

[10] W. Ohnishi and H. Fujimoto, "Multirate Feedforward Control with State Trajectory Generation based on Time Axis Reversal for Plant with Continuous Time Unstable Zeros," in *IEEE International Conference on Advanced Intelligent Mechatronics*, pp. 689–694, 2016.

[11] H. Fujimoto, Y. Hori, and A. Kawamura, "Perfect tracking control based on multirate feedforward control with generalized sampling periods," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 3, pp. 636–644, 2001.

[12] T. Chen and B. A. Francis, *Optimal Sampled-Data Control Systems*. 1996.

[13] T. Sogo, "Calculation of the Non-causal Solution for the Model-matching Problem and Its Application to Preview Feedforward Control," *Transactions of the Society of Instrument and Control Engineers*, vol. 42, no. 1, pp. 40–46, 2006 (in Japanese).

[14] T. Sogo, "An algebraic expression of stable inversion for nonminimum phase systems and its applications," in *the 17th IFAC world congress*, vol. 17, pp. 2820–2825, 2008.

[15] A. Hara, K. Saiki, K. Sakata, and H. Fujimoto, "Basic examination on simultaneous optimization of mechanism and control for high precision single axis stage and experimental verification," in *34th Annual Conference of IEEE Industrial Electronics*, pp. 2509–2514, 2008.

[16] H. Fujimoto, K. Sakata, and K. Saiki, "Application of Perfect Tracking Control to Large-Scale High-Precision Stage," in *5th IFAC Symposium on Mechatronic Systems*, pp. 188–193, 2010.

[17] T. Shiraishi and H. Fujimoto, "Trajectory tracking control method based on zero-phase minimum-phase factorization for nonminimum-phase continuous-time system," in *International Workshop on Advanced Motion Control*, pp. 1–6, 2012.