EDA-Based Speed Control of a Networked DC Motor System With Time Delays and Packet Losses

Hongbo Li, Mo-Yuen Chow, *Fellow, IEEE*, and Zengqi Sun, *Senior Member, IEEE*

***Abstract*—This paper presents a new controller design method for networked dc motor system in the presence of time delays and packet losses. The sufficient condition under which the closed-loop system is asymptotically stable and the necessary condition under which the networked dc motor has zero steady-state tracking error are derived. Based on the obtained conditions, an output tracking controller design method is proposed, where the Estimation of Distribution Algorithm is used to optimize the control parameters to improve the system control performance. The proposed method can be easily applied to various applications, since it is simple and has no assumptions on time delay and packet loss models. Simulation and experimental results are given to demonstrate the effectiveness of the proposed approach.**

***Index Terms*—Estimation of Distribution Algorithm (EDA), networked control systems (NCSs), packet loss, time delay.**

I. Introduction

THE control of dc motors has been an attractive subject of research during the past several decades since dc motors have been widely utilized in various fields. In traditional dc motor control, system components are located in the same place and connected by point-to-point wiring. However, in many practical systems, dc motor and controller are difficult to be located in the same place, and thus, signals are required to be transmitted from one place to another. To reduce the cost of installation and offer the ease of maintenance, communication networks are used to connect those system components, which gives rise to the so-called networked control systems (NCSs). Recently, NCSs have attracted much attention from research communities. Issues such as time delays [1]–[4], packet losses [8], scheduling [9], and security [10] have been investigated with results reported in the literature. In addition, due to their great advantages, NCSs have been finding applications in a broad range of areas such as dc motors [1]–[3], robots [4], vehicles [5], car suspension system [6], ball maglev system [7], rotary manipulator [17], and process control [18].

Time delay in NCSs is a major cause for system performance deterioration and potential system instability. In the literature, time delays have been modeled by using various formulations, such as constant delay, independently random delay [11], or random delay governed by a Markov chain. Another important issue in NCSs is the packet loss phenomenon. There are two typical approaches to describe packet losses in the literature. The first approach assumes that packet losses follow certain probability distributions such as Markov chain. The second one is the deterministic method, which specifies the packet losses in a time average sense or places bound on the number of consecutive packet losses. Note that the packets in NCSs usually suffer time delays and packet losses simultaneously during network transmissions. Therefore, the analysis and synthesis of NCSs with time delays and packet losses are a challenge but a practical problem. In the literature, some important methodologies, such as stochastic control [11], robust control [5], fuzzy control [4], predictive control [3], and memoryless state-feedback control (MSFC) [12]–[15], are proposed to compensate the time delays and/or packet losses. For more details on this topic, we refer the reader to [3]–[5], [11]–[15], and the references therein.

MSFC is a simple and effective control method. It has a simple structure and does not require controller memory to store past information. Therefore, the MSFC of NCSs with time delays and packet losses has received considerable attention in the past few years. Among the reported results in the literature, to mention a few, the stability issue is addressed in [12], the stabilization problem is investigated in [13] and [14], and the optimal stabilizing controller is designed in [15]. Note that the MSFC controllers in the aforementioned results are all regulators and none of them takes output tracking problem into account. However, it is well known that output tracking control has been widely used in various fields, and it has been well recognized that output tracking controller design is more general than regulator design, which motivates this paper.

In this paper, we address the optimal output tracking problem for a networked dc motor system with time delays and packet losses. The sufficient condition under which the closed-loop system is asymptotically stable and the necessary condition under which the networked dc motor has zero steady-state tracking error for a step reference speed are derived. An output tracking controller design method is proposed, where the Estimation of Distribution Algorithm (EDA) is used to optimize the

Manuscript received June 10, 2008; revised January 5, 2009. First published February 6, 2009; current version published April 29, 2009. This work was supported in part by the National Science Foundation of China under Grants 60574035 and 60674053, in part by the National Key Project for Basic Research of China under Grant 2002cb312205, and in part by the U.S. National Science Foundation under Grant IIS-0426852.

H. Li is with the Electrical and Computer Engineering Department, North Carolina State University, Raleigh, NC 27606 USA, and also with the Department of Computer Science and Technology, State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing 100084, China (e-mail: hli7@ncsu.edu; hb-li04@mails.tsinghua.edu.cn).

M.-Y. Chow is with the Electrical and Computer Engineering Department, North Carolina State University, Raleigh, NC 27606 USA (e-mail: chow@ncsu.edu).

Z. Sun is with the Department of Computer Science and Technology, State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing 100084, China (e-mail: szq-dcs@mail.tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2009.2013749

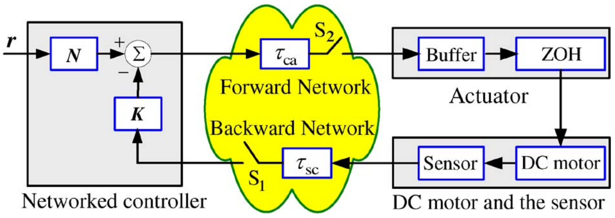


Fig. 1. Structure of networked dc motor system.

control parameters to improve the system control performance. Note that our previous works [15] and [16] have shown the advantages of using EDA to optimize the control parameter for networked system, and there are at least two significant improvements of this paper over our previous works [15] and [16]. The first is that the framework of this paper is more general. Reference [15] assumes that time delays are less than two sampling periods, and [16] allows the network from sensor to controller only. This paper eliminates those assumptions and constraints. The second is that this paper studies output tracking controller design while [15] and [16] address the regulator design only.

The remainder of this paper is organized as follows. The problem formulation is given in Section II. Some preliminary results for networked dc motor system are derived in Section III. The controller design problem is solved in Section IV. Numerical and experimental results are provided in Section V. Finally, we conclude this paper in Section VI.

II. Problem Formulation

As shown in Fig. 1, the networked dc motor system considered in this paper can be grouped into the following four modules: 1) the dc motor and the sensor; 2) the communication network; 3) the networked controller; and 4) the actuator. Each module is described in the following sections.

A. DC Motor and the Sensor

The electromechanical dynamics of the dc motor can be described as

$$\frac{di_a}{dt} = -\frac{R}{L}i_a - \frac{K_b}{L}\omega + \frac{1}{L}u \tag{1}$$

$$\frac{\omega}{dt} = \frac{K}{J}i_a - \frac{B}{J}\omega \tag{2}$$

where i_a is the armature winding current, ω is the rotor angular speed, R is the armature winding resistance, L is the armature winding inductance, K_b is the back electromotive force constant, u is the armature winding input voltage, K is the torque constant, J is the system moment of inertia, and B is the system damping coefficient.

By letting $\boldsymbol{x} \triangleq [i_a, \omega]^{\mathrm{T}}$, the dc motor can be expressed as

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) \end{cases} \tag{3}$$

with $\boldsymbol{y}(t)$ being the rotor angular speed and

$$\boldsymbol{A} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_b}{L} \\ \frac{K}{J} & -\frac{B}{J} \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}$$
$$\boldsymbol{C} = [0 \quad 1] \qquad \boldsymbol{D} = 0. \tag{4}$$

When the sampling period is specified to h, the dc motor can be discretized as

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{F}\boldsymbol{x}(k) + \boldsymbol{G}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) \end{cases} \tag{5}$$

with

$$\boldsymbol{F} = e^{\boldsymbol{A}h} \qquad \boldsymbol{G} = \int_0^h e^{\boldsymbol{A}\tau}\, d\tau \boldsymbol{B}. \tag{6}$$

It is assumed that the full-state variables are available in the networked system. At each sampling period, the sampled dc motor state and its time stamp (i.e., the time the motor state is sampled) are encapsulated into a packet and sent to the controller via the network.

B. Data Network

In practice, the packets in NCSs usually suffer time delays and packet losses during network transmissions. As shown in Fig. 1, sensor-to-controller and controller-to-actuator delays are denoted by τ_{sc} and τ_{ca}, respectively, and two switches S_1 and S_2 are used to model the packet losses in the backward and forward networks. When S_1 is open, the sensor packet is lost during network transmission from the sensor to the controller, whereas when S_1 is closed, the packet is successfully transmitted to the controller with a sensor-to-controller delay τ_{sc}. Similar arrangements are made for the forward network.

C. Networked Controller

The networked controller takes the following form:

$$\boldsymbol{u} = \boldsymbol{K}x + \boldsymbol{N}r \tag{7}$$

where $\boldsymbol{x}$ is the motor state in the arriving sensor packet, $\boldsymbol{r}$ is the reference input, and $\boldsymbol{K}$ and $\boldsymbol{N}$ are the control parameters to be designed. Networked controller (7) is event driven. Whenever there is a sensor packet arriving at controller, the controller starts calculating the control signal. Immediately after the calculation, the new control signal and the time stamp of the used motor state are encapsulated into a packet and sent to the actuator via the network. The time stamp in the control packet will ensure the actuator to select the appropriate control signal to control the dc motor.

D. Actuator

The actuator is time driven. It is assumed that the actuator and the sensor have the same sampling period h and they are synchronized. Note that the actuator and the sensor are at the same location, and therefore, the synchronization between them can be easily achieved by hardware synchronization, for instance, using special wiring to distribute a global clock signal to the sensor and the actuator. The actuator has a buffer size of one, which means that the latest control packet is used to control the dc motor. When a control packet arrives at the actuator, it will be used to update the buffer if its time stamp is newer than that of the control signal in Buffer. Otherwise, it will be discarded. The actuator uses a zero-order hold (ZOH) circuit. At each sampling period, the ZOH circuit reads the control signal from buffer and uses it to control the dc motor.

The objective of this paper is to provide a synthesis method to design optimal stabilizing gain $\boldsymbol{K}$ and coefficient $\boldsymbol{N}$ for networked controller (7), such that the networked system is asymptotically stable, meets specified control performance, and tracks the reference speed without steady-state error.

III. Preliminary Results

This section is devoted to present some preliminary results for the concerned networked system. Those preliminary results will play a significant role in the controller design stage.

A. Sufficient Stability Condition

The sufficient condition under which the gain matrix $\boldsymbol{K}$ of networked controller (7) can guarantee the networked system to be asymptotically stable is presented in this section.

In the considered networked system, a sensor packet is called *effective sensor packet* for networked controller (7), if it is successfully transmitted from the sensor to the controller and to the actuator, and the corresponding control packet is used to control the dc motor. Let $S \triangleq \{i_1, i_2, \ldots\}$, which is a subsequence of $\{0, 1, 2, \ldots\}$, denote the sequence of time indexes of the *effective sensor packets*. Note that only *effective sensor packets* are used to control the dc motor, and therefore, the sensor packets between two *effective sensor packets* can all be considered as dropped packets.

Based on the aforementioned notations, the packet loss in the networked dc motor system can be defined as

$$\left\{\eta(i_m) \triangleq i_{m+1} - i_m, \quad i_m \in S\right\} \tag{8}$$

which means that, from i_m to i_{m+1}, the number of dropped packets is $\eta(i_m) - 1$. Without loss of generality, we assume that the maximum number of consecutive packet losses is $\boldsymbol{N}_{\text{drop}}$.

Let $\tau_{i_m} \triangleq \tau_{\text{sc},i_m} + \tau_{\text{ca},i_m}$ express the round-trip time (RTT) delay encountered by the mth *effective sensor packet*. A nature assumption on RTT delay τ_{i_m} can be made that $\tau_{i_m} \leq \tau_{\max}$, where $\tau_{\max}$ denotes the upper bound of τ_{i_m}.

Let $N_{\text{delay}} \triangleq \lceil \tau_{\max}/h \rceil$, where $\lceil \cdot \rceil$ is the ceiling operator, which returns the smallest integer that is greater than or equal to a specified number. Introduce the augmented state $\boldsymbol{z}(i_m) = [\boldsymbol{x}_{i_m}^{\mathrm{T}} \quad \boldsymbol{x}_{i_{m-1}}^{\mathrm{T}} \quad \cdots \quad \boldsymbol{x}_{i_{m-N_{\text{delay}}}}^{\mathrm{T}}]^{\mathrm{T}}$ into the aforementioned networked dc motor system; the closed-loop system can be described by the following discrete-time switched model:

$$\boldsymbol{z}(i_{m+1}) = \boldsymbol{M}_{\lceil \tau_{i_m}/h \rceil} \boldsymbol{z}(i_m) \tag{9}$$

where $\lceil \tau_{i_m}/h \rceil$ is a piecewise constant function called a switch signal, which takes values in a finite set $\mathbb{S} \triangleq \{0, 1, \ldots, N_{\text{delay}}\}$. For $i \in \mathbb{S}$, $\boldsymbol{M}_i$ takes the following form:

$$\boldsymbol{M}_i = \begin{bmatrix} \boldsymbol{F}^{\eta(i_m)} + \Pi_1^i & \Pi_2^i & \cdots & \Pi_{\boldsymbol{N}_{\text{delay}}}^i & \Pi_{\boldsymbol{N}_{\text{delay}}+1}^i \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \tag{10}$$

where

$$\begin{aligned} \Pi_1^i &= \sum_{r=0}^{\eta(i_m)-i} \mathbf{F}^r \mathbf{G} \boldsymbol{K} \\ \Pi_j^i &= \sum_{r=0}^{c_j^i} \mathbf{F}^{\eta(i_m)-i+r+1} \mathbf{G} \boldsymbol{K} \; (j \in [2, \ldots, i+1]) \\ \Pi_j^i &= 0, \qquad j \in [i+2, \ldots, N_{\text{delay}}+1]. \end{aligned} \tag{11}$$

For more details of the modeling process and the used notations, please refer to our previous work [14]. Reference [14] has also shown that networked dc motor system (9) in the presence of time delays and packet losses is asymptotically stable, if, for $i, j \in \mathbb{S}$, there exist positive definite matrices $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ satisfying

$$\boldsymbol{M}_i^{\mathrm{T}} \boldsymbol{P}_j \boldsymbol{M}_i - \boldsymbol{P}_i < 0 \tag{12}$$

where $\boldsymbol{M}_i$ is of form (10) with a given gain matrix $\boldsymbol{K}$.

Noted that, in the considered networked system, there is no assumption on the time delay or packet loss model (i.e., the time delay can be constant, independently random, or random governed by a Markov chain, and the packet losses can be independently random or random governed by a Markov chain). Therefore, the NCS model (9), the stability condition (12), and the methods proposed in the following sections are quite general and can be applied to various applications.

B. Necessary Condition for Zero Steady-State Tracking Error

The necessary condition under which the networked dc motor has zero steady-state tracking error for a step reference speed is presented in this section.

The basic idea of addressing the concerned problem is to compute the steady-state values of the motor state and the control input (denoted as $\boldsymbol{x}_{\text{ss}}$ and $\boldsymbol{u}_{\text{ss}}$) that result in zero steady-state tracking error, and then, let them take these values. Following this idea, we consider the networked controller

$$\boldsymbol{u} = \boldsymbol{u}_{\text{ss}} - \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{x}_{\text{ss}}). \tag{13}$$

The rest of the problem is to determine the values of $\boldsymbol{x}_{\text{ss}}$ and $\boldsymbol{u}_{\text{ss}}$ such that the networked system has no steady-state error for any step reference input $\boldsymbol{r}$. To address this problem, let $\boldsymbol{x}_{\mathrm{ss}} = \boldsymbol{N}_x \boldsymbol{r}$ and $\boldsymbol{u}_{\mathrm{ss}} = \boldsymbol{N}_u \boldsymbol{r}$. With those substitutions, (13) is equivalent to

$$\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x} + (\boldsymbol{N}_u + \boldsymbol{K}\boldsymbol{N}_x)\boldsymbol{r}. \tag{14}$$

When the networked system is at steady state, the dynamics of the dc motor (3) can be rewritten as

$$\begin{aligned} 0 &= \boldsymbol{A}\boldsymbol{x}_{\mathrm{ss}} + \boldsymbol{B}\boldsymbol{u}_{\mathrm{ss}} \\ \boldsymbol{y}_{\mathrm{ss}} &= \boldsymbol{C}\boldsymbol{x}_{\mathrm{ss}} + \boldsymbol{D}\boldsymbol{u}_{\mathrm{ss}} \end{aligned} \tag{15}$$

where $\boldsymbol{y}_{\mathrm{ss}}$ is the steady-state output of the dc motor.

Letting $\boldsymbol{y}_{\mathrm{ss}} = \boldsymbol{r}$ and substituting $\boldsymbol{x}_{\mathrm{ss}} = \boldsymbol{N}_x \boldsymbol{r}$ and $\boldsymbol{u}_{\mathrm{ss}} = \boldsymbol{N}_u \boldsymbol{r}$ into (15) lead to

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{N}_x \\ \boldsymbol{N}_u \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{16}$$

By solving (16), $\boldsymbol{N}_x$ and $\boldsymbol{N}_u$ can be easily obtained as

$$\begin{bmatrix} \boldsymbol{N}_x \\ \boldsymbol{N}_u \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{17}$$

Apparently, if $\boldsymbol{N}_x$ and $\boldsymbol{N}_u$ in (14) take the values obtained from (17), the networked system will have zero steady-state tracking error for any step reference speed. By noting the difference between (7) and (14), we obtain

$$\boldsymbol{N} = \boldsymbol{N}_u + \boldsymbol{K}\boldsymbol{N}_x. \tag{18}$$

Therefore, the necessary condition under which the networked dc motor has zero steady-state tracking error for any step reference speed is that the gain matrix $\boldsymbol{K}$ and the coefficient $\boldsymbol{N}$ satisfy (18).

IV. Controller Design

This section is devoted to design the optimal stabilizing gain $\boldsymbol{K}$ and the coefficient $\boldsymbol{N}$ for networked controller (7). We will transform the controller design problem into an optimization problem and then solve the optimization problem.

Optimization Variables: In networked controller (7), the gain matrix $\boldsymbol{K}$ and the coefficient $\boldsymbol{N}$ are two parameters to be designed. Note that the coefficient $\boldsymbol{N}$ can be determined according to (18), which is a function of $\boldsymbol{K}$. Therefore, the optimization variables in the optimization problem are in fact only the state-feedback gain matrix $\boldsymbol{K}$, which takes the following form:

$$\boldsymbol{K} = [K_1, K_2]. \tag{19}$$

Cost Function: The *percent overshoot* and the *settling time* are used to evaluate the networked dc motor system performance. The corresponding cost function is defined as follows:

$$\begin{aligned} J &= \lambda_1 J_1 + \lambda_2 J_2 \\ J_1 &= \begin{cases} 0, & P.O. \leq P.O._n \\ (P.O. - P.O._n)/P.O._n, & P.O. > P.O._n \end{cases} \\ J_2 &= \begin{cases} 0, & S.T. \leq S.T._n \\ (S.T. - S.T._n)/S.T._n, & S.T. > S.T._n \end{cases} \end{aligned} \tag{20}$$

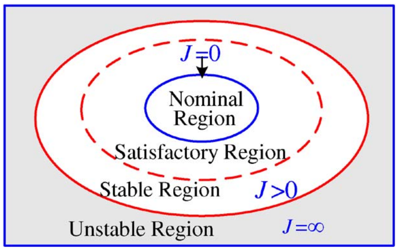


Fig. 2. Graphical representation of the regions used in this paper.

where $P.O.$ is the *percent overshoot*, $P.O._n$ is the nominal *percent overshoot*, $S.T.$ is the *settling time*, and $S.T._n$ is the nominal *settling time*. $P.O._n$, and $S.T._n$ are nominal performance measures that the system should achieve without network in the loop. If the NCS performance meets the nominal performance specifications, we have $J = 0$ because $J_i = 0$, where $i = 1, 2$. The cost J_1 penalizes high value of the percentage overshoot, and the cost J_2 penalizes long settling time. The problem dependent weights λ_1 and λ_2 are used to specify the relative significance of J_1 and J_2 on the overall system performance.

SRC: A vector set $\Psi \subset \mathbb{R}^{1\times 2}$ is called the *stable region* (Fig. 2) of the networked dc motor system, if any $\boldsymbol{K} \in \Psi$ satisfies the stability conditions described in (12), i.e., Ψ is composed of $\boldsymbol{K}$ that guarantees the networked system to be asymptotically stable. Obviously, the controller gains that can provide satisfactory control performance must be within the defined *stable region*, since satisfactory control performance not only requires a system to be asymptotically stable but also requires the system to meet given performance specifications. Therefore, it is sufficient to search the optimal controller gain within the *stable region*, which can effectively reduce the search space and guarantee the stability of NCSs during the optimization procedure. Based on this idea, *stable region constraint* (SRC) is introduced as follows:

$$SRC : \boldsymbol{V} \in \Psi \qquad \forall\, \boldsymbol{V} \tag{21}$$

which means that the optimization variables are constrained to the *stable region* throughout the optimization procedure.

OTC: To enable the networked dc motor track a reference speed with zero steady-state error, *output tracking constraint* (OTC) is introduced as follows:

$$OTC : \boldsymbol{N} = \boldsymbol{N}_u + \boldsymbol{K}N_x \tag{22}$$

which means that the gain matrix $\boldsymbol{K}$ and the coefficient $\boldsymbol{N}$ satisfy (18) during the optimization procedure.

In summary, the optimization problem can be expressed as

$$\begin{aligned} &\textbf{OP} : \min \boldsymbol{J}(\mathrm{NCS}, \boldsymbol{K}) \\ &\quad \text{s.t. SRC} : \boldsymbol{K} \in \Psi \qquad \forall \boldsymbol{K} \\ &\qquad \text{OTC} : \boldsymbol{N} = \boldsymbol{N}_u + \boldsymbol{K}\boldsymbol{N}_x. \end{aligned} \tag{23}$$

Equation (23) asks for $\boldsymbol{K}$ that minimizes the value of the cost function. Obviously, SRC can guarantee the stability of the networked system during the optimization procedure. On the other hand, by minimizing the cost function, the system control performance is optimized. Therefore, solving the optimization problem (23) offers a substantial advantage in that both stability and control performance can be guaranteed in the controller design stage.

For the optimization problem (23), heuristic search algorithms such as a genetic algorithm can be adopted to solve it. However, the behavior of these heuristic search algorithms depends on many parameters. For a researcher without associated experiences, the choice of suitable values for so many parameters is difficult. Unlike other evolutionary algorithms (EAs), EDA does not use crossover or mutation. Instead, EDA uses a probability distribution model derived from the promising solutions to generate new population. Thus, one of the most appealing advantages of EDA over classical EAs is the reduction in the number of parameters to be tuned by the user. Moreover, it is noted that the optimization problem (23) is subjected to SRC. Therefore, to accelerate the search speed, a generic characteristic expected for the heuristic search algorithm is that it can search the optimal solution within the defined stable region with high probability. As mentioned before, EDA adopts a probability distribution model derived from the promising solutions to generate new population, which enables EDA to possess the required characteristic. Motivated by the aforementioned analyses, we adopt EDA to solve the optimization problem (23) in this paper. The optimization variables in the search space are modeled as a multivariate normal distribution $p(\boldsymbol{K}) = \prod_{i=1}^{2} p(K_i)$, which is a product of two independent univariate normal distributions $p(K_i) = (\mu_i, \sigma_i)$. The EDA-based MSFC (E_MSFC) algorithm proposed in this paper can be summarized as follows.

1) **Initialization**. Generate n_1 gain candidates $\boldsymbol{K}^i = [K_1^i, K_2^i](i = 1, \ldots, n_1)$ meeting SRC in the initial search space randomly; for each $\boldsymbol{K}^i$, compute the corresponding coefficient $\boldsymbol{N}^i$ according to (18).
2) **Selection**. Compute the cost value J^i for each pair $(\boldsymbol{K}^i, \boldsymbol{N}^i)$; select the best $n_2 (n_2 < n_1)$ gain candidates $\boldsymbol{K}^i$ from the parent generation according to the cost values J^i.
3) **Updating**. Update the probability distribution model $p(\boldsymbol{K})$ using the selected n_2 promising gain candidates.
4) **Sampling**. Select the best gain candidate in the current population as $\boldsymbol{K}^1 = [K_1^1, K_2^1]$, generate $n_1 - 1$ gain candidates $\boldsymbol{K}^i = [K_1^i, K_2^i](i = 2, \ldots, n_1)$ meeting SRC based on the updated $p(\boldsymbol{K})$, and, for each $\boldsymbol{K}^i$, compute the corresponding coefficients $\boldsymbol{N}^i$ according to (18).
5) If the termination criterion is met, EXIT; otherwise, go back to step 2).

For more details on EDA, please refer to [19] and the references therein. Note that, although we do not know the exact *stable region* of an NCS, we can realize SRC based on the stability conditions described in (12). Taking step 1), for example, we can generate a gain candidate in the search space randomly and then check whether the gain candidate satisfies the stability conditions described in (12). If it does, the gain candidate will be selected as an individual of the population. Otherwise, we discard it and repeat the aforementioned steps until we obtain one satisfying the stability conditions described in (12). Similar arrangements are made for step 4).

TABLE I
PARAMETERS OF THE DC MOTOR

J	Inertia	$1.6511 \times 10^{-4} \mathrm{kg} \bullet \mathrm{m}^2$
L	Inductance	$3.14 \times 10^{-3} \mathrm{H}$
R	Resistance	4.961Ω
K	Torque Constant	$7.105 \times 10^{-2} \mathrm{N} \bullet \mathrm{m/A}$
B	Damping coefficient	$23.64 \times \mathrm{e}^{-6} \mathrm{N} \bullet \mathrm{m} \bullet \mathrm{sec/rad}$
K_b	Back-EMF Constant	$1.276 \times 10^{-2} \mathrm{Vs/rad}$

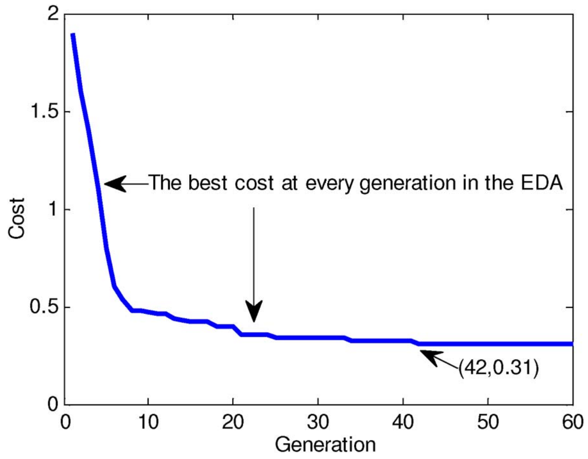


Fig. 3. Typical evolution process of EDA in solving the optimization problem.

V. Illustrative Example

To show the effectiveness of E_MSFC for networked dc motor speed control, it is applied to a networked dc motor system. A comparative study of E_MSFC with two existing approaches is constructed, where both numerical simulations and experimental verifications are provided.

The parameters of the dc motor used in this paper are listed in Table I. The sampling period of the networked system is specified to 0.1 s. To carry out the comparative study, it is assumed that $\tau_{i_m} \in (0 \text{ s}, 0.25 \text{ s}]$ and $\boldsymbol{N}_{\text{drop}} = 4$, which means that up to four consecutive packets can be lost during network transmissions. The performance specifications for the networked dc motor system are developed as follows:

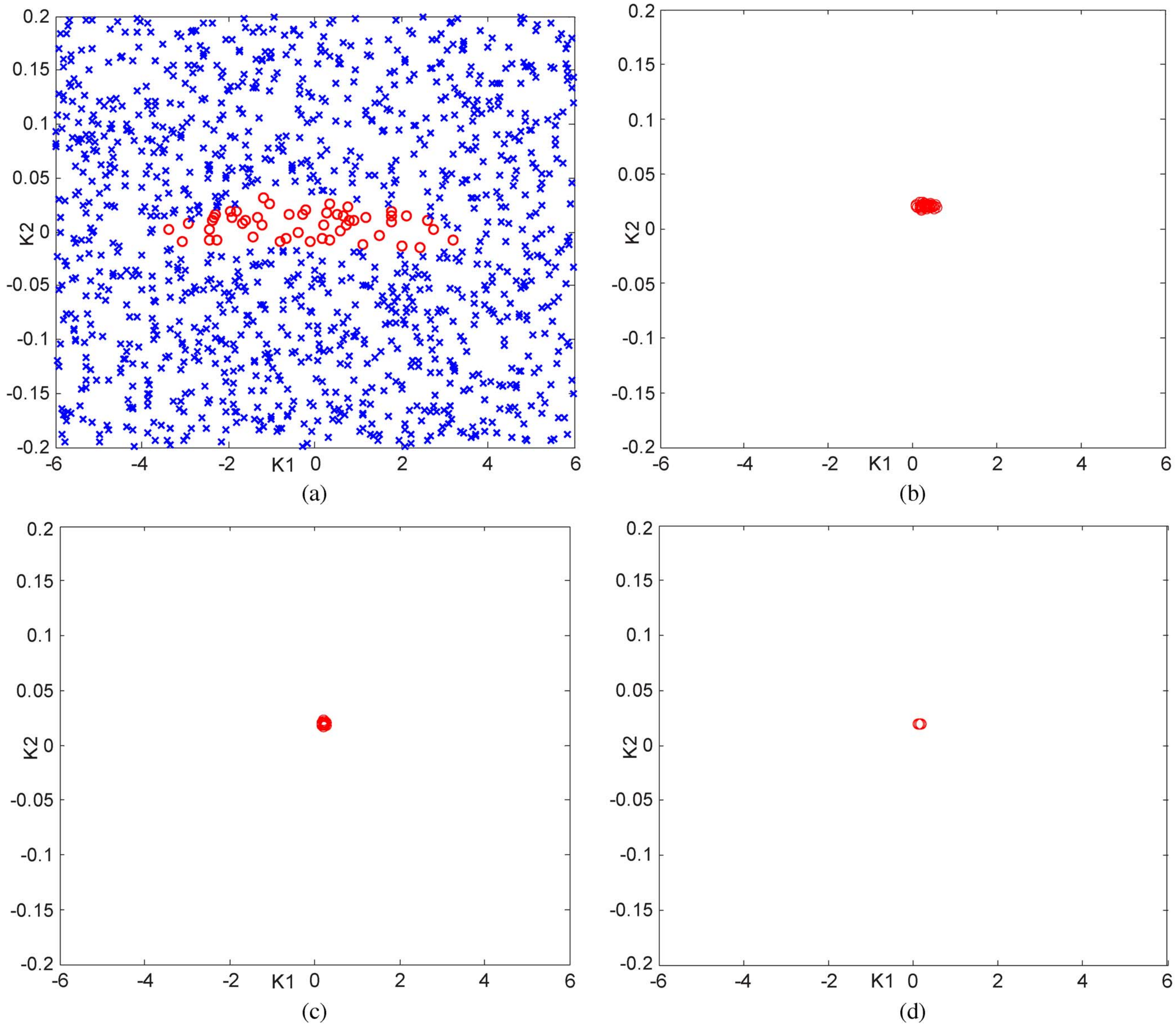
1) *percent overshoot*: $P.O. \leq 3\%$;
2) *settling time*: $S.T. \leq 2$ s.

A. Networked Controller Design

First, let us consider the state-feedback controller (SFC) proposed by Yue *et al.* in [13]. By using [13, Th. 1] to the aforementioned networked system, we obtain an SFC $\boldsymbol{u}(t) = 10^{-4}[0.4043, -0.0417]\boldsymbol{x}(t - \tau_k)$.

Then, we apply the state-feedback stabilizing controller (SFSC) in our previous work [14] to the aforementioned networked system and obtain an SFSC $\boldsymbol{u} = [-0.0031, -0.0056]\boldsymbol{x}$.

Finally, let us consider the proposed E_MSFC method. This example uses $n_1 = 50$ and $n_2 = 20$ for the EDA algorithm, with $P.O._n$, $S.T._n$, λ_1, and λ_2, in the cost function (20) specified to 1%, 1 s, 1, and 1, respectively. Fig. 3 shows the typical evolution process of EDA in solving the optimization problem (23), which clearly shows that the system control performance is improved from generation to generation by



(a)

(b)

(c)

(d)

Fig. 4. Population distribution of EDA at the (a) first, (b) fifth, (c) tenth, and (d) fortieth generations.

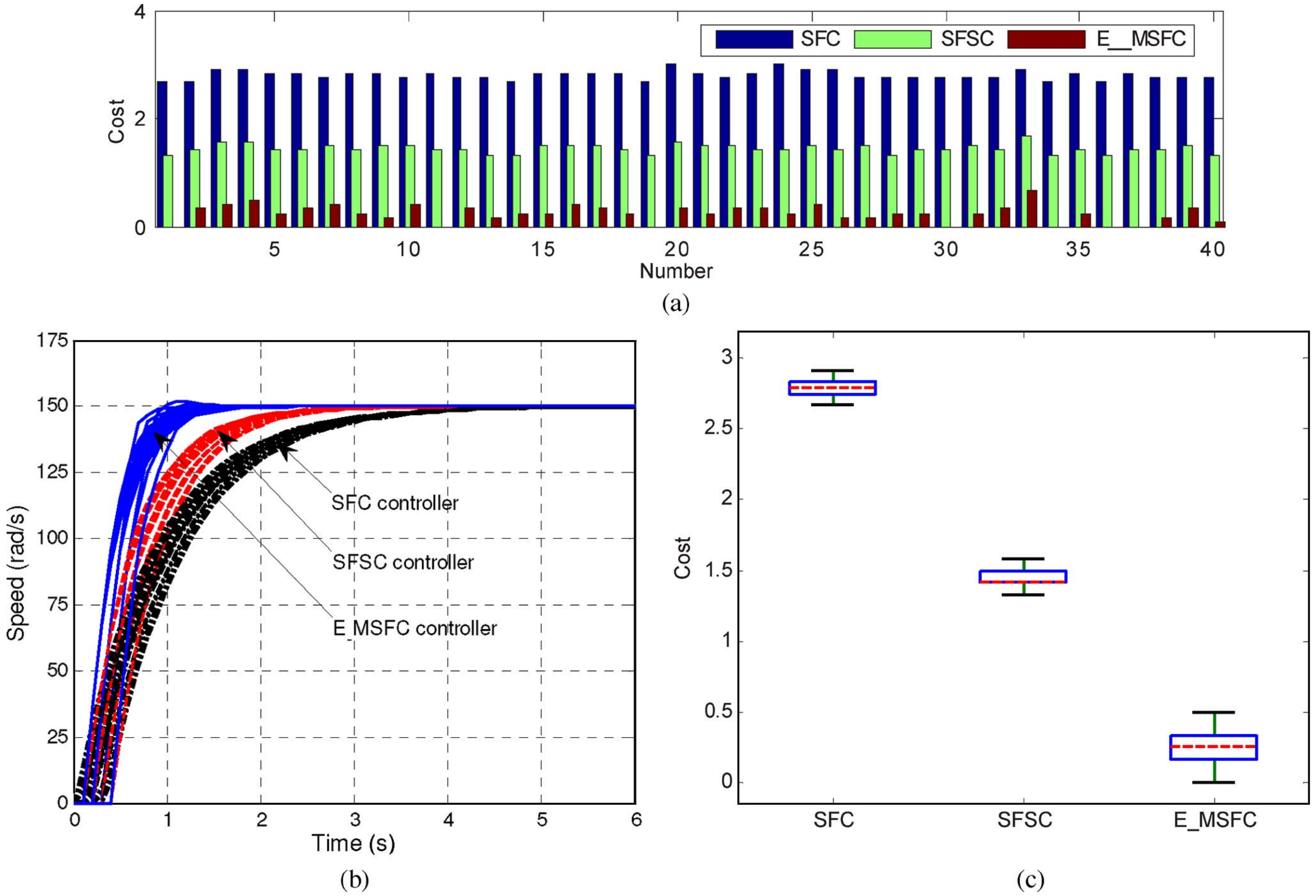
using EDA. The population distribution of EDA at different generations is shown in Fig. 4, where the circles represent the individuals of population and the crosses represent the discarded gain candidates based on the SRC criteria. From Fig. 4(a), we can approximate the boundary of the *stable region* of the networked dc motor system. Fig. 4 also shows that EDA can generate a new population in the *stable region* for the next generation with high probability, and from generation to generation, the search space gets smaller and smaller, which can accelerate the search speed. That demonstrates the advantage of using EDA in solving the optimization problem (23). As shown in Fig. 3, the optimal cost of 0.31 has been achieved at the forty-second generation, where the designed controller is $\boldsymbol{u} = [-0.1588, -0.0190]\boldsymbol{x} + 0.0311\boldsymbol{r}$.

Note that the networked system is designed to drive the dc motor to a preset speed. However, the SFC and SFSC are regulators, and therefore, they cannot drive the dc motor to a nonzero speed. For comparison purposes, we modify the two comparative controllers by further introducing the reference input. We use the same feedback gain matrices but further apply (17) and (18) to them. As a result, the two comparative controllers are modified as $\boldsymbol{u}(t) = 10^{-4}[0.4043, -0.0417]\boldsymbol{x}(t - \tau_k) + 0.0196\boldsymbol{r}$ and $\boldsymbol{u} = [-0.0031, -0.0056]\boldsymbol{x} + 0.0140\boldsymbol{r}$, respectively.

It is worth noting that our previous works [15] and [16] cannot be applied to the aforementioned networked system since both-sides network is considered in the networked system and the RTT delays are larger than two sampling periods. Therefore, [15] and [16] are not considered in the comparative study. This remark also demonstrates the significant improvement of this paper over our previous works [15] and [16].

B. Numerical Simulations

In the numerical simulation scenario, the networked dc motor system is simulated using MATLAB under fully controlled environments. With the initial state $\boldsymbol{x}_0 = [0, 0]^{\mathrm{T}}$, 40 paired simulation results are shown in Fig. 5, where Fig. 5(a) shows the



(a)

(b)

(c)

Fig. 5. Typical simulation results using different networked controllers.

TABLE II
STATISTICS OF NETWORKED DC MOTOR PERFORMANCE

	SFC	SFSC	E_MSFC
Mean value	2.7979	1.4437	0.3134
Standard deviation	0.0883	0.0809	0.1550

system cost values, Fig. 5(b) shows the system performances, and Fig. 5(c) shows the corresponding box-and-whisker diagram over the 40 runs. The statistics of the 40 simulation results are listed in Table II. By paired simulation results, we mean that the same network condition is used in the simulation of networked system using the SFC, the SFSC, and the E_MSFC controller. Note that the 40 paired runs used 40 different network conditions and they were generated randomly under the given network constraints $\tau_{i_m} \in (0\ \text{s}, 0.25\ \text{s}]$ and $\boldsymbol{N}_{\text{drop}} = 4$.

Two-sample tests of hypothesis are also performed on the aforementioned simulation results. First, we use the hypothesis test to determine whether the E_MSFC controller provides higher control performance statistically than the SFC does. For this purpose, the null hypothesis

$$H_0^1 : J_{\text{E\_MSFC}} \geq J_{\text{SFC}} \tag{24}$$

is tested against the alternative hypothesis

$$H_a^1 : J_{\text{E\_MSFC}} < J_{\text{SFC}}. \tag{25}$$

The decision is made based on the P-values of the tests that show the probability of obtaining the existing experimental data given the null hypothesis; thus, a low P-value leads to the rejection of the null hypothesis. Let $J_{\text{dif}} \triangleq J_{\text{E\_MSFC}} - J_{\text{SFC}}$, where $J_{\text{E\_MSFC}}$ and J_{SFC} denote the system cost using the E_MSFC controller and the SFC, respectively. Corresponding to the 40 paired simulation results shown in Fig. 5(a), the mean and standard deviation of J_{dif} are given as follows:

$$\bar{d} = -2.5500 \qquad s_d = 0.1397. \tag{26}$$

Substituting $\bar{d}$ and s_d into the test statistic t, we have

$$t = \frac{\bar{d}}{s_d/\sqrt{n}} = \frac{-2.55}{0.1397/\sqrt{39}} = -113.9924. \tag{27}$$

Corresponding to $t = -113.9924$, the level of significance is p-value < 0.0001. Thus, we reject the null hypothesis H_0^1 and conclude that there is significant evidence (p-value $<$ 0.0001) that E_MSFC controller provides smaller cost value than SFC does. Moreover, based on the mean value of system cost using SFC and E_MSFC controller (denoted as $\bar{J}_{\text{SFC}}$ and $\bar{J}_{E\_\text{MSFC}}$, respectively), as listed in Table II, we can further conclude that, in comparison with SFSC on the average, E_MSFC controller reduces the system cost by 88.8% $[(\bar{J}_{\text{SFC}} - \bar{J}_{E\_\text{MSFC}})100\%/\bar{J}_{\text{SFC}}]$.

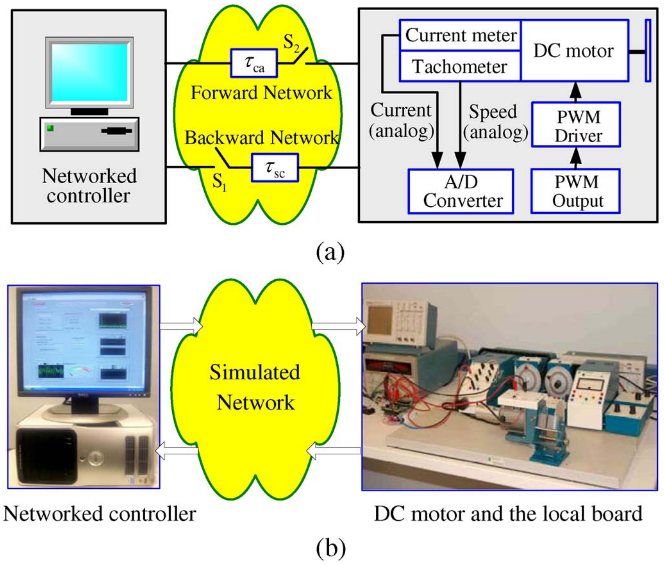


Fig. 6. (a) Block diagram of networked dc motor system. (b) Actual networked dc motor system setup.

For the two-sample tests of E_MSFC controller and SFSC, the null and alternative hypotheses are given by

$$H_0^2: \quad J_{E\_\mathrm{MSFC}} \geq J_{\mathrm{SFSC}} \tag{28}$$

$$H_a^2: \quad J_{E\_\mathrm{MSFC}} < J_{\mathrm{SFSC}}. \tag{29}$$

By following a similar procedure as described in the first hypothesis test, we obtain $t = -112.9102$ for this case. Corresponding to $t = -112.9102$, the level of significance is p-value < 0.0001. Thus, we reject the null hypothesis H_0^2 and conclude that there is significant evidence (p-value < 0.0001) that E_MSFC controller provides smaller cost value than SFSC does. Moreover, based on the mean value of system cost using SFSC and E_MSFC controller (denoted as $\bar{J}_{\mathrm{SFSC}}$ and $\bar{J}_{E\_\mathrm{MSFC}}$, respectively), as listed in Table II, we can further conclude that, comparing with SFC on the average, E_MSFC controller reduces the system cost by 78.25% $((\bar{J}_{\mathrm{SFSC}} - \bar{J}_{E\_\mathrm{MSFC}})100\%/\bar{J}_{\mathrm{SFSC}})$.

From the simulation results and aforementioned analysis, we can see that, comparing with SFC and SFSC, E_MSFC controller reduces the system cost by 88.8% and 78.25%, respectively. The substantial performance improvement of E_MSFC controller comes as no surprise since the performance optimization is considered in the E_MSFC controller design.

C. *Experimental Verifications*

To further illustrate the effectiveness and applicability of E_MSFC, a networked dc motor platform is constructed. The block diagram of networked dc motor system is shown Fig. 6(a), and the actual system setup is shown in Fig. 6(b).

As shown in Fig. 6, the experimental apparatus consists of a PC, a local board, and a dc motor with sensors. The networked controller was implemented in Labview 8.2 on a PC running Windows XP. The local board is on the dc motor side and is used for two functions. The first function is to convert the control signal read from the buffer into a pulsewidth-modulation (PWM) signal and then send the PWM signal to drive the dc motor. The second function is to encapsulate the motor state into a packet and send it to the PC controller via the network. The local board and the sensor are time driven, and they are synchronized by using the same clock signal. In the experimental verifications, the PC controller and the dc motor were linked by a simulated network. The simulated network creates time delays by delaying the packets transmitted between the PC controller and the dc motor and creates packet losses by discarding the transmitted packets. The reasons for using the simulated network rather than using the actual network are as follows. 1) The experimental results can be compared with the simulation results under the same network condition. 2) The experiment is ensured to be repeatable for various investigations.

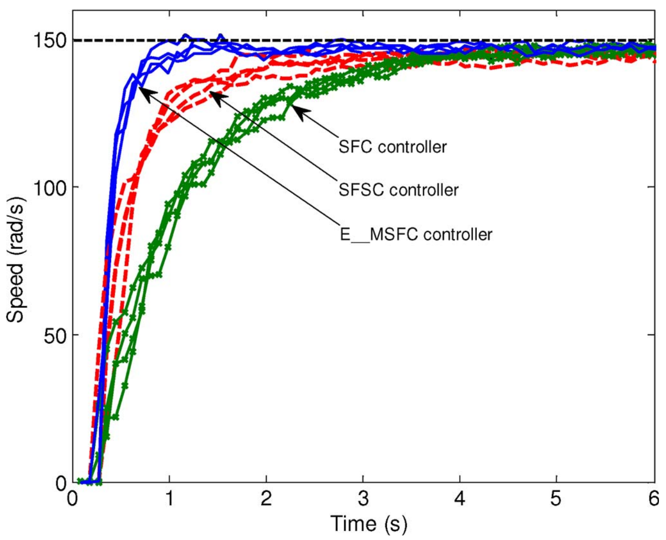


Fig. 7. Typical experimental results using different networked controllers.

With the same initial state and control parameters as used in numerical simulations (see Section V-B), four typical paired experimental results are shown in Fig. 7, where the system responses with SFC, SFSC, and E_MSFC controller are depicted by cross–solid, dashed, and solid lines, respectively. Apparently, the experimental results shown in Fig. 7 are consistent with the simulation results shown in Fig. 5(b), except that there are small steady-state errors in the experimental results. Note that the steady-state errors in the experimental results come as no surprise, since there is an inevitably modeling error, and most importantly, there are inevitably some nonlinearities such as dead zone and fabrication in the experimental apparatus.

From the simulation results shown in Fig. 5(b) and experimental results shown in Fig. 7, we can also see that the system control performance using E_MSFC controller meets the given performance specifications (i.e., $P.O. \leq 3\%$, and $S.T. \leq 2$ s) very well, which demonstrates that E_MSFC controller provides a satisfactory performance. However, as shown in Figs. 5(b) and 7, the $S.T.$ of networked system using SFC and SFSC does not satisfy the given specification $S.T. \leq 2$ s. This phenomenon demonstrates the fact that a stabilizing controller can only guarantee the system to be asymptotically stable while the system control performance may be not satisfied. Note that both system stability and control performance are taken into account in the design stage of E_MSFC. This substantial advantage enables E_MSFC to overcome the aforementioned shortcoming of stabilizing controller.

In summary, the aforementioned simulation and experimental results have demonstrated the effectiveness of the proposed E_MSFC method for real-world application. It has been shown that the E_MSFC method is more general than the ones in [13] and [14] at the application scope, since E_MSFC controller is an output tracking controller while the ones in [13] and [14] are regulators. Furthermore, the E_MSFC method can reduce the system cost, substantially comparing with the methods in [13] and [14], and in the examples used, the E_MSFC method reduces the system cost by more than 78.25%.

VI. Conclusion

This paper has presented an E_MSFC method for networked dc motor system with time delays and packet losses. The proposed E_MSFC controller is an output tracking controller, and therefore, it is more general than the regulator at the application scope. One advantage of E_MSFC method is that both system stability and control performance are considered during the controller design stage; another is that it can be easily implemented to various applications since it has simple structure and has no assumptions on time delay and packet loss models. Simulation and experimental results are given to demonstrate the effectiveness of the proposed approaches.

In future work, we will consider more performance requirements such as H_∞ specification during the controller design stage. Moreover, we will also extend the results to the case where full state measurement is not available.

References

[1] M.-Y. Chow and Y. Tipsuwan, “Gain adaptation of networked DC motor controllers based on QoS variations,” *IEEE Trans. Ind. Electron.*, vol. 50, no. 5, pp. 936–943, Oct. 2003.

[2] K. C. Lee, S. Lee, and M. H. Lee, “QoS-based remote control of networked control systems via Profibus token passing protocol,” *IEEE Trans. Ind. Informat.*, vol. 1, no. 3, pp. 183–191, Aug. 2005.

[3] G.-P. Liu, Y. Xia, J. Chen, D. Rees, and W. Wu, “Networked predictive control of systems with random network delays in both forward and feedback channels,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1282–1297, Jun. 2007.

[4] C.-L. Hwang, L.-J. Chang, and Y.-S. Yu, “Network-based fuzzy decentralized sliding-mode control for car-like mobile robots,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 574–585, Feb. 2007.

[5] P. Seiler and R. Sengupta, “An H_∞ approach to networked control,” *IEEE Trans. Autom. Control*, vol. 50, no. 3, pp. 356–364, Mar. 2005.

[6] M. Gaid, A. Cela, and Y. Hamam, “Optimal integrated control and scheduling of networked control systems with communication constraints: Application to a car suspension system,” *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 4, pp. 776–787, Jul. 2006.

[7] W. J. Kim, K. Ji, and A. Srivastava, “Network-based control with real-time prediction of delayed/lost sensor data,” *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 1, pp. 182–185, Jan. 2006.

[8] J. Xiong and J. Lama, “Stabilization of linear systems over networks with bounded packet loss,” *Automatica*, vol. 43, no. 1, pp. 80–87, Jan. 2007.

[9] P. Martí, J. Yépez, M. Velasco, R. Villà, and J. M. Fuertes, “Managing quality-of-control in network-based control systems by controller and message scheduling co-design,” *IEEE Trans. Ind. Electron.*, vol. 5, no. 6, pp. 1159–1167, Dec. 2004.

[10] M. Long, C.-H. Wu, and J. Y. Hung, “Denial of service attacks on network-based control systems: Impact and mitigation,” *IEEE Trans. Ind. Informat.*, vol. 1, no. 2, pp. 85–96, May 2005.

[11] S. Hu and Q. Zhu, “Stochastic optimal control and analysis of stability of networked control systems with long delay,” *Automatica*, vol. 39, no. 11, pp. 1877–1884, Nov. 2003.

[12] M. Garcia-Rivera and A. Barreiro, “Analysis of networked control systems with drops and variable delays,” *Automatica*, vol. 43, no. 12, pp. 2054–2059, Dec. 2007.

[13] D. Yue, Q.-L. Han, and C. Peng, “State feedback controller design of networked control systems,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 51, no. 11, pp. 640–644, Nov. 2004.

[14] H. Li, M.-Y. Chow, and Z. Sun, “State feedback stabilization of networked control systems,” *IET Control Theory Appl.*, to be published.

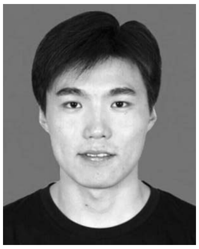
[15] H. Li, M.-Y. Chow, and Z. Sun, “Optimal stabilizing gain selection for networked control systems with time delays and packet losses,” *IEEE Trans. Control Syst. Technol.*, to be published.

[16] H. Li, Z. Sun, B. Chen, and H. Liu, “Intelligent scheduling control of networked control systems with networked-induced delay and packet dropout,” *Int. J. Control, Autom. Syst.*, vol. 6, no. 6, pp. 915–927, 2008.

[17] K. Natori and K. Ohnishi, “A design method of communication disturbance observer for time-delay compensation, taking the dynamic property of network disturbance into account,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 5, pp. 2152–2168, May 2008.

[18] C. Lazar and S. Carari, “A remote-control engineering laboratory,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2368–2375, Jun. 2008.

[19] P. A. Bosman and J. Grahl, “Matching inductive search bias and problem structure in continuous estimation-of-distribution algorithms,” *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1246–1264, Mar. 2008.



Hongbo Li received the B.S. degree from Northeastern University, Shenyang, China, in 2004. He is currently working toward the Ph.D. degree in the Department of Computer Science and Technology, Tsinghua University, Beijing, China.

He is also with the Electrical and Computer Engineering Department, North Carolina State University, Raleigh. His current research interests include networked control systems and intelligent control.

Mo-Yuen Chow (S’81–M’82–SM’93–F’07) received the B.S. degree in electrical and computer engineering from the University of Wisconsin, Madison, in 1982, and the M.Eng. and Ph.D. degrees from Cornell University, Ithaca, NY, in 1983 and 1987, respectively.

He is currently a Professor with the Electrical and Computer Engineering Department, North Carolina State University, Raleigh. His core technology is diagnosis and control, artificial neural networks, and fuzzy logic with applications to areas including motors, power systems, networked control systems, and unmanned vehicles.

Zengqi Sun (SM’93) received the Ph.D. degree in control engineering from Chalmers University of Technology, Goteborg, Sweden, in 1981.

He is currently a Professor with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His current research interests include intelligent control, robotics, fuzzy systems, neural networks, and evolutionary computing.