A Model Reference Observer for Time-Delay Control and Its Application to Robot Trajectory Control

Pyung H. Chang, Member, IEEE, and Jeong W. Lee

Abstract—This paper addresses the estimation problem of the states and their derivatives for time-delay control (TDC), a robust control technique for nonlinear systems. To this end, an observer design method is presented. In addition, the sufficient conditions are discussed and implementation issues are addressed. Finally, experiments were undertaken on a SCARA-type robot subject to substantial inertia variations and external disturbances. The results showed that the controller/observer performs quite robustly under inertia variations and disturbances and is much less sensitive to sensor noise than the controller using numerical differentiations.

I. INTRODUCTION

It is sometimes necessary to design controllers with certain performances (such as accuracy, stability, speed, etc.) for nonlinear plants in the presence of significant plant uncertainties. The plant uncertainties, in general, include external disturbances, unpredictable parameter variations, and unmodeled plant nonlinear dynamics. To name a few among such plants, there are robot manipulators picking and placing payloads, pneumatic actuators moving inertia-changing workpieces, and flight control systems under large wind resistances.

Among the few techniques to address the control problem, time-delay control (TDC), together with sliding-mode control [1], [2], has attracted attention as an effective robust, nonlinear control algorithm [3]. To explain briefly, TDC does not require any real-time computation of nonlinear dynamics and the uncertainties, nor does it require such parameter estimations as in adaptive control. Instead, it adopts a very efficient scheme to estimate the nonlinear dynamics and the uncertainties: to approximate them with the time-delayed values of control inputs and derivatives of state variables at the previous time step. The estimation thus made is used to cancel the nonlinear dynamics and uncertainties for which the desired dynamics (reference model) are substituted. As a result, TDC shows quite robust responses under the aforementioned uncertainties, yet is computationally very efficient. These merits have been clearly demonstrated in the successful applications of TDC to a robot [4], [5], a magnetic bearing [6], and an electrohydraulic servo-motor [7].

To apply TDC to a plant, it is necessary to be able to measure all of the state variables and their derivatives. Unfortunately, this is not always the case in practice. In many plants, even state variables are not always available, not to mention their derivatives. Even if all the state variables are measurable, additional sensors or numerical differentiators for their derivatives are needed at the cost of further disadvantages—the use of derivative sensors makes the overall system more complex and expensive and the use of differentiators makes the system more sensitive to measurement noise. Hence, the measurability requirement presents a serious limitation on the implementations of TDC to real plants.

Regarding numerical differentiators, on the one hand, this can be alleviated by using the convolution method put forward by Youcef-Toumi and Shortlidge [5] which requires one to have all the state variables available. As to the measurability requirement, on the other hand, the input–output linearization method of Youcef-Toumi and Wu [8] enables one to apply TDC only with output variables and their rth derivatives, where r denotes the relative degree of system. In this case, one needs a way to estimate the rth derivatives which may probably have to be done by numerical differentiation.

To avoid noise problems, by the way, a low-pass filter may be used in conjunction with the differentiation. From our experience, using a filter often turns out to be effective. Nevertheless, low-pass-filtering is sometimes observed to fail in reducing noise, no matter how well tuned it may be. Instead, it induces even larger overshoots—especially when nonlinearity effects are substantial.

In the context described so far, the problem may be stated as: How can we apply TDC to systems where only a part of state variables are available, yet alleviate the noise problems due to numerical differentiations? As a solution to this problem, this paper proposes an observer design method that can stably reconstruct state variables and their derivatives, thereby trying to contribute to improving TDC.

In the following section, the control problem is defined and the TDC algorithm is reviewed. Section III presents an observer design method for TDC. Section IV deals with the issues involved in the implementation of the proposed observer such as the numerical scheme and the computational load. In Section V, the effectiveness of the proposed observer is evaluated through experiments on the trajectory control of a robot driven with high speed which is subject to a substantial inertia variation and external disturbances. Finally, in Section VI the results are summarized and conclusions are drawn.

II. TDC LAW

To provide the basis for the observer design method, the TDC algorithms in two different formulations will be briefly
reviewed; more detailed expositions will be found in [3] and [4].

A. Two Different Formulations

There are two formulations of problems for TDC techniques: the one is proposed by Youcef-Toumi [3], and the other proposed by Hsia [4]. Although based on similar concepts, they are different in the following two respects:

- In the former formulation, the plant is a generic nonlinear system; in the latter formulation, it is a specific system (e.g., robot dynamics).
- Furthermore, in the former, the desired states to track are specified with the states of linear time-invariant model driven by command input; whereas in the latter the desired states are specified with the joint trajectories determined a priori.

Since these are major formulations for TDC, it appears worthwhile to investigate the relationship between them. To this end, let us first examine each of them more in detail, and then seek the relationship.

1) Formulation Based on Youcef-Toumi: In the Youcef-Toumi formulation, the nonlinear plant in question is described as follows:

$$\dot{x} = f(x, t) + B(x, t)u + d(t)$$

where $x \in \mathbb{R}^n$ denotes the state vector, $u \in \mathbb{R}^p$ the control input, $f(x, t)$ the plant dynamics, which is unknown, $d(t)$ the unknown disturbance, and $B(x, t)$ the control distribution matrix, the ranges (instead of exact values) of which are known.

Now let the desired performances be specified with the response of a stable linear time-invariant reference model as

$$\dot{x}_m = A_m x_m + B_m r$$

where $x_m \in \mathbb{R}^n$ denotes the state vector of the reference model, $r \in \mathbb{R}^p$ the command vector, $A_m$ the system matrix, and $B_m$ the command distribution matrix. Then the control objective is to find a control input $u$ that makes the states of the plant asymptotically track the response of the reference model, (2). In other words, the tracking error $e_m \triangleq x_m - x$ is required to satisfy the following error dynamics:

$$\dot{e}_m = A_m e_m.$$  \hspace{1cm} (3)

2) Formulation Based on Hsia: In Hsia’s formulation, the plants to be controlled are robot dynamics as follows:

$$M(\theta)\dot{\theta} + \dot{V}(\theta, \dot{\theta}) + G(\theta) + D(\theta, \dot{\theta}) = \tau$$

where $\theta \in \mathbb{R}^p$ denotes the joint vector with $p$ representing the degrees of freedom of the robot, $M(\theta) \in \mathbb{R}^{p \times p}$ the inertia matrix, $V(\theta, \dot{\theta}) \in \mathbb{R}^p$ a vector function due to the Coriolis force and centrifugal force, $G(\theta) \in \mathbb{R}^p$ a vector function due to the gravitational force, $D(\theta, \dot{\theta}) \in \mathbb{R}^p$ a term accounting for frictions and unmodeled nonlinearities, and $\tau \in \mathbb{R}^p$ the joint torque vector.

Let the desired states to track be the desired joint trajectories, $\theta_d, \dot{\theta}_d, \ddot{\theta}_d$. Then the control objective is to find a control input $\tau$, such that the tracking error, $e_d \triangleq \theta_d - \theta$, satisfies the following error dynamics:

$$\dot{e}_d + K_v e_d + K_p e_d = 0$$  \hspace{1cm} (5)

where $K_v$ and $K_p$ are usually set to be positive definite diagonal matrices.

3) The Relationship Between the Two Formulations: The robot dynamics equation is expressed in the form of a state equation, with the state vector and input vector $x = \begin{pmatrix} \theta \\ \theta \end{pmatrix}$ and $u = \tau$, respectively, it becomes clear from the following that the robot dynamics equation may be regarded as a particular form derived from (1):

$$\dot{x} = \begin{pmatrix} 0_p \\ I_p \end{pmatrix} x - \begin{pmatrix} -M(x)^{-1} [V(x) + G(x) + D(x)] \\ 0_p \end{pmatrix} \begin{pmatrix} \theta \\ \theta \end{pmatrix}$$

where $I_p$ and $0_p$ denote $p$-dimensional identity matrix and zero matrix, respectively.

The desired trajectory $x_d$ can be related to the reference model (2), by observing that the error dynamics (5) may be expressed as

$$\dot{x}_d - \dot{\theta}_d = A_m \dot{x}_d - \dot{x}_d.$$

Subtracting (7) from (3) leads to

$$\dot{x}_d - A_m x_d = \dot{x}_m - A_m x_m$$  \hspace{1cm} (8)

and

$$B_m r = \dot{x}_d - A_m x_d.$$  \hspace{1cm} (9)

Let $r_d$ denote a command vector $r$ that satisfies (9). Then one may interpret (9) as follows. Since $x_d$ is the response of the reference model (2) driven by $r_d$, knowing the desired trajectory $x_d$ is equivalent to having the reference model (2) and a particular command $r_d$. This interpretation conforms to the intuition that the performance specification with $x_d$ appears somewhat more specific than that with the reference model, or $x_m$.

Now that the relationship between the two formulations has been clarified, the control law derived from one formulation can be used to obtain the control law for another. Since it appears natural to begin with the more general formulation, the control law for the former formulation will be obtained first and then it will be applied to obtain the control law for the latter.

B. Control Law

To find the control law, rearrange (1) into the following form:

$$\dot{x} = f(x, t) + Bu$$

\hspace{1cm} (10)

\[1\] In the original formulation, $\dot{e}_m = (A_m + K)e_m$ was used; but without losing the essence of the idea, (3) may be also used.
where $\dot{B}$ denotes a constant matrix representing the known range of $B(x,t)$, and $\hat{f}(x,t)$ the unknown part consisting of all the uncertainties in the plant and disturbances, which are expressed as

$$\dot{f}(x,t) = f(x,t) + [B(x,t) - \dot{B}]u + d(t). \tag{11}$$

Subtracting (10) from (2) leads to

$$\dot{e}_m = A_m e_m + [-\dot{f}(x,t) + A_m x + B_m r - \dot{B} u]. \tag{12}$$

When the number of control inputs is equal to the number of the states, the following input:

$$u = \dot{B}^{-1}[-\dot{f}(x,t) + A_m x + B_m r] \tag{13}$$

eliminates all the terms within the bracket in (12), thereby achieving the control objective. The number of control inputs, however, is usually smaller than the number of the states, in which case only the best approximation in the least-square sense is available as the following:

$$u = \dot{B}^{+}[-\dot{f}(x,t) + A_m x + B_m r] \tag{14}$$

where $\dot{B}^{+}$ denotes a pseudoinverse of $\dot{B}$, namely $\dot{B}^{+} \equiv (\dot{B}^T \dot{B})^{-1} \dot{B}^T$. With this input, the sum of the terms within the bracket of (12) becomes

$$(I - \dot{B}\dot{B}^{+})[-\dot{f}(x,t) + A_m x + B_m r]. \tag{15}$$

The conditions on which this sum always equals zero and hence the control objective is achieved are discussed in [3].

In the robot dynamics case, one can easily derive from (6) that

$$\dot{B}^{+} = [0_2 \bar{M}] \tag{16}$$

and (15) becomes identically zero. Incidentally, note that $\bar{M}(\theta)$ has been approximated by $\bar{M}$, which is a constant diagonal matrix determined on the basis of overall stability analysis [4].

When implementing the algorithm in (14), a real-time estimation for the total effect of uncertainties is required for $\dot{f}(x,t)$, the total effect of plant uncertainties. Note that it is this estimation that requires a significant amount of computation in several nonlinear control methods such as the computed torque method [9] and the sliding mode control [2]. In contrast, TDC adopts a particularly efficient estimation method based on the following ideas: First, make use of the fact that $\dot{f}(x,t)$ may be assumed mostly as a continuous function, from which it follows that, for a sufficiently small $L$

$$\dot{f}(x,t) \cong \dot{f}(x,t-L). \tag{17}$$

Second, use (10) together with (17). Then, one obtains the following estimation for the total effect of uncertainties:

$$\dot{f}(x,t) = \dot{x} - \dot{B} u \cong \dot{x}(t-L) - \dot{B} u(t-L). \tag{18}$$

Substituting this approximate estimation into (14) leads to the following TDC control law:

$$u = \dot{B}^{+}[-\dot{x}(t-L) + \dot{B} u(t-L) + A_m x + B_m r] \tag{19}$$

where the initial values for $\dot{x}(t-L)$ and $u(t-L)$ at $t = 0$ are set to be $\dot{x}(0)$ and $u(0)$, respectively.

Note that the control law is intended primarily for digital control, where the value of $L$, the time-delay, is normally set to be the sampling period, so that the continuity assumption of $\dot{f}(x,t)$ may be valid.\(^2\) Hence, for a system where $\dot{f}(x,t)$ changes very rapidly, $L$ need to be so much smaller, so that the rapid change may still remain manageable to the controller.

Finally, the control law for the robot dynamics is easily derived: By substituting (16) and (9) into (19), one can obtain the same control law as in [4]

$$\tau(t) = \dot{M} \dot{\theta}(t) + K_v \dot{e}_d(t) + K_p e_d(t) + \tau(t + L) - \dot{M} \dot{\theta}(t - L). \tag{20}$$

It is noteworthy that the control law, not requiring the real-time computation of robot dynamics, is quite simple and computationally efficient.

In another viewpoint, the ever-increasing computational power might ultimately be able to render the efficiency issue to a trivial one. Then it would be more helpful to include the real-time computation of robot dynamics. In the meantime, however, computational efficiency based on the simplicity is still something to be desired, especially so in the industry, where the real-time computation of dynamics could be expensive and intimidating.

### III. OBSERVER DESIGN

As is clearly shown in (19), the TDC control law requires the estimation of states and their derivatives. In practice, this requirement sets nontrivial limitations on the application of TDC to real plants. As a solution to this problem, use of an observer may be considered. Several approaches exist for designing observer structures for nonlinear plants [10–15]. Most of these approaches estimate the states by using nonlinear plant model (including robot dynamics [12–14]), which depending on its complexity could require an intensive amount of computation. In this section a different observer structure is proposed that does not need such computation. In addition, the conditions for the closed-loop stability are discussed.

#### A. Derivation of Observer Equation

Consider the nonlinear plant in (1) again, and the output $y \in \mathbb{R}^m$ that is linearly related to the state vector

$$\dot{x} = f(x,t) + B(x,t)u + d(t)$$

$$y = C x \tag{21}$$

where $C$ is the output distribution matrix.

The idea behind the observer design may be described as follows. For a certain class of plants, TDC effectively cancels the uncertainties, and thereby enables the plant dynamics to immediately follow the reference model. This observation

\(^2\)Note that the total uncertainty $\dot{f}(x,t)$ includes the system nonlinearity $f(x,t)$ and the disturbance $d(t)$; thus the selection of $L$ needs to be based on the higher bandwidth between those of $f(x,t)$ and $d(t)$.\]
leads to a useful strategy for an observer design: to reconstruct the states, use the reference model (linear and certain)—instead of the plant model (nonlinear and uncertain).

Thus in system (21), the states are reconstructed by using the following linear observer

\[
\dot{z} = A_m z + B_m r + F(\theta - \theta_d)
\]

where \( z \in \mathbb{R}^n \) denotes an observer state vector, \( F \in \mathbb{R}^{n \times m} \) a constant observer gain matrix, and \( \theta \) an observer output vector.

For robot control, as often is the case, the measurable states are assumed to be the joint vector, \( \theta \), in which case, \( C = [I, 0] \). Substituting (7) and (9) into (22) yields the following observer for robot dynamics:

\[
\dot{z}_1 = z_2 + F_1 (z_1 - \theta)
\]
\[
\dot{z}_2 = \dot{\theta}_d - K_c (z_2 - \theta_d) - K_p (z_1 - \theta_d)
\]
\[
+ F_2 (z_1 - \theta)
\]

where \( z = [z_1, z_2]^T \) and \( F = [F_1, F_2] \), with \( F_1, F_2 \in \mathbb{R}^{p \times p} \).

It is noteworthy that (23) coincides with the smooth observer proposed in [15]. A close inspection reveals that the essential idea behind the two observers is quite similar: to estimate the states, the desired trajectory is used, instead of the nonlinear plant model. Moreover, the two observers share an important consideration in their design, taking into account their intended controllers and their overall stability.

Yet they are different in that the smooth observer is intended specifically for robot control (the computed torque method), whereas our observer for the control of more general dynamic systems. Hence, designing the smooth observer together with the controller necessitates model properties inherent to robot dynamics, whereas ours do not. Furthermore, recall that the desired trajectory used in our observer was derived as a particular case of a more general specification, (2).

When the observer (22) (together with TDC) is connected to the plant, the control input \( u \) is to be obtained, by using the reconstructed states \( z \), instead of the states \( x \). In addition, the uncertainties at time \( t \) are to be estimated with the reconstructed state \( z \) at \( t - L \). Thus, the control input \( u \) and the estimation using time delay are determined as

\[
u = B^+ [-\dot{z}(t - L) + \dot{B}u(t - L) + A_m \dot{z} + B_m r].
\]

For robot control, (20) is used as before; when the proposed observer is used, however, \( e_d(t) \) is substituted with \( e_d = \theta_d - z_1, \dot{e}_d = \dot{\theta}_d - z_2 \), and \( \theta(t - L) \) with \( \dot{z}_2(t - L) \), where \( z_1 \) and \( z_2 \) are obtained by using (23). The overall system can be illustrated with the block diagram in Fig. 1.

As to the stability of resulting system consisting of the plant, TDC, and the proposed observer, a theorem based on the results of Youcef-Toumi and Reddy [16] has been stated with its proof in [17]. This theorem provides a sufficient condition for the stability of the overall system. Thus if the proposed observer is designed so that the observer gain matrix \( F \) and time delay \( L \) meet this condition, then the resulting system is made stable.

IV. IMPLEMENTATION ISSUES

To implement the proposed observer in a real-time controller, we need to select a computation scheme, and to evaluate its required computation amount. Hence, issues regarding the computation schemes and computation effort are discussed.

A. Computation Scheme

To solve for \( z \) in (22) requires a numerical integration schemes, such as the Euler method (for rather a rough, yet computationally efficient estimation) or the Runge-Kutta method (for rather an accurate, yet computationally intensive estimation). Nevertheless, using the fact that (22) is a linear time-invariant system yields an efficient solution without sacrificing the accuracy. As is well known from the linear control theory, the solution for (22) is given as

\[
\Phi(t, t_0) = e^{(A_m + FC)(t-t_0)}
\]

\[
\Gamma_1(t, t_0) = \int_{t_0}^{t} e^{(A_m + FC)\tau} d\tau B_m
\]

\[
\Gamma_2(t, t_0) = \int_{t_0}^{t} e^{(A_m + FC)\tau} d\tau F
\]

with \( t_0 \) the initial time.

Since the control law is intended primarily for digital controllers, a discrete form of (25) is required. Assuming a fixed sampling period, \( T_s \), for the observer, the following equation is obtained:

\[
z(k + 1) = \Phi z(k) + \Gamma_1 r(k) + \Gamma_2 y(k)
\]

where

\[
\Phi = e^{(A_m + FC)T_s}
\]

\[
\Gamma_1 = \int_{0}^{T_s} e^{(A_m + FC)\tau} d\tau B_m
\]

\[
\Gamma_2 = \int_{0}^{T_s} e^{(A_m + FC)\tau} d\tau F
\]

This sampling time—term it the observer sampling time—is several times smaller than that of TDC and \( L \).
Note that the matrices \( \Phi, \Gamma_1, \) and \( \Gamma_2 \) are constant matrices that can be determined on off-line basis, once the observer gain \( F \) and sampling period \( T_s \) are given. It was observed through simulations and experiments that the estimation using (26) achieves approximately the same accuracy as that of the Runge-Kutta method at a computational cost similar to that of the Euler method.

### B. Computation Amount

Based on the aforementioned scheme, the computation amount can be readily estimated. As before, the general case in (22) is dealt with first, and then the particular case in (23) next.

1) The General Case: The dimensions of the matrices in (26) are again \( \Phi \in \mathbb{R}^{n \times n}, \Gamma_1 \in \mathbb{R}^{n \times r} \), and \( \Gamma_2 \in \mathbb{R}^{n \times m} \). Hence, to evaluate \( z(k+1) \) in (22), one needs to execute \( n^2 + nr + nm - n \) additions and \( n^2 + nr + nm \) multiplications, in general. These matrices, however, being often sparse ones whose elements are mostly either zeros or unities, the computation amount tends to be much less than the estimation above.

2) The Robot Case: Since \( F_1 \) and \( F_2 \) may be set to be diagonal matrices, (23) can be rearranged into \( p \) decoupled sets of two first-order state equations as follows:

\[
\begin{align*}
(\dot{z}_1)_i &= (z_2)_i + (F_1)_i(z_1 - \theta)_i \\
(\dot{z}_2)_i &= (\theta_1)_i - (K_s)_i(z_2 - \theta_3)_i - (K_p)_i(z_1 - \theta_1)_i \\
&+ (F_2)_i(z_1 - \theta)_i \quad \text{with } i = 1 \cdots p (27)
\end{align*}
\]

where \( p \) denotes the number of the degrees of freedom of the robot, and \( (\cdot)_i \) denotes either the \( i \)th diagonal element for matrices, or the \( i \)th element for vectors. For each set of the state equations (or each joint), after having transformed into the discretized form, one needs \( 9p \) additions and \( 12p \) multiplications. Hence, the amount of computation for a \( p \) degrees of freedom robot totals \( 9p \) additions and \( 12p \) multiplications.

Note that these computation amounts are those required at each observer sampling time; the amounts at each controller sampling time are about four to five times larger.

### V. EXPERIMENTS

To assure the validity of the proposed observer in a nonlinear plant, we applied the observer to the trajectory control of a robot driven with high speed. Incidentally, the observer was initially experimented to the position control of a DC servo motor system [17], yet the robot system—a system with a higher degree of nonlinearity—would be more appropriate for our experiment.

To evaluate the performance of the observer, the command following ability has been examined under the following conditions:

- When there are substantial inertia variations due to changes in payloads, thereby testing the robustness against parameter variations, and
- When external disturbances are applied to the robot, thus testing the disturbance rejection ability.

In addition, the sensitivity to sensor noise was also examined.

More specifically, the command following ability was examined with regard to the tracking accuracy for two kinds of commands: a step type and a trajectory type. The robustness was tested by obtaining the position responses with three payload conditions: 0, 10, and 20 kg (0, 100, and 200% of the maximum rated payload). The external disturbance was produced by attaching a spring to the end effector, which exerted torques of up to 6.74 and 4.86 kgf-m (about 34% of the maximum rated torques) for joints 1 and 2, respectively. Furthermore, the sensitivity to noise was compared in the reconstructed states (joint velocity and acceleration) by the two methods: using the observer and the numerical differentiation.

Obviously, how the observer performs as compared with the numerical differentiator is our immediate interest, yet how they compare to the control schemes other than TDC appears important as well. Thus we included in the comparison a well-tuned PID control, the performance of which is widely recognized.

The robot used in the experiment is a SCARA robot having two degrees of freedom. The lengths of the two links are \( l_1 = 35 \) cm, \( l_2 = 20 \) cm, respectively; their masses are \( m_1 = 11.17 \) kg, \( m_2 = 6.82 \) kg; the distance from the joint axis to the center of mass for each link is \( l_1 = 30 \) cm, and \( l_2 = 18 \) cm, respectively; the moment of inertia about the joint axis \( l_1 = 1.03 \) kg m\(^2\), \( l_2 = 0.224 \) kg m\(^2\), each. At joint 1, an AC servo motor with a stall torque of 239 N cm is used to transmit power through a harmonic drive with gear reduction ratio of 100:1, while at joint 2 a motor having a stall torque of 92 N cm is used with gear reduction ratio 80:1. Each joint has a resolver attached at its shaft for sensing the angular displacement with the resolution of 4096 pulses/rev. The digital implementation of the controller and observer were made with the sampling frequency of 800 Hz in a multiprocessors based system called CONDOR, which is described in detail in [18].

#### A. System Representation

The dynamics of the planar robot, with the gravity term in (4) neglected, may be reduced to the following:

\[
M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + D(\theta, \dot{\theta}) = r
\]

where the joint vector, \( \theta = [\theta_1 \ldots \theta_2] \), the inertia matrix

\[
M(\theta) = \begin{bmatrix} a_1 + 2a_2 \cos \theta_2 & a_3 + a_2 \cos \theta_2 \\ a_3 + a_2 \cos \theta_2 & a_3 \end{bmatrix}
\]

with \( a_1 = l_1^2 m_2 + l_1^2 (m_1 + m_2) \), \( a_2 = l_2 l_2 m_2 \), and \( a_3 = l_2^2 m_2 \), and

- The values of \( a_1, a_2 \), and \( a_3 \) were obtained, by using an off-line estimation method based on the least-square method, to be \( \dot{a}_1 = 0.0053 \) kg \( \cdot \) m\(^2\), \( \dot{a}_2 = 0.0001 \) kg \( \cdot \) m\(^2\), and \( \dot{a}_3 = 0.0040 \) kg \( \cdot \) m\(^2\).
Equation (28) may be expressed in a state equation form as

$$\frac{d}{dt} [\theta] = \begin{bmatrix} -M^{-1}(\theta)(V(\theta, \dot{\theta}) + D(\theta, \dot{\theta})) & + & M^{-1}(\theta) \\ \tau_1 \\ \tau_2 \end{bmatrix}.$$  (30)

The reference model is specified with a linear time-invariant system with the following system matrix and input distribution matrix

$$A_m = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\omega_n^2 & 0 & -2\zeta\omega_n & 0 \\ 0 & -\omega_n^2 & 0 & -2\zeta\omega_n \end{bmatrix};$$

$$B_m = \begin{bmatrix} 0 & 0 \\ \omega_n^2 & 0 \\ 0 & \omega_n^2 \end{bmatrix}.  \tag{31}$$

In addition, if $M(\theta)$, as proposed by Hsia [4], is approximated by $\hat{M}$

$$\hat{M} = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix}.  \tag{32}$$

where $\alpha$ is a constant, the control law is rendered to an especially simple and efficient form as

$$\begin{bmatrix} \tau_1(t) \\ \tau_2(t) \end{bmatrix} = \begin{bmatrix} \tau_1(t - L) \\ \tau_2(t - L) \end{bmatrix} - \hat{M} \begin{bmatrix} \dot{\theta}_1(t - L) \\ \dot{\theta}_2(t - L) \end{bmatrix}$$

$$+ \hat{M} \begin{bmatrix} \omega_n^2 & 0 \\ 0 & \omega_n^2 \end{bmatrix} \begin{bmatrix} \theta_{d1} - \theta_1 \\ \theta_{d2} - \theta_2 \end{bmatrix} + \begin{bmatrix} 2\zeta\omega_n & 0 \\ 0 & 2\zeta\omega_n \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \tag{33}$$

where the states are reconstructed by the observer in (23).

B. Experimental Results

Figs. 2–4 show the responses of joints 1 and 2 to the step input of $\theta_{d1} = \theta_{d2} = 60$ degree in three different cases: with no payload nor disturbances, with payload only, and with payload and disturbance. Similarly Figs. 5–7 show the
responses in these three cases, this time to a trajectory-type command input, which is

\[ \theta_1 = 30 \sin(\pi t) + 30 \sin(1.5\pi t) \text{ degree}; \quad 0 < t < 4 \text{ s} \]
\[ \theta_2 = -30 \sin(\pi t) - 30 \sin(1.5\pi t) \text{ degree}; \quad 0 < t < 4 \text{ s}. \]

Note that both the two commands require joint velocities of about 200 degrees/s—approximately the maximum joint speeds available—which may be regarded as fast enough to induce sufficient nonlinearities.

The reference model was determined so that its natural frequency may be \( \omega_n = 10 \), and damping ratio \( \zeta = 1 \). Besides, the value of \( \alpha \) in \( M \) was obtained in order that it may satisfy the stability condition proposed by Hsia [4].

As shown in Fig. 2, under no payload and no disturbance, one has similar performances with the three control schemes, yet slightly better results with PID than the others in terms of overshoot and rising time. The comparison of the reconstructed velocities and accelerations confirms that the observer reconstructs the states smoothly and stably. By comparison, the numerical differentiation is observed to amplify the sensor noise to a substantial degree.

As shown in Fig. 3, the increase in the payload (0, 100, and 200% of the maximum rated payload) results in sluggish responses (larger rise time), with all of the three control schemes; more prominent with PID, and the least with TDC using numerical differentiation. Furthermore, it is noteworthy that the response with PID becomes more oscillatory and shows larger overshoot as the payload increases, perhaps owing to the decrease in the effective damping. In contrast, the other schemes using TDC do not show any new oscillation or overshoot that was not present in the no payload case, thereby demonstrating the robustness of TDC to parameter variations.

Fig. 4 shows that the addition of the spring disturbance to the payload variations has little effect to the responses with TDC, confirming the disturbance rejection ability of TDC.
observed in other experiments. With PID, it appears that one can achieve similar disturbance rejection. Besides, the decrease in overshoot in joint 1 seems to result from the fact that the spring force is exerted opposite to the direction of the overshoot, thus suppressing it.

Figs. 5–7 show the responses to a trajectory type command, which—owing to the continuously changing motion pattern—could be much more demanding in terms of the command following ability of a system. A general tendency one can observe is that the tracking accuracy of PID, different from the step response cases, becomes noticeably worse than the two other schemes. Other aspects also deserve detailed discussions as follows.

Fig. 5 shows the case with no payload. The tracking accuracy of joint 2 is similar with the two schemes using TDC, whereas that of joint 1 with the observer is better than that with numerical differentiator. More specifically, initially \((0 < t < 0.3 \text{ s})\) the two schemes achieve similar accuracy; subsequently, however, the accuracy with the observer becomes the better.

As the payload increases, the tracking accuracy with numerical differentiator becomes even worse—as observed in Fig. 6, not only that of joint 1, but also that of joint 2 deteriorate, revealing rugged subharmonics with substantial magnitudes. By contrast, the scheme using observer shows virtually no changes to the payload variation except at the initial response of joint 1.

Similar to the step responses, the spring disturbance causes little effect to the responses as shown in Fig. 7; only slight differences are observed at the initial responses of joint 1. In addition, the comparison of the reconstructed velocities and accelerations shown in Fig. 5 indicates the advantage of using the observer.

**VI. CONCLUSION**

In this paper an observer design method for TDC has been proposed and the overall stability of observer and controller has been discussed. It was shown that the proposed observer reconstructs states and their derivatives work quite well in
the presence of plant uncertainties, while preserving the performance of TDC alone. Thus the TDC algorithm may be extended to the systems where all states and their derivatives are not measurable. In addition, since the proposed observer does not require an accurate model of plant dynamics, its structure is simple and easy to implement.

The effectiveness of the proposed observer was evaluated through the experiments in a high-speed robot. It was demonstrated that the designed control system is quite robust to external disturbance and inertia variation. It turned out that the proposed observer worked well, and the overall system of observer and controller was less sensitive to sensor noise.

ACKNOWLEDGMENT

The authors would like to thank Samsung Electronics Co. for providing a FARA robot, with which the experiments have been made.

REFERENCES