Hindawi Publishing Corporation
Chinese Journal of Engineering
Volume 2014, Article ID 295210, 6 pages
http://dx.doi.org/10.1155/2014/295210

Research Article

PD-SVM Integrated Controller for Robotic Manipulator Tracking Control

Neha Kapoor and Jyoti Ohri

National Institute of Technology, Kurukshetra 136119, India

Correspondence should be addressed to Neha Kapoor; ernehakapoor@rediffmail.com

Received 15 October 2013; Accepted 16 December 2013; Published 3 February 2014

Academic Editors: Y. Lei and B. Sun

Highly precise tracking of a robotic manipulator in presence of uncertainties like noise, disturbances, and friction has been addressed in this particular paper. An integrated proportional derivative and support vector machine (SVMPD) controller has been proposed for manipulator tracking. To illustrate the efficiency of the proposed controller, simulations have been done on a 2-DOF manipulator system. Performance of the proposed controller has been checked and verified with respect to to a simple PID controller and the radial bias neural network proportional integral derivative (RBNNPD) controller. It has been proved that the proposed controller can achieve better tracking performance as compared to other controllers as the range of errors is less and the time taken by the controller has reduced up to 14 times as compared to RBNN.

1. Introduction

Despite of the increase in complexity of modern control systems, due to simplicity and satisfactory performance, most industrial robots employ simple independent PD/PID controller [1–3]. In spite of so much popularity of PD controllers, this controller has some of the major drawbacks like frustrating tuning process and nonadaptability to time varying, nonlinear and coupling problems. Hence, with the growth of modern intelligent control techniques; PD is made to hybrid with these intelligent control techniques. These intelligent PD controllers have shown improved performance in past like fuzzy PID [4, 5], neural network based PID [6, 7], and genetic algorithm based PID [8].

Among all of the controllers, learning capability of the NN and SVM is used for learning nonlinear functions and any other characteristics of a system had made the researchers to pay a lot of attention. After learning, NN and SVM networks have the capabilities of generalization and then respond optimally to the unknown situations. In this, some drawbacks of NN the like; there is no general way for deciding the network topology, that number of neurons required are also has to be determined experimentally, and local minima, limits its wide application. Hence, in recent years, SVM theory, has replaced many of the control problems. SVM was originally developed for pattern recognition and classification tasks [9]. It has been proved that is newly developed SVM by Vapnik [9] has better generalization and easier to find global optimal solution in comparison to neural networks with less number of training data [10, 11]. Vapnik in 1995 successfully extended the SVM to solve the regression problems. With the passage of time SVM had shown a great performance in regression and time series prediction applications [12].

Uncertainties present in the robotic manipulator dynamics are one of the major limitations in achieving a high precision performance control. A few of these uncertainties include the friction, noise and disturbances in the path of the manipulator. To improve the performance of the tracking performance of the manipulator, in this paper PD-SVM integrated controller has been proposed. Supremacy of the proposed controller has been proved by simulations on a 2-DOF robotic manipulator. Further paper is sectioned as follows Section 2 has problem formulation in it, Section 3 has the various controllers applied to the manipulator, simulation example is given in Section 4 and lastly conclusions are given in Section 5.

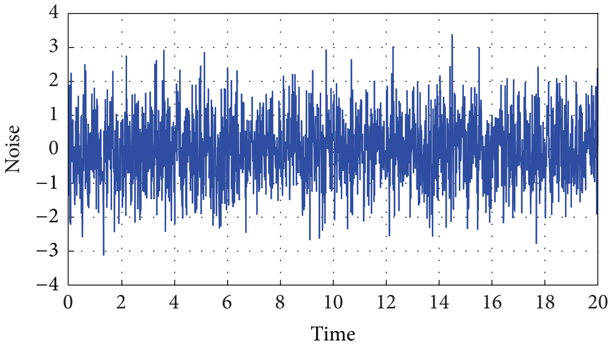

Figure 1: Uniform random white noise.

2. Problem Formulation

The dynamic equations of the robot manipulator are found through the use of the Lagrangian formulation, and the dynamic equation of an n-degree of freedom manipulator can be written as [13, 14] in

$$M(q)\ddot{q} + V(q,\dot{q}) + G(q) + T_d + F = \tau \tag{1}$$

with $q \in R^n$ as the join position variables, $\tau \in R^n$ as vector of input torque, $M(q) \in R^{n\times n}$ being the inertia matrix which is symmetric and positive definite, $V(q,\dot{q}) \in R^{n\times n}$ being the coriolis and centripetal matrix, $G(q) \in R^n$ includes the gravitational forcesing and $T_d \in R^n$ being the uncertainties in the robotic manipulator dynamics. $F \in R^n$ is the LuGre friction model.

2.1. Disturbances. External changes that affect performance of the system are referred to be as disturbances. Whatever the source of the change-internal or external, it is desired that the controller chosen is like that it makes the effects of the change (disturbance) as small as possible. These disturbances are the major limitations to achieve high precision performance control of nonlinear systems like manipulators. Major disturbances taken in this paper are as follows.

(a) Continuous disturbance: disturbance is changes continuously with time are continuous disturbance. affect of this torque has been seen on the robotic manipulator.

(b) Uniform random white noise: uniform random white noise is a random signal with a flat (constant) power spectral density. A pictorial view of inserted uniform random white noise in the manipulator system is Figure 1.

(c) Friction: The LuGre model is a dynamic friction model presented in [15]; Friction is modeled as the average deflection force of elastic springs, when a tangential force is applied the bristles will deflect like springs, if the deflection is sufficiently large the bristles start to slip. The average bristle deflection for a steady state motion is determined by the velocity. It is lower at low velocities, which implies that the steady state deflection decreases with increasing velocity, this models the phenomenon that the surfaces are pushed apart by the lubricant. LuGre Fiction can be modeled mathematically as follows:

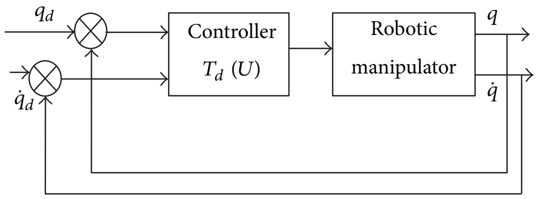

Figure 2: General block diagram for manipulator control.

$$\begin{gathered}\dot{z} = v - \frac{|v|}{g(v)}z, \\ F = \sigma_o z + \sigma_1 \dot{z} + \sigma_2 v, \\ g(v) = F_c + (F_s - F_c)\exp\left(\frac{v}{v_s}\right)^2,\end{gathered} \tag{2}$$

where z is average bristle deflection, σ_o is stiffness of bristles, σ_1 is bristle damping coefficient, σ_2 is viscous damping coefficient, v is relative velocity between moving parts, F_c is coulomb coefficient, F_s is static coefficient, and v_s is stribeck velocity.

3. Controllers

The control problem in the manipulator can be defined as finding out the input torque given to the manipulator joints, in order to have the accurate tracking performance even in the presence of the uncertainties. These input torqueses are taken as the output from the controllers. Figure 2 represents the general block diagram of the control system for the manipulator.

The controller (U) is PD, RBNNPD and SVMPD and the T_d are the various disturbances naming continuous disturbance, white noise and F is LuGrefriction.

Let the tracking error vector and error velocity be defined in (3) as

$$\begin{gathered}e = q - q_d, \in R_n, \\ \dot{e} = \dot{q} - q_d, \in R_n.\end{gathered} \tag{3}$$

3.1. Proportional Derivative (PD) Controller. The following control term is obtained for the PD controller:

$$\tau = K_p e(t) + K_d \dot{e}(t) - T_d - F, \tag{4}$$

where K_p and K_d are suitable positive definite diagonal $n \times n$ matrices. Values of the controller constants, that is, K_d and K_p, are being decided by TAE (Trial And Error) method.

3.2. Radial Bias Neural Network PD (RBNNPD) Controller. One of the most important and widely used techniques for regression is based on neural networks. Learning capability

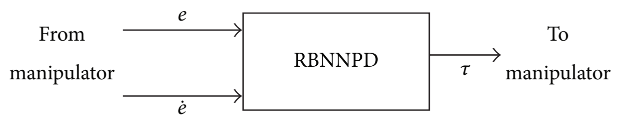

FIGURE 3: RBNNPD controller.

of the NN is used for learning nonlinear functions for direct/indirect dynamics and any other characteristics of a system had made the researchers to pay a lot of attention to the field. After learning, NN has a capability of generalization and then respond optimally to the unknown situations. Radial bias function network (RBFN) is a type of neural network having great mapping ability.

Input torque given to the manipulator is very important. In this paper, a RBNN network is trained by using the data of a simple PD controller. Input given to the NN is error and velocity error (e and $\dot{e}$) and output is taken as torque. Gaussian function is used as the activation function of each neuron in the hidden layer. The excitation values of these Gaussian function are distributed between the input values of the sliding surface value, s. The output of the network is given by (5) as

$$u = \sum_{j=1}^{n} w_j \exp\left[\frac{-\left\|s - c_j\right\|^2}{{\sigma_j}^2}\right], \tag{5}$$

where j is the jth neuron of the hidden layer, c_j is the central position of the neuron j, and σ_j is the spread factor of Gaussian function. A general diagram representing RBNNPD is shown in Figure 3.

This trained RBNNPD is used further to find the output torque to be given to the manipulator.

3.3. Support Vector Machine PD (SVMPD) Controller. Support vector machine is used to replace the neural networks by overcoming all the drawbacks of the neural network [11]. In a short time of span, control using SVM technique has shown the better results [16]. The main objective of the regression is to estimate a function from a set of samples given. Defining some of the basics of the ε-SVM regression as N data points $\{x_k, y_k\}$ $(k = 1 \cdots N)$ with $x_k \in R^n$ and output $y_k \in R^n$. The models take the form as in (6)

$$f(x) = \langle w, x\rangle + b, \tag{6}$$

where $\langle \cdot, \cdot \rangle$ denotes inner product. Optimal regression function is determined by the minimum of

$$\phi(w) = \frac{1}{2}\|w\|^2. \tag{7}$$

ε is the tolerance loss function. ε-SVR is defined as maximizing the geometric margin and minimizing the training error. This can be formulated as

$$\begin{aligned} &\min_{(w,b,\xi_i,\xi_i^*)} \left(\frac{1}{2}\|w\| + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)\right) \\ &\text{subject to } y_i - \langle w, x_i\rangle - b \le \varepsilon + \xi_i \\ &\qquad \langle w, x_i\rangle + b - y_i \le \varepsilon + \xi_i^* \\ &\qquad \xi_i, \xi_i^* \ge 0, \end{aligned} \tag{8}$$

where $\|w\|$ is the Euclidean norm of weights representing model complexity. $C > 0$ determines the tradeoff between model complexity and empirical loss function. ξ_i, ξ_i^* are the slack variables representing upper and lower constraints on output of system. The most common choice for the loss function is the following ε-insensitive loss function defined by Vapnik [9]:

$$L_g[y, f(x)] = \begin{cases} 0, & |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon, & |y - f(x)| > \varepsilon. \end{cases} \tag{9}$$

Introducing Lagrange multiplier and Kernel techniques, the resulting convex programming problem expressed in (10) is solved by its Wolfe dual formulation

$$\begin{aligned} L = \min_{\propto_i, \propto_i^*} \Bigg(&\frac{1}{2}\sum_{i,j=1}^{l}(\propto_i^* - \propto_i)(\propto_j^* - \propto_j) K(x_i, x_j) \\ &- \sum_{i=1}^{l}(\propto_i^* - \propto_i) y_i + \varepsilon \sum_{j=1}^{l}(\propto_i^* + \propto_i)\Bigg) \end{aligned} \tag{10}$$

$$\text{Subject to } \sum_{i=1}^{l}(\propto_i^* - \propto_i) = 0,$$

where

$$0 \le \propto_i, \quad \propto_i^* \le C, \quad i = 1 \cdots l. \tag{11}$$

$\propto_i, \propto_i^*$ are nonzero Lagrange multipliers [17] and $K(x_i, x_j)$ is the kernel function. For linearly separable data linear kernels are used and for nonlinear separable data non linear kernels are used. In this paper non linear ε-SVM regression technique with Radial Bias kernel function is used which can be formulated as

$$K(x_i, x_j) = \exp^{(-\gamma|u-v|^2)}. \tag{12}$$

Using KKT conditions offset b can be calculated as in (13)

$$b = \begin{cases} y_i - \sum_{j=1}^{l}(\propto_j^* - \propto_j) K(x_i, x_j) + \varepsilon & \propto_i > 0, \\ y_i - \sum_{j=1}^{l}(\propto_j^* - \propto_j) K(x_i, x_j) - \varepsilon & \propto_i^* > 0. \end{cases} \tag{13}$$

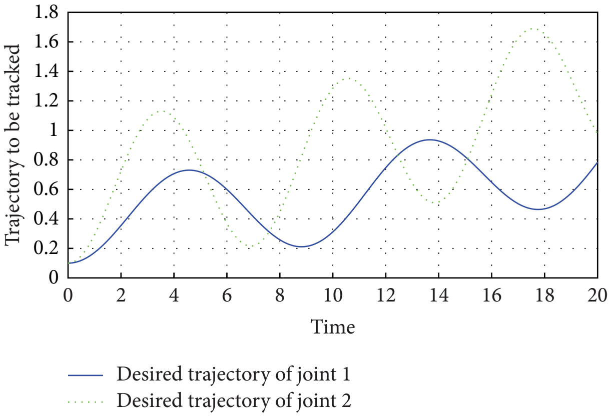

Figure 4: Desired trajectories by joint 1 and joint 2.

Finally output of SVM is given as

$$\hat{f}(x) = \sum_{i=1}^{l} (\propto_i^* - \propto_i) K(x_i, x_j) + \bar{b}, \tag{14}$$

where $\bar{b}$ is the average of b by using (13).

Error of the system should be

$$\left| \hat{f}(x) - f(x) \right| \le \varepsilon. \tag{15}$$

This ε-SVM regression controller is trained by using the data of a PD controller. Same as RBNNPD inputs given to the ε-SVM regression model is error and velocity error (e and $\dot{e}$) and output is taken as torque.

4. Simulation Example

4.1. Robot Dynamics. In order to show the effectiveness of the proposed intelligent control law, this have been applied to two-links robot with the parameters given below. The dynamics of a 2-DOF manipulator used in all types of controllers and satisfying (1) is given as

$$\begin{gathered} M(q) = \begin{bmatrix} 8.77 + 1.02 * \cos q_2 & 0.76 + .51 * \cos q_2 \\ 0.76 + .51 * \cos q_2 & 0.62 \end{bmatrix}, \\ V(q, \dot{q}) = \begin{bmatrix} -.51 \sin(q_2) \dot{q}_2 & -.51 \sin(q_2)(\dot{q}_1 + \dot{q}_2) \\ -.51 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix}, \\ G(q) = \begin{bmatrix} 74.48 \sin(q_1) + 6.174 \sin(q_1) + q_2 \\ 6.174 \sin(q_1) + q_2 \end{bmatrix}. \end{gathered} \tag{16}$$

This 2-DOF manipulator has been commanded to track the path shown in Figure 4 and given by

$$\begin{gathered} q_1^d = \left[.3 \sin\left(.7t - \frac{\Pi}{2} \right) + .3 \sin\left(.1t - \frac{\Pi}{2} \right) + .7 \right], \\ q_2^d = \left[.5 \sin\left(.9 * t - \frac{\Pi}{2} \right) + .5 \sin\left(.1t - \frac{\Pi}{2} \right) + 1.1 \right]. \end{gathered} \tag{17}$$

Table 1: Comparative results for the various controllers.

Controllers (with uncertainties)	Mean square error (MSE) Joint 1	 Joint 2	Time taken (in sec)
PD	0.0035	0.0142	1.24
RBNNPD	0.0024	0.0062	79.589
SVMPD	$4.65E - 06$	$7.39E - 04$	5.891

4.2. Disturbances. Continuous disturbance in the path of manipulator is taken as

$$T_d = \begin{bmatrix} 2\cos(.02t) \\ 1.7\sin(.02t) \end{bmatrix}. \tag{18}$$

And the LuGre Friction model is defined by constants as

$$\begin{gathered} \sigma_0 = .6, \quad \sigma_1 = .009, \quad \sigma_2 = .6, \quad v_s = .04, \\ F_s = .01, \quad F_c = 10. \end{gathered} \tag{19}$$

4.3. Controllers Parameters. Design parameters for PD controller are taken as

$$K_p = \begin{bmatrix} 800 & 0 \\ 0 & 120 \end{bmatrix}, \qquad K_d = \begin{bmatrix} 70 & 0 \\ 0 & 50 \end{bmatrix}. \tag{20}$$

For RBNNPD controller spread is taken as 2.

For ε-SVM regression, two MATLAB SVM libraries, namely, *svmtrain* and *svmpredict,* are used.

These libraries have the given format. For training SVM

> model = svmtrain(training_output, training_input, "libsvm options").

To predict class of new input data according to pretrained model

> predict_label = svmpredict(testing_output, testing_input, "libsvm options").

Values for radial bias kernel function are $[\gamma, C] = [1991, .5064]; [.73, 9099]$ for joint 1 and joint 2, respectively.

Figures 5 and 6 represent the actual trajectory tracked by the robotic manipulator with different controllers. It has been clearly represented in Figures 5 and 6 that for both the joints 1 and 2 the tracking performance of the SVMPD is much improved than the PD and RBNNPD controllers. Figures 7 and 8 are giving the continuous errors in joints 1 and 2 with different controllers. In Figures 7 and 8, it can be observed that the range of errors in joint 1 and joint 2 is minimum for SVMPD. RBNNPD has lesser continuous error when compared to a classical PD controller. Comparative results for the three controllers, namely, PD, RBNNPD and SVMPD have been tabulated in Table 1.

It can be observed from Table 1 that mse error of SVMPD is about 10^6 and 10^4 times lesser than the other two PD and RBNNPD controllers. Time taken by SVMPD is about 14 times lesser than the time taken by RBNNPD for control purpose.

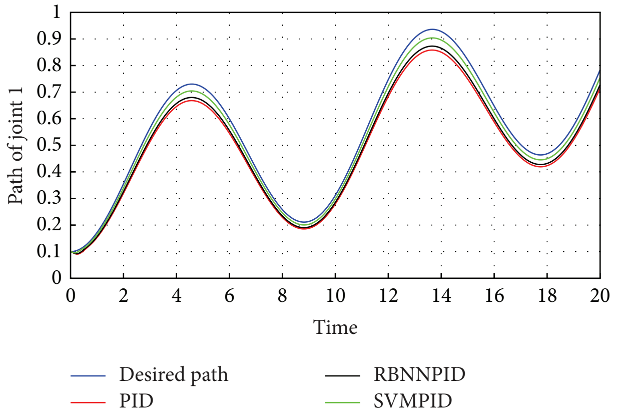

FIGURE 5: Trajectory tracked by joint 1. Path of joint 1 with different controllers.

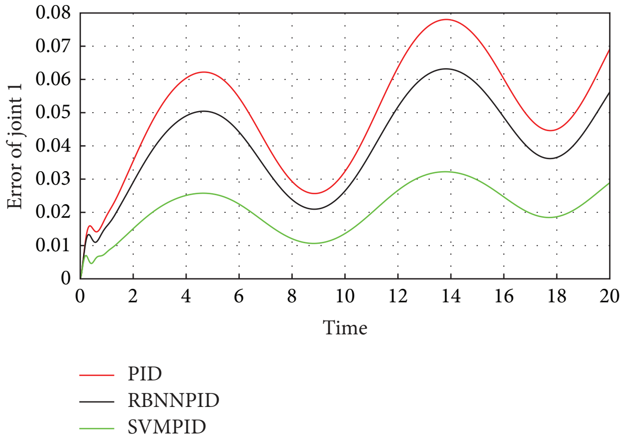

FIGURE 7: Error of joint 1 with different controllers.

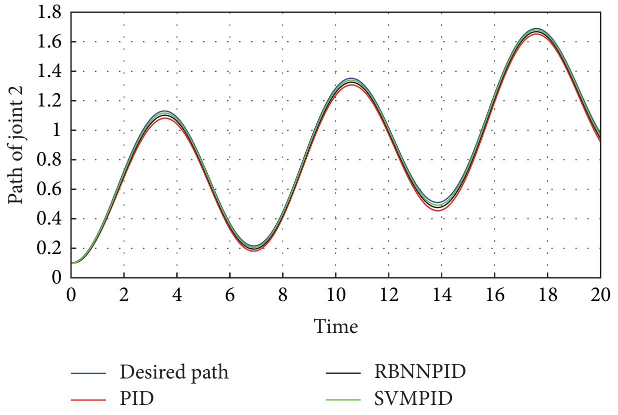

FIGURE 6: Trajectory tracked by joint 2. path of joint 2 with different controllers.

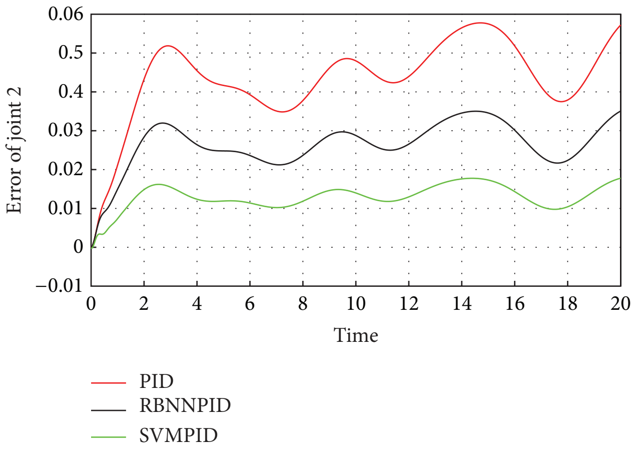

FIGURE 8: Error of joint 2 with different controllers.

5. Conclusion

In this paper, problem of path tracking of robotic manipulator with disturbances has been solved upto the best level. Along with the widely used controllers like PD, RBNNPD one of the advance upcoming controller, namely, SVMPDd has been implemented. Results have been checked by simulations on a 2-DOF manipulator. It has been found in the results that the proposed SVMPD controller takes less time for control and is the most robust controller. Even in presence of various disturbances, SVMPD performs better than the other controllers.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] K. J. Astrom and T. Hagglund, *PID Controllers: Theory Design and Tuning*, Instrument Society of America, Research Triangle Park, NC, USA, 1995.

[2] A. O. Dwyer, *Handbook of PI and PID Controller Tuning Rules*, Imperial College Press, London, UK, 2003.

[3] Y. Choi, W. K. Chung, and I. H. Suh, "Performance and H∞ optimality of PID trajectory tracking controller for Lagrangian systems," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 857–869, 2001.

[4] S.-J. Huang and J.-S. Lee, "A stable self-organizing fuzzy controller for robotic motion control," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 2, pp. 421–428, 2000.

[5] C.-M. Ko, S.-K. Park, T.-S. Yoon, D.-H. Kim, and G.-J. Chung, "Comparative study of fuzzy PD control and PI control for heavy duty robot," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '09)*, pp. 1–5, May 2009.

[6] M.-G. Zhang, Z.-G. Wang, and P. Wang, "Adaptive PID decoupling control based on RBF neural network and its application," in *Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR '07)*, pp. 727–731, Beijing, China, November 2007.

[7] J. Chen and T.-C. Huang, "Applying neural networks to on-line updated PID controllers for nonlinear process control," *Journal of Process Control*, vol. 14, no. 2, pp. 211–230, 2004.

[8] C. Li and J. Lian, “The application of immune genetic algorithm in PID parameter optimization for level control system,” in *2007 IEEE International Conference on Automation and Logistics (ICAL ’07)*, pp. 782–786, Jinan, China, August 2007.

[9] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[10] H. Du and S. S. Nair, “Modeling and compensation of low-velocity friction with bounds,” *IEEE Transactions on Control Systems Technology*, vol. 7, no. 1, pp. 110–121, 1999.

[11] P. Phatiwuttipat, W. Kongprawechon, K. Tungpimolrut, and S. Yuenyong, “Cardiac auscultation analysis system with neural network and SVM technique,” in *Proceedings of the 8th Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON ’11)*, pp. 1027–1030, May 2011.

[12] E. Sariyildiz, K. Ucak, G. Oke, and H. Temeltas, “A trajectory tracking application of redundant planar robot arm via support vector machines,” in *Proceedings of the International Conference on Adaptive and Intelligent Systems (ICAIS ’11)*, vol. 6943 of *Lecture Notes in Artificial Intelligence*, pp. 192–202, Springer, 2011.

[13] M. W. Spong and M. Vidyasagar, *Robit Dynamics and Control*, Wiley, New York, NY, USA.

[14] L. Sciavicco and B. SIciliano, *Modeling and Control of Robot Manipulators*, Spinger, 2nd edition, 2006.

[15] H. Olsson, K. J. Åström, C. Canudas de Wit, M. Gäfvert, and P. Lischinsky, “Friction models and friction compensation,” *European Journal of Control*, vol. 4, no. 3, pp. 176–195, 1998.

[16] V. Blanz, B. Scholkopf, H. Bülthoff, C. Burges, V. Vapnik, and T. Vetter, “Comparison of view-based object recognition algorithms using realistic 3D models,” in *Proceedings of the Artificial Neural Networks (ICANN ’96)*, C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, Eds., vol. 1112 of *Springer Lecture Notes in Computer Science*, pp. 251–256, Berlin, Germany, 1996.

[17] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.