Support Vector Networks in Adaptive Friction Compensation

G. L. Wang, Y. F. Li, *Senior Member, IEEE*, and D. X. Bi

***Abstract*—This paper presents our research on how support vector regression (SVR) and parametric adaptive learning, which are normally used independently, can be exploited together to benefit adaptive neural control. In the context of friction compensation for servo-motion control systems, we present the notion of support vector networks which play an essential role in combining SVR and adaptive neural network (NN) in cooperation for friction estimation. The analysis shows that the proposed support vector network contributes not only to the performance improvement but also to the practical usefulness in adaptive friction compensation. Experimental results are reported to demonstrate the effectiveness of the proposed approach.**

***Index Terms*—Friction compensation, neural network (NN), servo motion systems, support vector regression (SVR).**

I. Introduction

A. Background and Motivations

NEURAL NETWORK (NN) has been widely used in unknown function approximation needed directly or indirectly in designing adaptive controllers. Much work has been conducted on broadening the class of the systems for which adaptive neural control can be applied [1]. In these studies, the Lyapunov stability theory plays a crucial role of performing the learning rule design with the guaranteed stability and required robustness. However, the approximation property of NNs is only locally applicable and approximation errors always exist. These give rise to instability which challenges the Lyapunov design for adaptive neural control. Among various instability factors, the effects of the architecture and initialization of NNs are significant. First, the approximation accuracy is highly dependent on the NN's structures. In particular, an underdetermined NN can deteriorate the approximation accuracy while an overdetermined NN can lead to heavy computational burden. Second, initializing NNs considerably affects the neural control performance and even the stability of the closed-loop systems. Indeed, the initial estimation errors in the NN weights inevitably result in the transient performance limitation, which may make the adaptive neural controller practically infeasible.

Manuscript received December 1, 2005; revised September 18, 2006; accepted February 5, 2007. This work was supported by the National Science Foundation (NSF) of China under Project 60473120 and by the Research Grants Council of Hong Kong under Project CityU117605. The work of G. L. Wang was supported by the Guangdong Nature Science Foundation (GDNSF) under Project 06023190 and by the Scientific Research Foundation (SRF) for Returned Overseas Chinese Scholars (ROCS), State Education Ministry (SEM) of China.

G. L. Wang is with the School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, P.R. China.

Y. F. Li and D. X. Bi are with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong (e-mail: meyfli@cityu.edu.hk).

Digital Object Identifier 10.1109/TNN.2007.899148

As a linearly parameterized neural network (LPNN), the radial basis function (RBF) networks are a class of the most important NN parameterizations commonly used in adaptive neural control [2]. The construction of an RBF network relies heavily on the conditions and the *a priori* knowledge for the studied systems which are in general unknown. An empirical choice often leads to an over- or underdetermined NN structure. Therefore, it is highly desirable to construct a well-conditioned structure with a satisfactory initialization systematically.

Support vector regression (SVR) was developed in statistical learning theory [3], [4]. From the viewpoint of function approximation, the SVR is regarded as an LPNN, which can serve as a powerful tool of reconstructing a function from sparsely observed data. Of great interest is that SVRs are self-structured without the curse of dimensionality even for a large number of inputs. Additionally, SVR training can be performed by solving a convex optimization problem without local minima solutions. Via structural risk minimization, the generalizing capability of SVR can be guaranteed without overfitting problems. These features suggest that the SVR is a potential alternative for approximating unknown functions in control system design.

In this paper, we present our studies on how SVR and parametric adaptive learning, which are normally used independently, can be exploited together to create a parsimonious NN architecture with satisfactory initialization, which will benefit the design of adaptive NN-based control. We present the proposed approach in the context of friction compensation for servo motion control systems although the method has much wider potential applications. Experimental studies are conducted to demonstrate the effectiveness of the proposed approach.

B. Related Works

The construction of feedforward multilayer neural networks (MNNs) is a well-discussed problem. Nonlinear optimization algorithms are most straightforward approaches to MNN's training but suffer from the problem of local minima. Thus various alternatives have been proposed to facilitate efficient MNN's training. Among these well-developed techniques, additive (or constructive, or growing) methods [5] and subtractive (or destructive, or pruning) methods [6] are well-known strategies which determine the NN's structure through successive refining steps but along two different directions: adding a new relevant neuron in a forward fashion or removing an old irrelevant neuron in a backward fashion, respectively. It has been shown that the combination of these two approaches can offer more efficient solutions [7]. To avoid prior offline learning phases, an online self-structuring MNN has been proposed in the design of adaptive neural controller in [8], in which a simple additive mechanism is introduced to achieve

the structural adaptation. Nevertheless, on one hand, the global property of active neuron functions within the networks makes the convergence of MNN learning slower; on the other hand, the nonlinearity of MNNs makes the design of adaptive neural control difficult and the results obtained more conservative. The support vector networks proposed in this paper are LPNNs, which is complementary to the MNNs in avoiding the shortcomings mentioned previously.

An RBF network is a special MNN with a fixed hidden layer consisting of RBFs. The construction of the RBF networks is reduced to the selection of the center and shape of basis functions, which has triggered much research works grouped as supervised and unsupervised training. The basic idea of the unsupervised techniques is to choose the centers as the templates of the input training data. One of these techniques is the clustering method, such as k-means clustering [9] and vector quantization [10]. The main drawback of these methods is the large number of the resultant hidden centers which is known as the curse of dimensionality. Alternatively, the supervised techniques select the optimal centers successively from the input training data based on minimizing the network output errors in a forward or backward fashion. The orthogonal least square (OLS) algorithm is a powerful approach to the supervised training [11]. Recently, support vector machine (SVM) with the RBF kernel has been used to construct the RBF network for classification, in which the centers are trivially selected as the support vectors [12]. It has been shown that the SVM-based training outperforms the clustering methods. Compared with the OLS methods, the SVM-based training targets at minimizing the bounds of not only the training error but also generalization error and is thus superior in maintaining better NN's generalization level. This paper extends the use of the SVM-based training in three aspects. First, the SVR-based training is used for unknown function approximation in control systems. Second, the constructed LPNNs are no longer limited to the RBF networks via using available kernel functions other than RBFs. Third, the SVM-based training is incorporated into parametric adaptive learning to enhance the performance of adaptive neural control.

Friction compensation plays an important role in the controller design for a servo motion system. Friction is often treated as model uncertainty because it is a very complex nonlinear phenomenon hard to be described analytically. There have been extensive efforts on developing adaptive friction compensation schemes, where parametric uncertainties are mainly dealt with in a variety of empirical models [13]. Recently, NN parameterizations have aroused research interests in model-free adaptive friction compensation [14]–[18]. However, these studies mainly focus on the adaptation and stability of adaptive compensators without special considerations on the construction of NNs. In [19], SVR parametrization has been developed for friction modeling. It has been shown that there is no need to seek complex sensing techniques to collect the training data and thus the SVR parametrization can be easily implemented for a servo motion system such as a haptic display system [20]. The new contribution of this paper is to present the notion of support vector networks which play an essential role in combining SVR parametrization and adaptive NN in cooperation for friction estimation. In addition, an analysis is presented to show how the proposed support vector networks can improve the performance and enhance the practical usefulness in friction estimation.

This paper is organized as follows. First, our methodology is outlined in Section II. Then, main results are presented in Section III. Finally, the experimental results are reported to validate our approach.

II. Problem Formulation

Consider a 1-degree-of-freedom (1-DOF) planar robot manipulator described in the following standard form

$$I\ddot{q} + F = \tau \tag{1}$$

with $q \in \mathbb{R}$ denoting the joint position, $I > 0$ being the inertia of the link, F representing the friction term, and τ being the torque supplied by the joint actuator. To focus on the uncertainty issue of the friction effect, we assume that I is known. Let q_d be the joint pose reference trajectory. The goal is to drive the system to track the reference trajectory. Define the tracking error as $e = q_d - q$. It is convenient to introduce the filtered tracking error as

$$r = \lambda e + \dot{e} \tag{2}$$

where $\lambda > 0$ is a design parameter. In terms of the filtered error r, the dynamic system (1) can be rewritten as

$$I\dot{r} = I(\lambda\dot{e} + \ddot{q}_d) + F - \tau \tag{3}$$

which motivates the following approximation-based control scheme [21]

$$\tau = k_r r + I(\lambda\dot{e} + \ddot{q}_d) + \hat{F} \tag{4}$$

where $k_r > 0$ is another design parameter and $\hat{F}$ is the friction estimator and plays the role of compensating the friction effect. As such, the closed-loop system is asymptotically stable if F is exactly known or $\hat{F} = F$.

However, F is generally unknown or incompletely known. The approximation-based scheme (4) yields the closed-loop filtered tracking error system

$$I\dot{r} + k_r r = \tilde{F} \tag{5}$$

where $\tilde{F} = F - \hat{F}$ is the friction estimation error. Thus, the friction estimator affects the performance.

In view of the fact that the system operates in a bounded region, we trivially assume that $[q, \omega]^T$ evolves in a compact subset $\mathcal{Q} \times \bar{\Omega} \subset \mathbb{R}^2$ and $(q_d, \dot{q}_d) \in \mathcal{Q}_d \times \bar{\Omega}_d$ with $\mathcal{Q}_d \times \bar{\Omega}_d$ being a connected subset of $\mathcal{Q} \times \bar{\Omega}$. Let $\Omega_+ = \{\omega \in \bar{\Omega} | \omega \geq 0\}$ and $\Omega_- = \{\omega \in \bar{\Omega} | \omega \leq 0\}$ denote the positive- and negative-velocity regimes, and the subscripts "+" and "−" indicate the related friction components, respectively.

In this paper, the friction is modeled as a function of the angular velocity $\omega = \dot{q}$ in the form of

$$F(\omega) = f_+(\omega)\mu_+(\omega) + f_-(\omega)\mu_-(\omega) \tag{6}$$

where $\mu_+(\omega) = 1$ if $\omega \geq 0$; otherwise, $\mu_+(\omega) = 0$ and $\mu_-(\omega) = \mu_+(-\omega)$. In addition, $f_+(\omega)$ and $f_-(\omega)$ are assumed to be smooth functions over Ω_+ and Ω_-, respectively.

Our methodology is outlined as follows. It consists of two parts. The first part is to construct the SVRs of $f_{\pm}(\omega)$ (brevity for $f_{+}(\omega)$ and $f_{-}(\omega)$) based on training data sets $\mathcal{D}_{\pm} \subset \Omega_{\pm}$, that is

$$\hat{f}_{\pm}(\omega) = \hat{\boldsymbol{c}}_{\pm}^{T}\boldsymbol{\phi}_{\pm}(\omega) \tag{7}$$

where $\hat{\boldsymbol{c}}_{\pm} \in \mathbb{R}^{N_{\pm}}$, $\boldsymbol{\phi}_{\pm}(\omega) \in \mathbb{R}^{N_{\pm}}$, and $N_{\pm} > 0$ are the numbers of the support vectors in $\mathcal{D}_{\pm}$, respectively. This will be done in an offline fashion. With the previous notations, the friction effect captured via the resultant SVRs is represented in the compact form of

$$F_{\rm sv}(\omega) = \hat{\boldsymbol{c}}_{\rm sv}^{T}\boldsymbol{\psi}(\omega) \tag{8}$$

where $\boldsymbol{\psi}(\omega) = [\boldsymbol{\phi}_{+}^{T}(\omega)\mu_{+}(\omega), \boldsymbol{\phi}_{-}^{T}(\omega)\mu_{-}(\omega)]^{T} \in \mathbb{R}^{\bar{N}}$, $\hat{\boldsymbol{c}}_{\rm sv} = [\hat{\boldsymbol{c}}_{+}^{T}, \hat{\boldsymbol{c}}_{-}^{T}]^{T} \in \mathbb{R}^{\bar{N}}$, and $\bar{N} = N_{+} + N_{-}$. Let $l(\omega) = F(\omega) - F_{\rm sv}(\omega)$. $l(\omega)$ reflects the residual friction effect over $\bar{\Omega}$ when using $F_{\rm sv}(\omega)$ as an estimate of $F(\omega)$. The second part is to estimate $l(\omega)$ via the online NN-based approximation under the adaptive control framework. This NN inherits the structure of the SVRs and is put in the form of

$$l_{\rm ad}(\omega) = \hat{\boldsymbol{v}}_{\rm ad}^{T}\boldsymbol{\psi}(\omega) = \begin{cases} \hat{\boldsymbol{v}}_{+}^{T}\boldsymbol{\phi}_{+}(\omega), & \omega \in \Omega_{+} \\ \hat{\boldsymbol{v}}_{-}^{T}\boldsymbol{\phi}_{-}(\omega), & \omega \in \Omega_{-} \end{cases} \tag{9}$$

where $\hat{\boldsymbol{v}}_{\rm ad} = [\hat{\boldsymbol{v}}_{+}^{T}, \hat{\boldsymbol{v}}_{-}^{T}]^{T}$ is the estimate of the ideal NN weight vector $\boldsymbol{v}_{\rm ad}^{*}$ as

$$\boldsymbol{v}_{\rm ad}^{*} := \arg\min_{\boldsymbol{v}_{\rm ad} \in \mathbb{R}^{\bar{N}}} \left\{ \sup_{\omega \in \bar{\Omega}} \left| \boldsymbol{v}_{\rm ad}^{T}\boldsymbol{\psi}(\omega) - l(\omega) \right| \right\}. \tag{10}$$

As a new paradigm of friction reconstruction, $F(\omega)$ can be approximated by

$$\hat{F}(\omega) = F_{\rm sv}(\omega) + l_{\rm ad}(\omega) = \left(\hat{\boldsymbol{c}}_{\rm sv}^{T} + \hat{\boldsymbol{v}}_{\rm ad}^{T} \right) \boldsymbol{\psi}(\omega) \tag{11}$$

which is referred to as the support vector network.

Indeed, the support vector network (11) incorporates SVR with adaptive NN. The motivations behind this are highlighted by the following remarks.

Remark 1: In addition to the offline SVR $F_{\rm sv}(\omega)$, the support vector network involves online NN approximation $l_{\rm ad}(\omega)$ of the residual friction effect $l(\omega) = F(\omega) - F_{\rm sv}(\omega)$, which can enhance the friction estimation ability, especially, when friction varies. As a matter of fact, the friction behavior is often affected by various factors such as temperature and humidity, and thus, it usually changes with time. The term $l(\omega)$ tends to be significant and the adaptive effort $l_{\rm ad}(\omega)$ becomes necessary to compensate for it. In short, the adaptive NN complements the SVR with the online learning ability in friction compensation.

III. Support Vector Network for Adaptive Friction Compensation

A. Structure and Initialization

In what follows, we show how to construct and initialize a support vector network, i.e., how to determine $\boldsymbol{\psi}(\omega)$ and $\hat{\boldsymbol{c}}_{\rm sv}$. The basic idea is to construct the approximation of $f_{\pm}(\omega)$ from observed data. The SVR-based method developed in [19] is suitable for this task. We briefly describe the SVR formulation in the context of the friction model structure (6). For clarity, we omit the subscripts "$\pm$" without confusion.

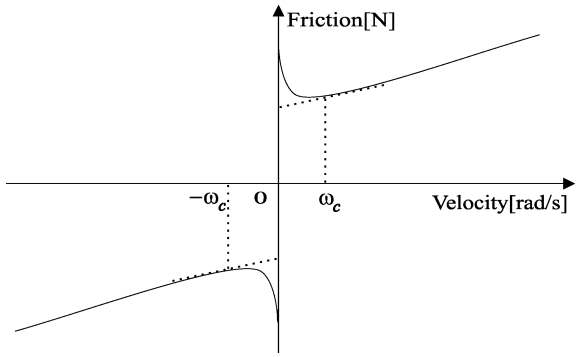

Fig. 1. Stribeck effect of static friction behavior.

No assumptions are made on the shape of $f(\omega)$ except that it has the downward bend behavior in the low-velocity regime, the so-called Stribeck effect as shown in Fig. 1, which has been well studied in [22]. In particular, $f(\omega)$ is assumed to be highly nonlinear and much smooth in the low-velocity regime. Beyond some cutoff critical velocity ω_c, the Stribeck effect almost vanishes and $f(\omega)$ is nearly linear in the high-velocity regime. We assume that there is a finite amount of available training data $\mathcal{D} = \{(\omega_j, F_j) | \omega_j \in \Omega\}_{j=1}^{M}$, where $F_j = f(\omega_j) + \nu_j$ for each j, ν_j is the measurement error of the jth training point, and M is the number of the training points. According to the cutoff critical velocity ω_c, we can subgroup the indices of the training points as $I_1 = \{i \in I_0 | \omega_i \le \omega_c\}$ and $I_2 = I_0 - I_1$, where $I_0 = \{1, 2, \ldots, M\}$. Motivated by the downward bend behavior of the Stribeck effect, the straightforward selection of ω_c is determined by

$$\omega_c =: \omega_{i_c}, \qquad i_c = \arg\min_{j} |F_j|. \tag{12}$$

As in [19], $f(\omega)$ is approximated underlying the training data set $\mathcal{D}$ by

$$\hat{f}(\omega) = \sum_{j=1}^{M} \alpha_j K(\omega_j, \omega) \tag{13}$$

with $K(\cdot, \cdot)$ being a kernel function defined over $\Omega \times \Omega$. This representation is motivated by seeking a solution of the following regularization problem:

$$\min_{\bar{f} \in \mathcal{F}_K} \sum_{l=1}^{2} C_l \sum_{j \in I_l} \left| F_j - \bar{f}(\omega_j) \right|_{\epsilon_l} + \frac{1}{2} \langle \bar{f}, \bar{f} \rangle_{\mathcal{F}_K} \tag{14}$$

in a reproducing kernel Hilbert space (RKHS) $\mathcal{F}_K$ defined by kernel K [23]. Here, $\langle \cdot, \cdot \rangle_{\mathcal{F}_K}$ denotes the scalar inner product of $\mathcal{F}_k$. $|\cdot|_{\epsilon}$ is Vapnik's ϵ-insensitive norm defined as $|\bar{F} - \bar{f}(\omega)|_{\epsilon} = \max\{0, |\bar{F} - \bar{f}(\omega)| - \epsilon\}$. In the context of the regularization theory [24], the first term in (14) is viewed as a stabilizer that reflects smoothing effort, while others totally describe the cost of the training error. C_ls are the parameters that make the tradeoffs between the training errors and the smoothing constraints.

Each ϵ_l provides a degree of freedom associated with the error tolerance control.

The regularization problem (14) can be transformed into the following equivalent problem:

$$\min_{\bar{f},\boldsymbol{\xi},\boldsymbol{\xi}^*} \sum_{l=1}^{2} C_l \sum_{j\in I_l} \left(\xi_j + \xi_j^*\right) + \frac{1}{2}\langle \bar{f}, \bar{f}\rangle_{\mathcal{F}_K} \tag{15}$$

subject to the constraints

$$\begin{aligned} \bar{f}(\omega_j) - F_j &\le \epsilon_l + \xi_j, \qquad j \in I_l; \qquad l = 1,2 \\ F_j - \bar{f}(\omega_j) &\le \epsilon_l + \xi_j^*, \qquad j \in I_l; \qquad l = 1,2 \\ \xi_j, \xi_j^* &\ge 0, \qquad j = 1,2,\ldots,M. \end{aligned} \tag{16}$$

$\boldsymbol{\xi}$ and $\boldsymbol{\xi}^*$ in (15) denote the slack vectors $[\xi_1, \xi_2, \ldots, \xi_M]^T$ and $[\xi_1^*, \xi_2^*, \ldots, \xi_M^*]^T$, respectively. This constrained optimal problem can be solved by the standard technique of Lagrangian multipliers. As a result, the vector of the coefficients $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_M]^T$ is the solution of the following quadratic program (QP) problem [19]:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{j,k=1}^{M} \alpha_j \alpha_k K(\omega_j, \omega_k) - \sum_{j=1}^{M} \alpha_j F_j + \sum_{l=1}^{2} \epsilon_l \sum_{j\in I_l} |\alpha_j| \tag{17}$$

subject to

$$\begin{aligned} |\alpha_j| &\le C_l, \qquad j \in I_l; \qquad l = 1,2 \\ \sum_{j=1}^{M} \alpha_j &= 0. \end{aligned} \tag{18}$$

Remark 2: Compared with the standard SVR [3], our augmented formulation (17) uses different regularization parameters and the ϵ-insensitive levels. The pairs (C_1, ϵ_1) and (C_2, ϵ_2) correspond to the smoothing efforts and the error-tolerances imposed on $\hat{f}(\omega)$ in the low- and high-velocity regimes, respectively. The logic behind this is to impose varying constraints of the smoothness and the error-tolerance on $\hat{f}(\omega)$ according to the different complexities of friction forces in the low- and high-velocity regimes.

Of special interest is the Karush–Kuhn–Tucker (KKT) conditions satisfied at the solution $\boldsymbol{\alpha}$, that is, for $j \in I_l, l = 1,2$

$$\alpha_j \left(\epsilon_l + \xi_j - F_j + f(\omega_j)\right) = 0, \qquad \alpha_j < 0 \tag{19a}$$

$$\alpha_j \left(\epsilon_l + \xi_j^* + F_j - f(\omega_j)\right) = 0, \qquad \alpha_j > 0 \tag{19b}$$

$$\left(C_l - |\alpha_j|\right)\xi_j = 0 \quad \left(C_l - |\alpha_j|\right)\xi_j^* = 0. \tag{19c}$$

Note that the KKT conditions imply the sparsity of the SVR parametrization. The training points which have nonzero coefficients α_j are the so-called support vectors, usually only a small fraction of the training data set $\mathcal{D}$. Indeed, nonzero α_j can be categorized as two groups. One consists of all the bounded ones, $|\alpha_j| = C_l$; the other involves ones $0 < |\alpha_j| < C_l$, which correspond to the points at the boundary $|F_j - f(\omega_j)| - \epsilon_l = 0$.

We summarize the procedure of constructing an SVR parametrization as follows:

Step 1) collecting the training data $\mathcal{D} = \{(\omega_j, F_j)\}_{j=1}^{M}$;
Step 2) determining ω_c, and then, collecting I_1 and I_2;
Step 3) specifying the kernel K and the pairs $\{(C_l, \epsilon_l)\}_{l=1}^{2}$;
Step 4) solving the QP problem for $\boldsymbol{\alpha}$.

With the computed $\boldsymbol{\alpha}$ by implementing the previous SVR training procedure, the SVR (13) is readily available for structuring and initializing a support vector network. Let $I_{\text{sv}} = \{i \in I_0 | \alpha_i > 0\}$ be the set of the support vector indices and $\mathcal{D}_{\text{sv}} = \{(\omega_{i_j}, F_{i_j}) | i_j \in I_{\text{sv}}\}_{j=1}^{N} \subset \mathcal{D}$ the support vector set, where N is the number of the support vectors. Accordingly, the representation (13) can be rewritten as the support vector expansion (8) with $\hat{c}_j = \alpha_{i_j}$ and $\phi_j(\omega) = K(\omega_{i_j}, \omega)$ for $j = 1, 2, \ldots, N$.

At this stage, we make the following assumption and remark:

Assumption: The matrix $[K(\omega_{i_j}, \omega_{i_k})]$ for the support vector set $\mathcal{D}_{\text{sv}}$ is strictly positive definite.

Remark 3: This assumption trivially holds for commonly used kernels including the spline kernel in the sequel [19].

B. Adaptation and Stability

This part will show how to equip the support vector network with the adaptation ability as well as the guaranteed stability. In particular, substituting (11) into (4) yields the augmented control scheme

$$\tau = k_r r + I(\lambda \dot{e} + \ddot{q}_d) + \left(\hat{\boldsymbol{c}}_{\text{sv}}^T + \hat{\boldsymbol{v}}_{\text{ad}}^T\right)\boldsymbol{\psi} \tag{20}$$

which needs the adaptive rule for estimating the NN weights $\hat{\boldsymbol{v}}_{\text{ad}}$, while the closed-loop system meets the satisfactory stability requirement. The Lyapunov-based adaptive technique is utilized to achieve this goal. First, the update rule of the NN and its stability is conditionally developed. Then, we present an analytical analysis which plays a twofold role. One is to clarify the relationship between the transient performance and the design parameters. The other is to identify the initial conditions which guarantee the stability of the tracking control.

With the notation of the support vector network defined in (11), the following approximation holds:

$$F(\omega) = \left(\hat{\boldsymbol{c}}_{\text{sv}}^T + \boldsymbol{v}_{\text{ad}}^{*\,T}\right)\boldsymbol{\psi}(\omega) + \epsilon(\omega), \qquad \omega \in \bar{\Omega} \tag{21}$$

where $\epsilon(\omega)$ is the approximation error with the upper bound $|\epsilon(\omega)| \le \epsilon_m$ for some $\epsilon_m > 0$. Accordingly, the friction estimation error $\tilde{F}(\omega)$ in (5) can be written as

$$\tilde{F}(\omega) = \tilde{\boldsymbol{v}}_{\text{ad}}^T \boldsymbol{\psi}(\omega) + \epsilon(\omega), \qquad \omega \in \bar{\Omega} \tag{22}$$

where $\tilde{\boldsymbol{v}}_{\text{ad}} = \boldsymbol{v}_{\text{ad}}^* - \hat{\boldsymbol{v}}_{\text{ad}}$. To motivate the update rule of $\hat{\boldsymbol{v}}_{\text{ad}}$, consider the following Lyapunov function candidate

$$V = \frac{1}{2} I r^2 + \frac{1}{2} \tilde{\boldsymbol{v}}_{\text{ad}}^T \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{v}}_{\text{ad}} \tag{23}$$

where $\boldsymbol{\Gamma}$ is some positive-definite matrix. If ω is supposed to evolve in $\bar{\Omega}$ for all $t \ge 0$ (we will show later that this is true if some initial conditions are necessarily met and the design parameters are appropriately chosen), the time-derivative of V along the state subject to (5) is given by

$$\begin{aligned} \dot{V} &= \left[\tilde{F}(\omega) - k_r r\right] r + \tilde{\boldsymbol{v}}_{\text{ad}}^T \boldsymbol{\Gamma}^{-1} \dot{\hat{\boldsymbol{v}}}_{\text{ad}} \\ &= -k_r r^2 + \tilde{\boldsymbol{v}}_{\text{ad}}^T \left[\boldsymbol{\psi}(\omega) r - \boldsymbol{\Gamma}^{-1} \dot{\hat{\boldsymbol{v}}}_{\text{ad}}\right] + r\epsilon(\omega). \end{aligned} \tag{24}$$

To guarantee the stability of V, we employ the update rule

$$\dot{\hat{\boldsymbol{v}}}_{\text{ad}} = \boldsymbol{\Gamma}\left[\boldsymbol{\psi}(\omega) r - \sigma \hat{\boldsymbol{v}}_{\text{ad}}\right] \tag{25}$$

where $\sigma > 0$ is a constant. Indeed, using the update rule (25) in (24) yields

$$\begin{aligned}\dot{V} &= -k_r r^2 - \sigma\|\tilde{\boldsymbol{v}}_{\rm ad}\|^2 + \sigma\tilde{\boldsymbol{v}}_{\rm ad}^T\boldsymbol{v}_{\rm ad}^* + \epsilon(\omega)r \\ &\le -\frac{k_r}{2}r^2 - \frac{\sigma}{2}\|\tilde{\boldsymbol{v}}_{\rm ad}\|^2 + \frac{\sigma}{2}\|\boldsymbol{v}_{\rm ad}^*\|^2 + \frac{2}{k_r}\epsilon_m^2 \end{aligned} \quad (26)$$

in which the last step is derived by using the inequality $ab \le \beta a^2 + \beta^{-1}b^2$ ($\beta > 0$). Note that $\|\tilde{\boldsymbol{v}}_{\rm ad}\|^2 \le \gamma\tilde{\boldsymbol{v}}_{\rm ad}^T\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{v}}_{\rm ad}$, where $\gamma > 0$ is the smallest eigenvalue of $\boldsymbol{\Gamma}$. Let

$$\rho_0 = \min(k_r I^{-1}, \sigma\gamma) \quad \rho_1 = \frac{\sigma}{2}\|\boldsymbol{v}_{\rm ad}^*\|^2 + \frac{2}{k_r}\epsilon_m^2. \quad (27)$$

Thus, we have

$$\dot{V} \le -\rho_0 V + \rho_1.$$

Hence

$$V(t) \le \frac{\rho_1}{\rho_0} + \left[V(0) - \frac{\rho_1}{\rho_0}\right]e^{-\rho_0 t}, \qquad t \ge 0. \quad (28)$$

This implies that r and $\tilde{\boldsymbol{v}}_{\rm ad}$ are uniformly ultimately bounded.

In what follows, we show that there exist some initial conditions for $[q, \omega]^T$ and appropriate design parameters $k_r > 0$, $\gamma > 0$, and $\sigma > 0$ such that $[q, \omega]^T$ always remains in $\mathcal{Q} \times \bar{\Omega}$ for all $t \ge 0$. To do so, we define

$$B_q = \sup\left\{B \big| \sup_{t\ge 0}|q_d(t) - q(t)| \le B \text{ implies } q(t) \in \mathcal{Q}, \forall t\right\}$$
$$B_\omega = \sup\left\{B \big| \sup_{t\ge 0}|\dot{q}_d(t) - \omega(t)| \le B \text{ implies } \omega(t) \in \bar{\Omega}, \forall t\right\}.$$

From (28), for all $t \ge 0$, we have

$$\begin{aligned}|r(t)| &\le \sqrt{\left[\frac{\rho_1}{\rho_0} + V(0)\right]\frac{2}{I}} \\ &\le \sqrt{\left[\frac{\rho_1}{\rho_0} + \frac{1}{2\gamma}\|\tilde{\boldsymbol{v}}_{\rm ad}(0)\|^2\right]\frac{2}{I}} + |r(0)| \\ &:= R_0\left[k_r, \sigma, \gamma, \tilde{\boldsymbol{v}}_{\rm ad}(0)\right] + |r(0)|. \end{aligned} \quad (29)$$

For brevity, we write $R_0[k_r, \sigma, \gamma, \tilde{\boldsymbol{v}}_{\rm ad}(0)]$ as R_0. From (2), we get

$$\begin{aligned}|e(t)| &\le e^{-\lambda t}|e(0)| + \int_0^t e^{-\lambda(t-\mu)}|r(\mu)|\,{\rm d}\mu \\ &\le |e(0)| + [R_0 + |r(0)|]\lambda^{-1} \\ &\le 2|e(0)| + |\dot{e}(0)|\lambda^{-1} + R_0\lambda^{-1}. \end{aligned} \quad (30)$$

Also note that

$$\begin{aligned}|\dot{e}(t)| &\le |r(t)| + \lambda|e(t)| \\ &\le 2[R_0 + |r(0)|] + \lambda|e(0)| \\ &\le 3\lambda|e(0)| + 2|\dot{e}(0)| + 2R_0. \end{aligned} \quad (31)$$

Let $B_q^0 = 2B_q - B_\omega\lambda^{-1}$ and $B_\omega^0 = 2B_\omega - 3\lambda B_q - R_0$. For $B_q^0 > 0$ and $B_\omega^0 > 0$, we can define

$$\begin{aligned}\mathcal{Q}_0 &= \left\{q(0) \in \mathcal{Q} \big| |q_d(0) - q(0)| \le B_q^0\right\} \subset \mathcal{Q} \\ \bar{\Omega}_0 &= \left\{\omega(0) \in \bar{\Omega} \big| |\omega_d(0) - \omega(0)| \le B_\omega^0\right\} \subset \bar{\Omega}. \end{aligned} \quad (32)$$

To meet the constraints $B_q^0 > 0$ and $B_\omega^0 > 0$, we should set the design parameter λ such that

$$\frac{B_\omega}{2B_q} < \lambda < \frac{2B_\omega}{3B_q} \quad (33)$$

and choose $k_r^* > 0$ and $\gamma^* > 0$ such that

$$R_0\left[k_r^*, \sigma, \gamma^*, \tilde{\boldsymbol{v}}_{\rm ad}(0)\right] \le 2B_\omega - 3\lambda B_q. \quad (34)$$

Then, given $[q(0), \omega(0)]^T \in \mathcal{Q}_0 \times \bar{\Omega}_0$, with λ satisfying (33), for $k_r > k_r^*$ and $\gamma > \gamma^*$, it can be easily verified that $|e(t)| \le B_q$ and $|\dot{e}(t)| \le B_\omega$ for all $t \ge 0$, thus $[q, \omega]^T$ always stays in $\mathcal{Q} \times \bar{\Omega}$.

At this stage, we summarize the previous discussions to guide the friction compensator design as follows:

Step 1) determine λ such that (33) holds;
Step 2) choose k_r^* and γ^* such that (34) holds;
Step 3) specify the design parameters $k_r > k_r^*$ and $\gamma > \gamma^*$;
Step 4) identify the initial conditions $\mathcal{Q}_0 \times \bar{\Omega}_0$ via (32).

As a consequence, the stability of the closed-loop system can be characterized as follows.

Theorem: For given $[q_d, \omega_d]^T \in \mathcal{Q} \times \bar{\Omega}$, if the controller (20) determined based on steps 1)–3) is applied to the plant (1), and $[q(0), \omega(0)]^T \in \mathcal{Q}_0 \times \bar{\Omega}_0$ defined by step 4), then $[q(t), \omega(t)]^T \in \mathcal{Q} \times \bar{\Omega}$ for all $t \ge 0$.

C. Discussions

This part will present an analysis that shows the advantages of the proposed method. This focuses on the role for the SVR to play in two aspects. First, the contribution to the performance improvement will be examined. Second, the enhancement of the practical usefulness will be highlighted.

Among various characteristics of tracking performance, two most important criteria are considered here. One is the steady-state tracking error defined by $\lim_{t\to\infty}|e(t)|$. From (28), we see that $\lim_{t\to\infty} V(t) \le \rho_1\rho_0^{-1}$, thus $\lim_{t\to\infty}|r(t)| \le \sqrt{2\rho_1(\rho_0 I)^{-1}} := R_1$. Furthermore, it can be derived as done for (29) that

$$\lim_{t\to\infty}|e(t)| \le \lim_{t\to\infty}|r(t)|\lambda^{-1} \le R_1\lambda^{-1}. \quad (35)$$

Another criterion is the maximum tracking error, that is, $\sup_{t\ge 0}|e(t)|$, bounded by

$$\sup_{t\ge 0}|e(t)| \le 2|e(0)| + |\dot{e}(0)|\lambda^{-1} + R_0\lambda^{-1} \quad (36)$$

which is inferred from (29). We can conclude from (35) and (36) that the smaller R_0 and R_1 are, the better the tracking performance is.

Recalling the theorem, the initial conditions impose some fundamental limitations on the practical implementation of tracking control. Observing (34), we see that the smaller R_0 is, the more relaxed the initial conditions are.

It is evident that the low values of $(\epsilon_m, \boldsymbol{v}^*_{\rm ad}, \tilde{\boldsymbol{v}}_{\rm ad}(0))$ can limit the level of R_0 and R_1, thus they can not only lead to a better tracking performance but also increase the practical usefulness without requiring large k_r, λ, and γ. As a consequence, the high-gain control can be avoided to improve the performance and to enhance the usefulness. The following observations argue that the SVR plays a role in limiting the levels of the triple $(\epsilon_m, \boldsymbol{v}^*_{\rm ad}, \tilde{\boldsymbol{v}}_{\rm ad}(0))$.

Observation 1: According to the universal approximation capability of SVR [25], the SVR can generate a well-conditioned NN structure automatically whenever well-shaped training data are available. Compared with other techniques for constructing NNs, the SVR-based training targets at minimizing the bounds of not only the training error but also the generalization error and is thus superior in practically limiting the level of ϵ_m.

Observation 2: The mechanism of the SVR training can effectively limit the level of $\boldsymbol{v}^*_{\rm ad} = [\boldsymbol{v}^{*T}_+, \boldsymbol{v}^{*T}_-]^T$. To understand this, in terms of $l(\omega)$, we rewrite (21) as

$$\boldsymbol{v}^{*T}_{\rm ad}\boldsymbol{\psi}(\omega) = l(\omega) - \epsilon(\omega). \tag{37}$$

Consider (37) over $\mathcal{D}_{\rm sv}$ (again, the subscript "$\pm$" is omitted) and put them in the compact form of

$$\boldsymbol{v}^{*T}\boldsymbol{K} = \boldsymbol{\iota} - \boldsymbol{\Upsilon} \tag{38}$$

where $\boldsymbol{K} = [K(\omega_{i_j}, \omega_{i_k})]$, $\boldsymbol{\iota} = [l(\omega_{i_1}), l(\omega_{i_2}), \ldots, l(\omega_{i_N})]^T$, and $\boldsymbol{\Upsilon} = [\epsilon(\omega_{i_1}), \epsilon(\omega_{i_2}), \ldots, \epsilon(\omega_{i_N})]^T$. It is followed from the KKT conditions (19) that for $(\omega_{i_j}, F_{i_j}) \in \mathcal{D}_{\rm sv}$

$$\left|l\left(\omega_{i_j}\right)\right| \le \epsilon_l + \max\left\{\xi_{i_j}, \xi^*_{i_j}\right\}, \qquad i_j \in I_l \cap I_{\rm sv}.$$

Let κ be the largest eigenvalue of $\boldsymbol{K}^{-1}$ in (38). Then, we have

$$\|\boldsymbol{v}^*\| \le \kappa\left(\sqrt{\sum_{l=1}^{2}\sum_{i_j \in I_l \cap I_{\rm sv}}\left(\epsilon_l + \max\left\{\xi_{i_j}, \xi^*_{i_j}\right\}\right)^2 + N\epsilon_m}\right). \tag{39}$$

Recall the implementation of the SVR training; the parameters ϵ_ls are the design parameters chosen for the tolerance errors which thus have limited values. The ξ_{i_j}s and $\xi^*_{i_j}$s are the slack terms which contribute to the cost minimized by SVR. Therefore, the first term on the right-hand side of (39) can be limited. According to Observation 1, the second term can also be at some limited level. We can conclude that the SVR training can limit the level of $\boldsymbol{v}^*$ (or $\boldsymbol{v}^*_\pm$) thus $\boldsymbol{v}^*_{\rm ad}$.

Observation 3: Motivated by Observation 2, we simply set $\hat{\boldsymbol{v}}_{\rm ad}(0) = 0$. Consequently, $\tilde{\boldsymbol{v}}_{\rm ad}(0) = \boldsymbol{v}^*_{\rm ad}$, which can be put at a limited level as addressed in Observation 2.

Remark 4: Of special interest is the fact that with a parsimonious structure $\boldsymbol{\psi}(\omega)$ as well as a satisfactory initialization $\hat{\boldsymbol{c}}_{\rm sv}$ offered by SVR the support vector network can guarantee the well-shaped transient behavior and augment the practical feasibility of the design of the adaptive neural control.

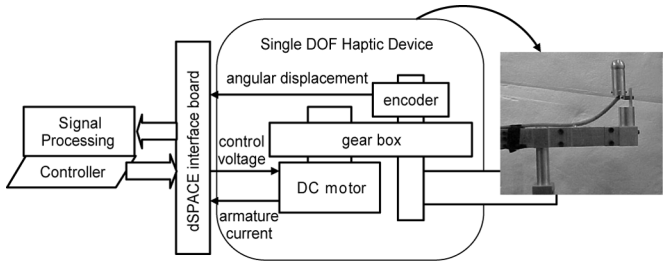

Fig. 2. Schematic diagram of experimental test bed.

IV. Experimental Study

The proposed support vector network was experimentally implemented on a force display device designed in our laboratory. In this paper, we focus on the free-space motion control for tracking a reference trajectory without involving force interaction with the environment. Fig. 2 is a schematic diagram of our test bed. This device is a typical planar 1-DOF rotational link with a vertical joint connected to a direct current (dc) motor through a gearbox with a ratio of $1:80$. The moment of inertial of the link is $I = 2.438 \times 10^{-2}$ kg·m^2. The joint angular displacement is measured by an optical encoder with a resolution of 500 pulses per revolution. The armature current of the motor is measured via a Hall sensor. With the known torque constant 19.5 N·m/A, the applied torque is acquired experimentally through measuring the armature current. These two measured signals are fed into an incremental sensor interface and an analog-to-digital (A/D) converter on an dSPACE (PS1103 PPC) controller board, respectively. The control signal is generated through an digital-to-analog (D/A) converter on the dSPACE card and is amplified by a power amplifier module. A PC PIII/860 MHz is used as the host computer for signal processing and control.

Two tracking tasks of interest are performed here: nonunidirectional and unidirectional positioning. The corresponding reference trajectories to be tracked are shown in Fig. 3. To understand the improvement in trajectory tracking performance due to the use of support vector networks, the following two scenarios for compensating the friction are studied for each tracking task: 1) with only the SVR and 2) with the proposed support vector network. Performance comparisons are made between the control schemes associated with these two scenarios. Two aspects are evaluated. The first one is the efficiency of the SVR training in structuring and initializing a support vector network, while the second is the effectiveness of the adaptive capability of the support vector network.

The experimental implementation is presented as follows.

A. SVR Versus Tustin Friction Model

We followed the SVR learning procedure described at the end of Section III-A to perform the SVR construction.

First, as done in [19], the training data were collected. We selected 54 training samples over the positive- and negative-velocity regime, respectively. Next, based on the algorithm in (12), the cutoff critical values ω_cs were determined. Thus, I_ls

TABLE I
TRAINING DATA AND TRAINING/MODELING RESULTS

velocity regime	training data $\lvert\omega_c\rvert$	training data $\lvert I_1\rvert/\lvert I_2\rvert$	SVR Training SV(N/M)	Tustin friction model $(f_c, f_s, f_\omega, \omega_s)$
positive	0.781	44/10	5/54(9.3%)	$(0.0122, 0.0179, 0.0002, 0.2)$
negative	1.221	49/5	4/54(7.4%)	$(-0.0144, -0.0199, -0.0011, 0.4)$

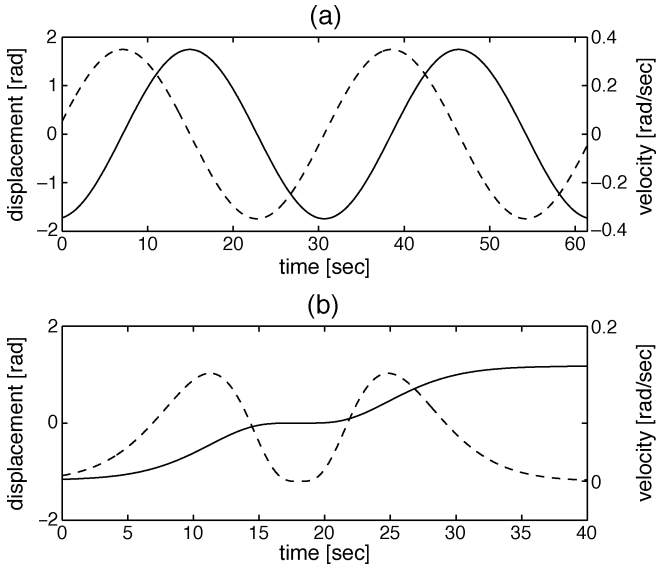

Fig. 3. Reference trajectories: solid line—angular displacement; dashed line—angular velocity. (a) Nonunidirectional movement. (b) Unidirectional movement.

are well defined. The results are listed in Table I, in which the notation $|I_l|$ denotes the numbers of the elements in I_l.

In our experimental studies, the spline kernel

$$K(\omega_1,\omega_2) = 1 + \omega_1\omega_2 + \frac{1}{2}|\omega_1-\omega_2|\min(\omega_1,\omega_2)^2 + \frac{1}{6}\min(\omega_1,\omega_2)^3 \quad (40)$$

was used in the SVR training. It was experimentally observed that this spline kernel is preferable to other kernel functions due to the lower complexity and the better generalization capability of the resultant SVR [20]. We take $(C_1,\epsilon_1) = (100, 0.0003)$ and $(C_2,\epsilon_2) = (1, 0.00065)$. These selections are mainly guided by the *a priori* knowledge of the static friction behavior as mentioned in Remark 2. These testing parameters were validated by means of the cross-validation technique.

Last, owning to the small size of our training data set, the corresponding QP problem was solved by a conventional QP solver using Matlab. The resulting SVR and the distribution of the support vector points are plotted in Fig. 4. The support vector numbers are listed in Table I. It is evident that the representations of the SVR is quite sparse.

Tustin model is one of the often used friction models for control design purpose [16]. It can be represented in the form of (6) with $f_\pm(\omega)$ ($f(\omega)$ for brevity) described by

$$f(\omega) = f_c + (f_s - f_c)\exp\left(-\left(\frac{|\omega|}{\omega_s}\right)\right) + f_\omega|\omega|$$

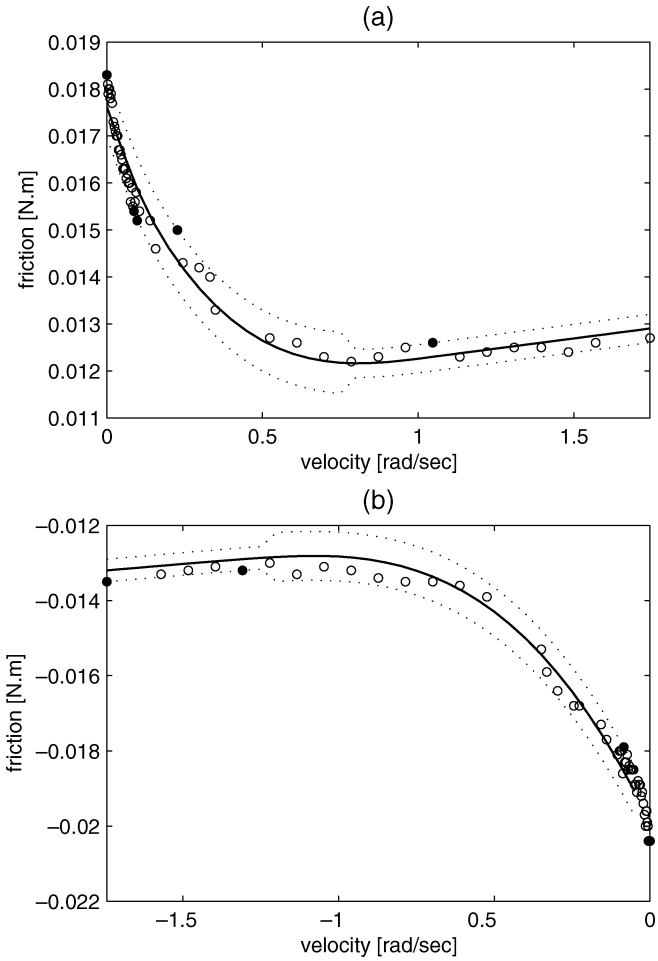

Fig. 4. SVR parameterizations and support vector distribution: solid circle—support vector points; dotted line—tolerant error margin. (a) Positive-velocity regime. (b) Negative-velocity regime.

where f_c is the level of the Coulomb friction force, f_s is the level of the static friction force, ω_s is the Stribeck velocity, and f_ω is the viscous coefficient. We estimate the parameters $(f_c, f_s, f_\omega, \omega_s)$ using the same data as in the SVR training. This was implemented by minimizing the mean square errors of the friction force prediction, using a nonlinear search technique. The estimated values are listed in Table I. The predicted friction force by the Tustin model is shown in Fig. 5. As expected, the training points fit the model well.

To evaluate the predicting capabilities of the Tustin model and the SVR, we collected another 34 testing samples, in which six samples are out of the range of the corresponding training data set, over both the positive- and negative-velocity regimes. Fig. 6 compares the SVR with the Tustin model. Two performance indices, the root-mean-square (rms) error and the maximum absolute error, are used to quantify the friction predicting capability.

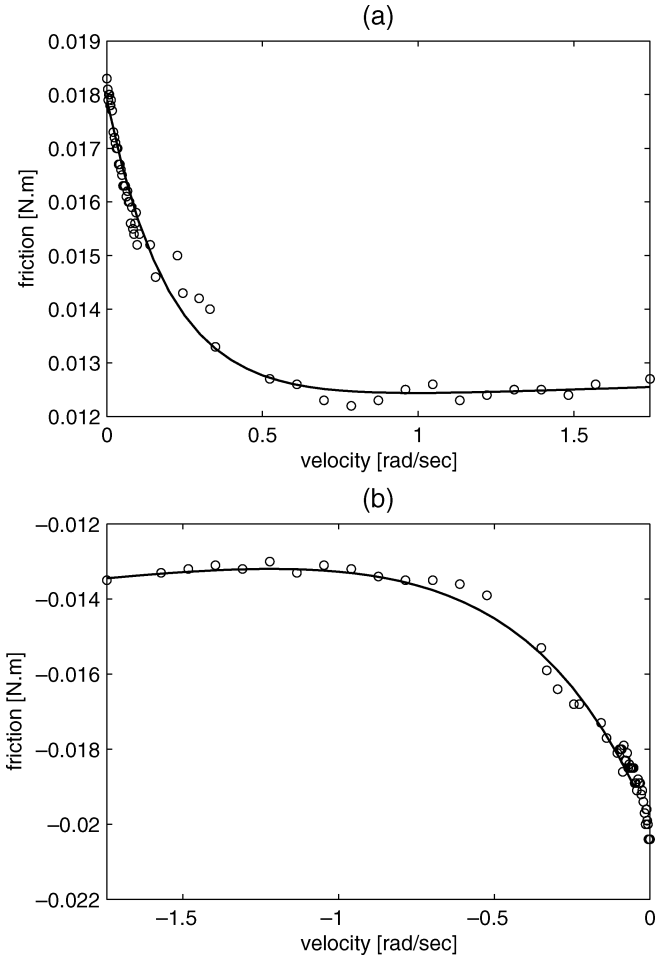

Fig. 5. Friction force prediction by model: circle—training points; solid line—Tustin model. (a) Positive-velocity regime. (b) Negative-velocity regime.

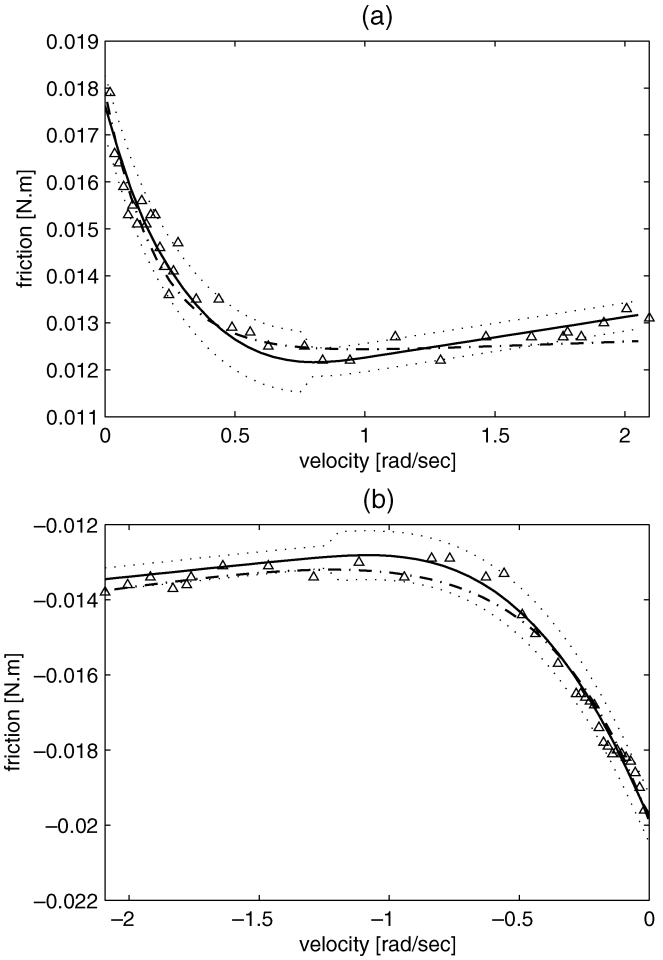

Fig. 6. SVR versus Tustin model: triangle—testing points; solid line—SVR; dotted–dashed line—Tustin model. (a) Positive-velocity regime. (b) Negative-velocity regime.

Table II lists the calculated results. It can be seen that both the rms errors and the maximum absolute errors of the friction force prediction by the SVR at the testing points are less than those by the friction model. Based on these observations, we argue that the SVR has the friction prediction confidence at a level similar to that by the Tustin friction model and it outperforms the conventional model-based approach.

TABLE II
COMPARISON OF FRICTION PREDICTION: TUSTIN MODEL VERSUS SVR

velocity regime	root mean square $\times 10^4$		maximum $\times 10^4$	
	Tustin model	SVR	Tustin model	SVR
positive	4.1187	3.5592	10.0000	8.1392
negative	3.5096	3.0726	9.4271	6.6364

B. Tracking Control Results

The controllers associated with two aforementioned scenarios can be represented in a unified form

$$\tau = \underbrace{I\ddot{q}_d}_{\text{(I)}} + \underbrace{k_p e + k_d \dot{e}}_{\text{(II)}} + \underbrace{\hat{F}}_{\text{(III)}}.$$

The component (I) is the inertia compensation term depending only on the reference trajectory. The component (II) is a typical proportional–derivative (PD) feedback control with $k_p = k_r\lambda$ and $k_d = k_r + I\lambda$. The component (III) is the friction estimator depending on which scenarios are being considered.

To stress the performance improvement due to the friction estimator, the same PD control scheme is used for both scenarios. The first scenario employs the SVR defined in Table I to capture the friction force. The second scenario uses a support vector network as a friction estimator. The design parameters were determined based on the guidelines described at the end of Section III-B.

TABLE III
DESIGN PARAMETERS FOR TRACKING CONTROLLER

task	B_q	B_ω	λ	k_r	γ	σ
non-unidirectional	0.3491	1.6442	2.8	0.7	0.3	1.0
unidirectional	0.9186	1.9514	1.2	0.5	0.3	1.0

For our system, $\mathcal{Q} = \{q \in \mathbb{R} | |q| \leq 2\pi/3\}$ and $\bar{\Omega} = \{\omega \in \mathbb{R} | |\omega| \leq 2.0933\}$ are of special interest. For the first tracking task, it is calculated based on the reference trajectory shown in Fig. 3(a) that $B_q = 0.3491$ and $B_\omega = 1.6442$. Following (33), λ should satisfy $2.3549 < \lambda < 3.1399$. In our studies, we set $\lambda = 2.8$. This leads to $B_q^0 = 0.111$, which defines the upper bound of the initial condition. It follows from (34) that the upper

TABLE IV
PERFORMANCE IMPROVEMENT BY USING SUPPORT VECTOR NETWORK

task	$\frac{\mathrm{RMS}_1-\mathrm{RMS}_2}{\mathrm{RMS}_1}\times 100\%$	$\frac{\mathrm{MAX}_1-\mathrm{MAX}_2}{\mathrm{MAX}_1}\times 100\%$
non-unidirectional	22.57	33.98
unidirectional	40.53	39.77

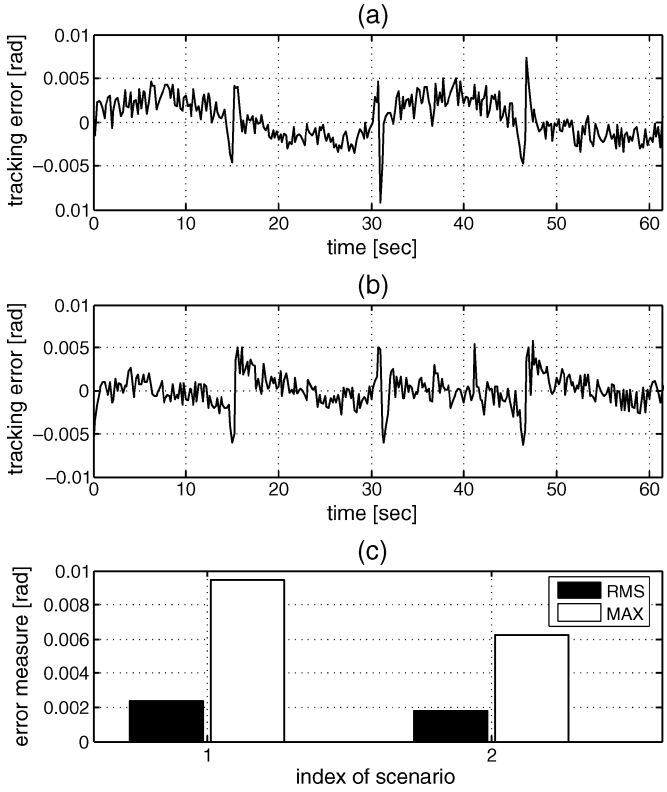

Fig. 7. Tracking errors for nonunidirectional positioning task: (a) with SVR parametrization, (b) with support vector network, and (c) the performance measure.

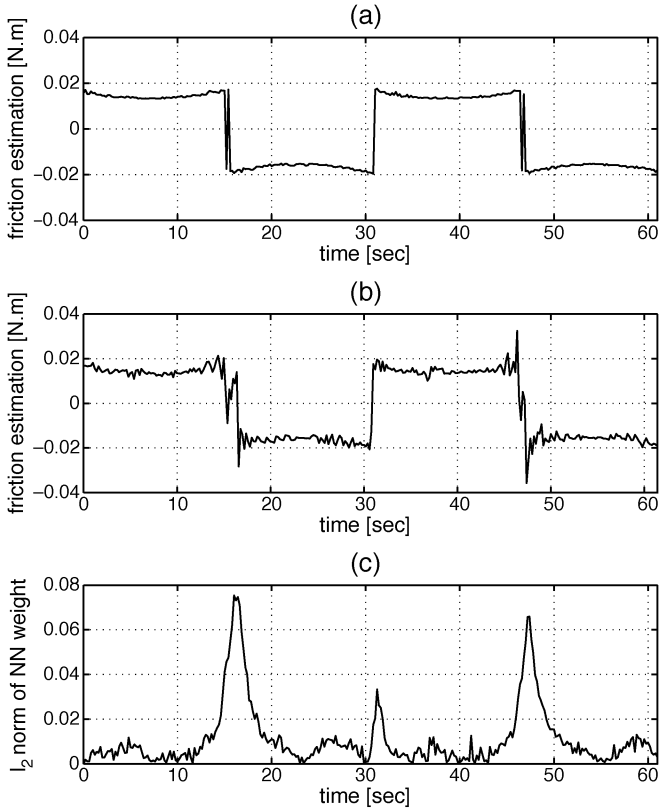

Fig. 8. Friction estimates for nonunidirectional positioning task: (a) with SVR parametrization, (b) with support vector network, and (c) l_2 norm of $\hat{\boldsymbol{v}}_{\mathrm{ad}}$.

bound R_0 is specified as 0.3560, which plays an important role in guiding the choices of the remaining design parameters. We set $k_r = 0.7$. The matrix $\boldsymbol{\Gamma} = [\gamma_{ij}] \in \mathbb{R}^{9\times 9}$ in the update rule (25) for $\hat{\boldsymbol{v}}_{\mathrm{ad}} \in \mathbb{R}^9$ is chosen as a diagonal matrix with all the diagonal elements as $\gamma_{ii} = 0.3$ $(i = 1, 2, \ldots, 9)$. In addition, we simply set $\sigma = 1$. Repeat the same design procedure based on the reference trajectory shown in Fig. 3(b) for the second tracking task. A list of the design parameters for both tracking tasks are given in Table III.

To highlight the friction compensation effects, the initial conditions $q(0)$ and $\dot{q}(0)$ are set to zeros for both scenarios. In addition, $\hat{\boldsymbol{v}}_{\mathrm{ad}}(0) = \mathbf{0}$ for the second scenario. The measures $\mathrm{rms}_i = \sqrt{\int_0^T |e(\mu)|\mathrm{d}\mu/T}$ and $\mathrm{MAX}_i = \max_{t\in[0,T]}|e(t)|$ are used to evaluate the tracking performance of the ith scenario, $i = 1, 2$. Here, $T > 0$ denotes the terminal time. Fig. 7 shows the tracking error results for the first tracking task. The results in Fig. 7(a) and (b) together imply that the support-vector-network-based friction compensation can reduce the tracking error significantly. Fig. 7(c) compares the tracking performance in terms of two measures. The second row in Table IV gives the percentage improvements of both performance criteria by the support vector network, which demonstrates that the support-vector-network-based friction compensator can considerably improve the tracking accuracy. In short, the support vector network shows more satisfactory performance than the SVR in friction compensation.

To show the effectiveness of the adaptive NN in friction estimation, the friction estimates of the two scenarios are compared and shown in Fig. 8. Fig. 8(a) and (b) gives the outputs of the friction estimator $\hat{F}$, which indicates the major role of the adaptive NN in capturing the residual friction effects. The boundedness of the NN weight estimate $\hat{\boldsymbol{v}}_{\mathrm{ad}}$ is shown in Fig. 8(c). From these observations, we can conclude that the adaptive NN can substantially improve the tracking performance.

As for the first tracking task, Figs. 9 and 10 give the experimental results for the second tracking task. Again, these results confirm the effectiveness of using the support vector network in tracking performance improvement. The percentage improvements by the support vector network are summarized in the third row of Table IV.

Remark 5: It should be noted that the percentage improvement, especially of the rms measure by the support vector network in the second tracking task, is more significant than that in the first. This is mainly because there are three velocity reversals

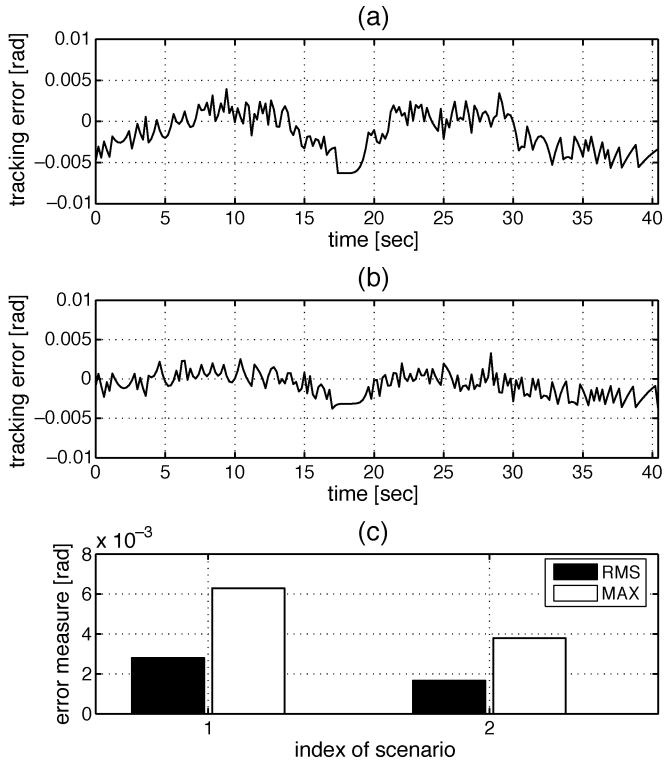

Fig. 9. Tracking errors for unidirectional positioning task: (a) with SVR parametrization, (b) with support vector network, and (c) the performance measure.

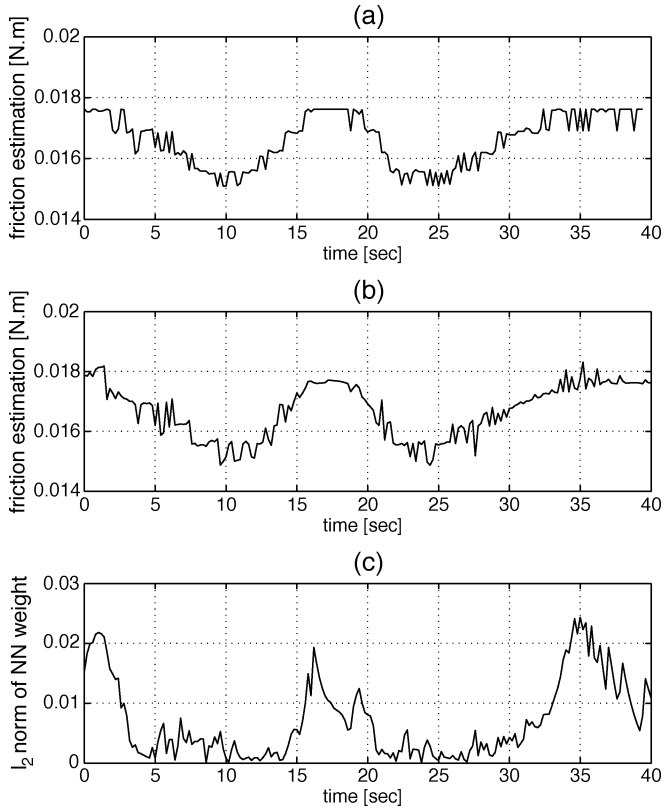

Fig. 10. Friction estimates for unidirectional positioning task: (a) with SVR parametrization, (b) with support vector network, and (c) l_2 norm of $\hat{\boldsymbol{v}}_{\mathrm{ad}}$.

in the reference velocity trajectory for the first tracking task. In this paper, the dynamic friction behavior around zero velocity is not accounted for in the friction model structure (6) and larger tracking errors are produced at the velocity reversals. Accurate tracking at extremely low velocity would require more sophisticated friction model structure that can account for the dynamic friction effect of velocity reversals. This is out of the scope of this work, but will be investigated in our future studies.

V. Conclusion

This paper presents the notion of support vector networks, a new paradigm for combining SVR with neural adaptive mechanism, in the context of friction compensation for servo motion control systems. This paper extends our previous work [19] in two aspects. The first one is to develop an enhanced adaptive friction compensator, while the second is an analysis that shows the improved performance and practical usefulness due to SVR. The proposed method was experimentally implemented on a force display device in our laboratory.

Compared with other NN's construction techniques, the SVR training is characterized by only a few parameters and is thus superior in setting up a parsimonious NN structure which can offer the guaranteed NN reconstruction capability as well as the required computational efficiency for adaptive NN-based control. In addition, the SVR straightforwardly provides satisfactory initialization of the NN weights which leads to the well-shaped transient behavior and thus augments the practical feasibility of the adaptive neural controllers. In short, the proposed support vector networks can effectively avoid the shortcomings due to the unsatisfactory architecture or initialization of NNs in adaptive neural control. Our methodology is presented on a servo motion system, but it can be applied to a much wider range of applications.

References

[1] S. S. Ge, C. C. Hang, and T. H. Lee, *Stable Adaptive Neural Network Control*. Norwell, MA: Kluwer, 2002.
[2] S. Fabri and V. Kadirkamannathan, "Dynamic structure neural networks for stable adaptive control of nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1151–1167, Sep. 1996.
[3] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[4] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. Annu. Conf. Comput. Learn. Theory*, D. Haussler, Ed., Pittsburg, PA, 1992, pp. 144–152.
[5] T. Y. Kwok and D. Y. Yeung, "Constructive algorithms for structure learning in feedforward neural network for regression problems," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 630–645, May 1997.
[6] R. Reed, "Pruning algorithms—A review," *IEEE Trans. Neural Netw.*, vol. 4, no. 5, pp. 740–747, Sep. 1993.
[7] I. Rivals and L. Personnaz, "Neural network construction and selection in nonlinear modeling," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 804–819, Jul. 2003.
[8] J. H. Park, S. H. Huh, S. H. Kim, S. J. Seo, and G. T. Park, "Direct adaptive controller for nonaffine nonlinear systems using self-stucturing neural network," *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 414–422, Mar. 2005.
[9] J. Moody and C. Darken, "Fast learning in network of locally-tuned processing units," *Neural Comput.*, vol. 1, pp. 281–294, 1989.
[10] T. Kohonen, *Self-Organizing Maps*. Berlin, Germany: Springer-Verlag, 1995.
[11] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least square learning algorithm for radial basis function networks," *IEEE Trans. Neural Netw.*, vol. 2, no. 2, pp. 302–309, Mar. 1991.
[12] B. Schölkopf, K.-K. Sung, C. J. C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2758–2765, Nov. 1997.

[13] B. Armstrong-Helouvry, P. Dupont, and C. Canudas de Wit, "A survey of models, analysis tools, and compensation methods for the control of machines with friction," *Automatica*, vol. 30, no. 7, pp. 1083–11838, 1994.
[14] C. Canudas de Wit and S. S. Ge, "Adaptive friction compensation for systems with generalized velocity/position friction dependency," in *Proc. 36th Conf. Decision Control*, San Diego, CA, 1997, pp. 2465–2470.
[15] S. N. Huang, K. K. Tan, and T. H. Lee, "Adaptive friction compensation using neural network approximation," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 30, no. 4, pp. 551–557, Nov. 2000.
[16] S. S. Ge, T. H. Lee, and J. Wang, "Adaptive NN control of dynamic systems with unknown dynamic friction," in *Proc. 39th Conf. Decision Control*, Sydney, Australia, 2000, pp. 1760–1765.
[17] H. Du and S. S. Nair, "Low velocity friction compensation," *IEEE Control Syst. Mag.*, vol. 18, no. 2, pp. 61–69, Apr. 1998.
[18] Y. H. Kim and F. L. Lewis, "Reinforcement adaptive learning neural-net-based friction compensation control for high speed and precision," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 1, pp. 118–126, Jan. 2000.
[19] G. L. Wang, Y. F. Li, and D. X. Bi, "Support vector machine network for friction modeling," *IEEE/ASME Trans. Mechatron.*, vol. 9, no. 3, pp. 600–607, Sep. 2004.
[20] D. Bi, Y. F. Li, T. S. Tso, and G. L. Wang, "Friction modeling and compensation for haptic display based on support vector machine," *IEEE Trans. Ind. Electron.*, vol. 51, no. 2, pp. 491–500, Apr. 2004.
[21] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipultors and Nonlinear Systems*. London, U.K.: Taylor & Francis, 1999.
[22] C. Canudas de Wit, P. Noel, A. Aubin, and B. Brogliato, "Adaptive friction compensation in robot manipulators: Low-velocities," *Int. J. Robot. Res.*, vol. 10, no. 3, pp. 189–199, 1991.
[23] G. Wahba, *Spline Models for Observational Data*, ser. Applied Mathematics. Philadelphia, PA: SIAM, 1990.
[24] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Adv. Comput. Math.*, vol. 13, pp. 1–50, 2000.
[25] B. Hammer and K. Gersmann, "A note on the universal approximation capability of support vector machines," *Neural Process. Lett.*, vol. 17, pp. 43–53, 2003.

G. L. Wang received the B.Sc., M.Sc., and Ph.D. degrees from the Nankai University, Tianjin, China, in 1986, 1989, and 1992, respectively.

In 1992, he joined the Department of Computer Science, Shantou University, where he was the Department Head from 1996 to 2000. As an Alexander von Humboldt Research Fellow, he was a Guest Scientist in the Control Engineering Laboratory, Ruhr-University of Bochum, Germany, from 2000 to 2001, and the Institute for System Theory in Engineering, University of Stuttgart, Stuttgart, Germany, in 2003. He was also a Visiting Research Fellow at the City University of Hong Kong, Kowloon, Kong Kong, from 2001 to 2002. Currently, he is a Full Professor at the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China. His research interests include sensor-based robot control and haptic interfaces for human machine systems.

Y. F. Li (SM'01) received the Ph.D. degree in robotics from the Department of Engineering Science, University of Oxford, Oxford, U.K., in 1993.

From 1993 to 1995, he was a Postdoctoral Research Associate in the AI and Robotics Research Group, Department of Computer Science, University of Wales, Aberystwyth, U.K. In 1995, he joined the City University of Hong Kong, Kowloon, Hong Kong, where currently he is an Associate Professor in the Department of Manufacturing Engineering and Engineering Management. His research interests include robotics, robot vision, robot sensing, and sensor-based control.

Dr. Li is an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.

D. X. Bi received the B.S. degree in mechatronics engineering and the M.S. degree in mechanical engineering from Harbin Institute of Technology, Harbin, China, in 1992 and 1998, respectively. Currently, he is working towards the Ph.D. degree at the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong.

His research interests include mechatronics system design, haptic display, and modeling and control of dynamic systems.