Modified extreme learning machine based motion control of robotic manipulators

Neha Kapoor* and Jyoti Ohri

Department of Electrical Engineering, National Institute of Technology, Kurukshetra, Haryana, India

*Email: ernehakapoor@rediffmail.com

Abstract. Precise trajectory tracking is a difficult task due to partially known and unknown dynamics and the disturbances present in the system. For improving the tracking performance of the robotic manipulator, this work proposes a novel PSO optimized kernel based Extreme Learning Machine (PSO-KELM) learning algorithm in which PSO is used to get the optimal values of the free kernel-parameters in KELM. The simulation results represent the good generalized performance and PSO-KELM outperforms the KELM and ELM based control techniques for the manipulator trajectory tracking. Comparative analysis of the proposed control schemes have been done with NN and SVM based controllers and various ELM based variants for the trajectory tracking problem in robotic manipulator.

1. Introduction

In modern times, the accuracy in position control in trajectory tracking of robotic manipulator is crucial to the commercial viability and widespread use in industries. Inherent characteristics of being highly non-linear and strongly coupled and the recent increasing demand for industrial robots has provided the researchers working in the area with a new challenging and time consuming problem of finding out the most accurate control input torque to robot actuators to take the robot to as close as to the desired trajectories. Hence, main motto of the work carried out in this paper is to propose the most recent and improved ELM based control techniques for the trajectory tracking problem of the manipulator. Joints are driven by actuators and hence, for practicality driving actuator dynamics have been taken into consideration. To validate the performance, proposed controllers are made to work with manipulators with planar elbow direct driven links and remotely driven links.

For improving the control performance for position control problem of robotic manipulator a tremendous research in [1-4 and references there in] has been done. However, most NN based forecasting methods cannot avoid a number of difficulties like choice of stopping criteria, learning rate and epochs, local minima etc. Support vector machines (SVM) based control schemes having better learning, have shown an improved performance [5-8]. Another new more recent regression technique, proposed by Huang [9,10], having similar performance as SVM is Extreme Learning Machine (ELM) but needs lesser optimization constraints and has better generation with less operation time and hence, has been used in many research areas apart from motion control problem of robotic manipulators [11]. Although ELM is an efficient modern technique, but a values of non-optimal input weights and hidden biases may lead to over fitting. Some improvements in the ELM basic algorithm have been mentioned in [12]. However, how to get the optimal combination of the hidden layer parameters is still...
unresolved. As the feature mapping function of hidden neurons is unknown, like SVM, ELM can be extended to kernel ELM, which is called the kernel based extreme learning machine (KELM), which leads to its stability improvement too [11]. In KELM, no hidden layer parameter tuning is required, and however, the kernel parameters should be chosen properly for improving performance of the KELM algorithm. Kernel parameters in kernel functions of the proposed algorithm have been optimized using and meta-heuristic optimization technique, i.e. particle swarm optimization. Effectiveness of the proposed algorithm for the problem undertaken i.e. motion control of a robotic manipulator has been shown in the simulation study carried out in this work.

Other sections of the paper are organized as follows: Section 2 contains the basics of the ELM/KELM. Particle swarm optimization based KELM has been presented in Section 3. Preliminaries are contained in Section 4 and simulation study has been described in Section 5. Section 6 has performance analysis and the result compilation followed by the conclusions in the Section 7.

2. Kernel extreme learning machines

Recently developed ELM tends to give the best generalization performance at tremendously fast learning speed. Huang et al. [13] contains a detail study of ELM with hidden neurons, H and activation function, f(x) can approximate training data, K without any error. ELM can universally very well approximate a continuous function [14]. Also, further Huang et al. [10] proposed some modified forms of ELM. Huang et al. [10] suggested that (8) provide least training error, minimum weight norms having greatest generalization, and a unique solution, and thus involves no local minima as with the back propagation learning algorithm. With some of the modifications new proposed ELMs can be found in [15 & 16].

Radial basis, sine, cosine and exponential functions are some of the differential activation function [10] for which ELM works. Inspite of simplified learning approach of ELM, two major issues of ELM are: a) the number of hidden neurons (i.e. the size of the SLFNs). Usually users choose the optimal number of hidden nodes by trial-and-error, b) to reduce the computation complexity of ELM especially if given large number of training data and if large number of hidden nodes required. Also, ELM may suffer either from under-fitting or over-fitting. Out of these two problems, the over-fitting is more important because it can easily lead to wild estimate. An over-fitting model can give poor predictive performance by exaggerating minor fluctuations in the data. Hence, in this work emphasis has been given on these aspects and for this kernel ELM (KELM) has been used and implemented for motion control problem of robotic manipulator.

Recently, Huang et al. [11] propose to use orthogonal projection and kernel methods for the designing of ELM. This provided a more stable solution of ELM with improved generalization capabilities as comparison to least square solution based ELM. Thus, with this new kernel method output function of ELM is given by:

$$h(x)\alpha = h(x)A^T\left(\frac{1}{\rho} + AA^T\right)^{-1}$$  \hspace{1cm} (1)

where all the symbols have their usual meanings and details of them can be found in [10, 11]. If hidden layer feature mapping h(x) is unknown, the kernel function is proposed. A kernel matrix for ELM can be represented as follows:

$$\chi_{ELM} = AA^T : \chi_{ELM_{ij}} = h(x_i)h(x_j) = B(x_i,x_j)$$  \hspace{1cm} (2)

where B(x_i,x_j) is a kernel function. Output function in (1) can be represented as:

$$\begin{bmatrix}B(x,x_1) \\
B(x,x_{K})\end{bmatrix}\left(\frac{1}{\rho} + \chi_{ELM}\right)^{-1}Y$$  \hspace{1cm} (3)

Here, a kernel function corresponding to h(x) is used and hence, exact knowledge of hidden layer feature mapping and the number of hidden nodes is not required. List of kernel functions used in ELM is similar to that are used with support vector machines, such as linear, polynomial and radial basis
function. Compared with the ELM learning algorithm, the hidden layer feature mapping need not be known and the numbers of hidden neurons require not to be selected in the KELM. Moreover, the KELM learning algorithm achieves similar or better generalization performance and is more stable compared to traditional ELM.

3. Particle swarm optimization based KELM

PSO [17], a recently developed heuristic optimization technique, has been proposed as an alternative to genetic algorithm (GA) [18]. Animal behavior based PSO has many attractive features as compared to GA. ISimple concept, easy implementation, quick convergence and more accurate optimization made PSO a worldwide famous computational algorithm. PSO is based on changing the velocity and position of each particle toward its pbest (current position best) and gbest (global best) locations according to equations (4) and (5) respectively, at each time step

\[ v_{id} = w \ast v_{id} + c_1 \varepsilon_1(p_{id} - x_{id}) + c_2 \varepsilon_2(p_{xd} - x_{id}) \quad (4) \]
\[ x_{id} = x_{id} + v_{id} \quad (5) \]

where, in a D-dimensional space, \( \vec{x}_i = (x_{i1}, x_{i2}, ..., x_{iD}) \) is a present position vector, \( \vec{p}_i = (p_{i1}, p_{i2}, ..., p_{iD}) \) is a best position vector, \( \vec{v}_i = (v_{i1}, v_{i2}, ..., v_{iD}) \) is a velocity vector, \( c_1 \) and \( c_2 \) are the acceleration coefficients having constant value: \( 2, \varepsilon_1 \) and \( \varepsilon_2 \) are the random number generators between \((0-1)\), \( w \) is the linearly changing inertia weight and \( d \) is the dimension \((1 \leq n \leq N)\). Instead of choosing a constant of \( w \) in (4), it can be chosen as in (6)

\[ w = (w_{max} - w_{min}) \left( \frac{\text{maxiter} - \text{iter}}{\text{maxiter}} \right) + w_{min} \quad (6) \]

Basic theory of particle swarm optimization (PSO) can be studied in author’s paper [19].

PSO-KELM theory:

In KELM learning algorithm, the regulation coefficient \( C \) and kernel parameters should be chosen appropriately for improving the generalization performance of neural networks. In [11], these parameters are tried in a wide range and this proved to be a time consuming task. And in [20], a hybrid kernel function is proposed for improving the generalization performance of KELM. However, how to choose the optimal values of the parameters of the kernel function has not been resolved. In this paper, an optimization approach, naming PSO, has been introduced to the KELM for choosing the optimal parameters of kernel function. Steps involved in this particular technique are:

Step 1  Initiate the population (particle) based on the kernel function and the velocity and position of each particle.
Step 2  Evaluate the fitness function of each particle.
Step 3  The velocity and position of the particle are modified according to the basic PSO algorithm.
Step 4  Step 2 and Step 3 are iterated repetitively until the maximal iteration time is satisfied.
Step 5  The optimal parameters of kernel function can be determined. Then, based on the optimized parameters, the hidden layer kernel matrix is computed.
Step 6  Determine the final output weights \( \beta \) in terms of the following equation:

\[ \beta = H^T((1/C + HH^T)^{-1}) \]

4. Preliminaries

4.1. Manipulator dynamics

The general dynamics of n-link robotic manipulator is given by following nonlinear differential equation [21]

\[ M(q)\ddot{q} + V(q, \dot{q}) + G(q) + F(q, \dot{q}) = \tau \quad (7) \]

with \( q \in \mathbb{R}^n \) is the joint position variables, \( \tau \) is vector of input torques, \( M(q) \) is the inertia matrix which is symmetric and positive definite, \( V(q, \dot{q}) \) is the coriolis and centripetal matrix, \( G(q) \) includes the gravitational forces and \( F(q, \dot{q}) \) is the frictional forces.
**Directly Driven Robotic Manipulator:**

Dynamic equation of a 2 DOF directly driven robotic manipulator satisfying equation (7) chosen in this work have been given in below as [21]. Equations representing the dynamics of directly driven link are given in (8)

\[
\begin{bmatrix}
    m_{11} & m_{12} \\
    m_{21} & m_{22}
\end{bmatrix}
\begin{bmatrix}
    \ddot{q}_1 \\
    \ddot{q}_2
\end{bmatrix}
+
\begin{bmatrix}
    v_{11} \dot{q}_1 \\
    v_{12} \dot{q}_2
\end{bmatrix}
+
\begin{bmatrix}
    g_1 \\
    g_2
\end{bmatrix}
+
\begin{bmatrix}
    F_1 \\
    F_2
\end{bmatrix}
=
\begin{bmatrix}
    \tau_1 \\
    \tau_2
\end{bmatrix}
\]  

(8)

In this equation,
\[m_{11} = (m_1 + m_2)a_1^2 + m_1a_1^2 + 2m_2a_2 \cos q_2; \quad m_{12} = m_2a_2^2 + m_2a_2 \cos q_2 = m_{21}; \quad m_{22} = m_2a_2^2; \quad v_{11} = -m_1a_1 \frac{2q_1}{2} \sin q_2; \quad v_{21} = m_2a_1 \dot{a}_2^2 \sin q_2; \quad g_{11} = (m_1 + m_2)g a_1 \cos q_1 + m_2a_2 \cos (q_1 + q_2); \quad g_{21} = m_2a_2 \cos (q_1 + q_2)\]

where \(m_1\) and \(m_2\) are the masses and \(a_1\) and \(a_2\) are the lengths of the links 1 & 2 respectively and \(g\) is the gravity acceleration, these have been chosen as below \(m_1 = 1\)kg; \(m_2 = 1\)kg; \(a_1 = a_2 = 1\)m; \(g = 9.8\)m/s².

**Remotely Driven Robotic Manipulator:**

The robotic manipulator considered here is a planar elbow manipulator with remotely driven link, whose dynamic model has been derived by Spong et al [21]. Equations representing the dynamics of remotely driven link are given in (9).

\[
\begin{bmatrix}
    m_{11} & m_{12} \\
    m_{21} & m_{22}
\end{bmatrix}
\begin{bmatrix}
    \ddot{q}_1 \\
    \dot{q}_2
\end{bmatrix}
+
\begin{bmatrix}
    v_{11} \dot{q}_1 \\
    v_{12} \dot{q}_2
\end{bmatrix}
+
\begin{bmatrix}
    g_1 \\
    g_2
\end{bmatrix}
+
\begin{bmatrix}
    F_1 \\
    F_2
\end{bmatrix}
=
\begin{bmatrix}
    \tau_1 \\
    \tau_2
\end{bmatrix}
\]  

(9)

In this equation,
\[m_{11} = m_1l_1^2 + m_2l_2^2 + l_1; \quad m_{12} = m_1l_1 \cos (p_2 - p_1); \quad m_{21} = m_2l_2 \cos (p_2 - p_1); \quad m_{22} = m_2l_2^2 + l_2; \quad v_{11} = -m_2l_2 \sin (p_2 - p_1); \quad v_{12} = m_2l_2 \sin (p_2 - p_1); \quad g_1 = (m_1l_1 + m_2l_2)g \cos (p_1); \quad g_2 = m_2l_2 g \cos (p_2)\]

where the subscripts 1 & 2 indicate link 1 & link 2. In this case, \(q_1\) is the angle with respect to horizontal axis; \(m_2\) is the weight; \(l_1\) is the inertia; \(l_2\) is the length; \(l_i\) is the distance from the joint to the centre of gravity; \(g\) is gravitational constant; \(\tau_i\) is the torque input. The parameters chosen are as following:
\[m_1 = 10; \quad m_2 = 5; \quad l_1 = 0.2; \quad l_2 = 0.2; \quad l_{c1} = 0.25; \quad l_{c2} = 0.5; \quad g = 9.8.\]

4.2. Actuator dynamics

A robot may have several kinds of influencing actuators-electric, pneumatic, or hydraulic. In this case DC motor with permanent magnet has been considered as actuator motor as it is popular in robotic applications. Detailed study and the numeric values of the dynamics of the armature controlled dc motor that drives the joints of manipulator can be obtained from [22] and has the same form as the robotic manipulator dynamics given in equation (7) and are given by:

\[
M_{RM}(q)\dot{q} + V_{RM}(q,\dot{q}) + G_{RM}(q) + F_{RM}(q,\dot{q}) = \tau_{RM}
\]

(10)

where,
\[M_{RM} = (J_M + R^2M); \quad V_{RM} = (B + R^2V); \quad G_{RM} = R^2G \quad \text{and} \quad F_{RM} = (RF_M + R^2F)\]

4.3. Uncertainties

- **Continuous Disturbance**: Disturbance changing continuously with time entering into the system is chosen as following:
\[T_d = 0.15 \times \cos (5t) + 0.15 \times \sin (5t)\]

(11)

- **Friction**: The LuGre model is a dynamic friction model presented in [25] is chosen here.

- **Payload changes**: An addition 30% extra load in mass of link 2 has been added as payload changes.

Predefined desired trajectories for joint 1 & 2 are given in (12) and (13) respectively below
\[ q_1^d = 0.3 \sin \left(0.7t - \frac{\pi}{2}\right) + 0.3 \sin \left(0.1t - \frac{\pi}{2}\right) + 0.7; \]  
\[ q_2^d = 0.5 \sin \left(0.9t - \frac{\pi}{2}\right) + 0.5 \sin \left(0.1t - \frac{\pi}{2}\right) + 1.1; \]  

Details of various performance evaluating indices opted here for performance evaluation can be found in [26].

5. Simulation example
In this ELM based control techniques error (e) and velocity error (\( \dot{e} \)) are given as input and the output is the control input torque (\( \tau \)). In the simulation study, number of neurons in ELM is chosen by trial and error and is equal to 29. In KELM radial bias kernel is chosen with gamma (\( \gamma \)) as 0.57 whereas in the proposed PSO-KELM, PSO is used to optimize gamma (\( \gamma \)) and the regulation coefficient (\( C \)). In PSO-KELM control technique PSO parameters are chosen as: number of population-25, number of iterations-30, linearly varying inertia weights ranging from 0.9 to 0.4 for \( w_{\text{max}} \) to \( w_{\text{min}} \), acceleration coefficients have been chosen constants =2. For this purpose range of \( \gamma \) is chosen from [0-4] and range of \( C \) is chosen from [200-400]. Cases constituted for the study under consideration are:

Case 1: under ideal condition with no uncertainties.

Case 2: with payload mass changes + LuGre friction + continuous disturbance + actuator dynamics.

Trajectory tracking for 2 dof directly driven robotic manipulator with the controllers naming, ELM, KELM and PSO-KELM of case 2 have been shown in figure 1(a) for joint 1 and (b) for joint 2 respectively. For clarity purpose, tracking results for these controllers have been zoomed in figure 2. figure 3 is representing the tracking error and control input torque, which clearly indicates the better tracking accuracy of PSO-ELM then KELM and ELM. Tracking results of case 1 are found to be similar to the results of case 2 and hence omitted. Also, tracking results for remotely driven manipulator have been found similar to directly driven manipulator and hence not shown here. Tracking results for NN and SVM based controllers for the motion control of a 2 dof robotic manipulator has been taken from the previous papers [23] (for NN) and [24] (for SVM) by the author.

Performance indices representing various performance evaluating factors for directly driven and remotely driven manipulators for case 1 & 2 have been given in table 1 and 2 respectively. Table 3 gives the execution time taken by the controllers.

**Figure 1.** Trajectory to be Tracked with ELM, KELM and PSO-KELM.

**Figure 2.** Trajectory Tracked with ELM, KELM and PSO-KELM.
**Figure 3.** (a) Tracking error: $e_1$; (b) Tracking error: $e_2$; (c) control input torque: $\tau_1$ and (d) control input torque: $\tau_2$.

**Table 1.** Quantitative performance indices for a 2 DOF robotic manipulator: Case 1.

| Controller | $\|e_1\|_2$ | $\|e_2\|_2$ | $\|\tau_1\|_2$ | $\|\tau_2\|_2$ | $c_{1u}$ | $c_{2u}$ |
|------------|-------------|-------------|---------------|---------------|-----------|-----------|
| Directly Driven |
| NN         | 0.33        | 0.49        | 998.7         | 355.08        | 0.00055   | 0.00035   |
| SVM        | 0.2         | 0.401       | 1.90E+03      | 244.68        | 6.70E-04  | 4.90E-04  |
|            | 0.17354     | 0.3880      | 1.5304e+03    | 946.069       | 3.9242e-04 | 3.95e-04  |
| ELM        | 0.147       | 0.4301      | 1.348E+03     | 797.51        | 3.44E-04  | 8.60E-04  |
| KELM       | 0.1124      | 0.2322      | 1.022E+03     | 444.5         | 6.70E-04  | 4.90E-04  |
| PSO-KELM   |             |             |               |               |           |           |
|            |             |             |               |               |           |           |
| Remotely Driven |
| NN         | 1.16        | 1.57        | 3.30E+04      | 1.50E+0        | 3.20E-04  | 7.60E-04  |
| SVM        | 1.56        | 1.15        | 3.30E+04      | 1.50E+0        | 3.20E-04  | 7.60E-04  |
|            |             |             |               |               |           |           |
| ELM        | 1.07        | 1.10        | 1.6+3         | 9.27E+0        | 4.30E-04  | 4.70E-04  |
| KELM       | 0.983       | 1.02        | 3.20E+03      | 1.60E+0        | 3.30E-04  | 8.20E-04  |
| PSO-KELM   |             |             |               |               |           |           |

**Table 2.** Quantitative performance indices for a 2 dof in the paper: Case 2.

| Controller | $\|e_1\|_2$ | $\|e_2\|_2$ | $\|\tau_1\|_2$ | $\|\tau_2\|_2$ | $c_{1u}$ | $c_{2u}$ |
|------------|-------------|-------------|---------------|---------------|-----------|-----------|
| Directly Driven |
| NN         | 0.33        | 0.49        | 998.7         | 355.08        | 0.00055   | 0.00035   |
| SVM        | 0.1         | 0.45        | 845.4         | 275.8         | 0.00073   | 0.000707  |
| ELM        | 0.19        | 0.45        | 1566.4        | 946.06        | 0.00073   | 0.000707  |
| KELM       | 0.1498      | 0.3892      | 1460          | 810           | 0.00043   | 0.00022   |
| PSO-KELM   | 0.1006      | 0.347       | 1222          | 466.05        | 0.00067   | 0.00049   |
| Remotely Driven |
| NN         | 2.01        | 2.22        | 1.74E+03      | 1032          | 5.50E-04  | 4.34E-04  |
| SVM        | 0.73        | 1.18        | 1.50E+03      | 946.06        | 3.90E-04  | 3.90E-04  |
| ELM        | 1.90        | 1.58        | 1.50E+03      | 946.06        | 3.90E-04  | 3.90E-04  |
| KELM       | 1.17        | 1.52        | 1.60E+03      | 987.73        | 4.40E-04  | 1.80E-04  |
| PSO-KELM   | 1.03        | 1.11        | 3.32E+03      | 1.07E+03      | 3.50E-04  | 7.50E-04  |
| Controller | time (in seconds) | $e_M$ | $e_F$ |
|------------|------------------|-------|-------|
| NN         | 40.17            | 0.0084| 0.0343|
| SVM        | 4.66             | 0.0019| 0.0074|
| ELM        | 0.567            | 0.0023| 0.0077|
| KELM       | 0.055            | 0.0022| 0.0056|
| PSO-KELM   | 16               | 0.0011| 0.0007|

By observing the results it can be observed that in joint 1, SVM is having the minimum tracking error index. It is having error lesser than KELM and even PSO-KELM. Also, the tracking error of SVM and ELM is almost same, but the tracking error reduces in KELM and it further reduces in PSO-KELM. Moreover, the tracking error for neural network based control technique is found to be highest. Control input torque indices tabulated in tables concludes that the control input torques reduce in PSO based KELM when compared to KELM and ELM. One of the major advantages of being reduced execution time in ELM based control technique can be observed. It can be observed in these results that the neural network based control technique is having the highest execution time, followed by PSO-KELM (table 3). In fact KELM based control technique has the minimum execution time. Moreover, transient error and steady state error (table 3) are found to be minimum in proposed controller i.e. PSO-KELM.

6. Conclusions
In this study, a novel PSO-KELM, having hybrid of KELM and PSO has been proposed. In PSO-KELM algorithm PSO approach has been used to get the optimal value of the kernel parameters in KELM. It has been observed from the outcomes of the comparisons of the proposed controller, that PSO-KELM is able to cope up with the unknown dynamics and the external disturbances in the robotic manipulator system. These proposed controllers not only overcome the drawbacks of some of the well known intelligent techniques but also enhances the control performance even while incorporating the actuator dynamics too. The proposed control schemes can also achieve better generalization performance with directly driven and remotely driven robotic manipulators.

References
[1] S. Purwar, I. N. Kar and A. N. Jha.: Neuro-sliding mode control of robotic manipulators, IEEE International Conference on Robotics, Automation and Mechatronics, pp. 595-600, Singapore, Dec. 2004.
[2] A. Ishiguro, T. Furuhashi, S. Okuma and Y. Uchikawa.: A neural network compensator for uncertainties of robotic manipulators, IEEE Transactions on Industrial Electronics, vol. 38, pp. 565–70, Dec. 1992.
[3] Y. Fang, T. W. S. Chow and X. D. Li.: Use of a recurrent neural network in discrete sliding-mode control, IEEE—Control Theory and Applications, vol. 146, no. 1, pp. 84–90, Jan. 1999.
[4] C.-M. Lin and C.-F. Hsu.: Neural-network hybrid control for antilock braking systems, IEEE Transactions on Neural Networks, vol. 14, no. 2, pp. 351–59, Mar. 2003.
[5] G.B. Huang, L. Chen, C.K. Siew.: Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Transactions on Neural Networks, 17 (2006) 879-92.
[6] D. Zu, T. Mei, M. Luo, K. Guan and D. Zhu.: Fuzzy SVM Controller for Robotic Manipulator Based on GA and LS Algorithm, Sixth International Conference on Fuzzy Systems and Knowledge Discovery, pp. 268-71, 2009.
[7] Z. Yibo and K. Liuting.: A Control Method for Flexible Manipulator Based on SVM, Sixth International Conference on Natural Computation (ICNC 2010), pp. 3462-64, 2010.
[8] D. Zu, T. Mei and L. Sun.: Fuzzy Support Vector Machines Control for 6-DOF Parallel Robot,
[9] G. B. Huang, Q.Y. Zhu, C.K. Siew.: Extreme Learning Machines, Pattern Recognition, 38(2006)1759-63.
[10] G.B. Huang, Q.Y. Zhu, C.K. Siew.: Extreme learning machine: theory and applications, Neurocomputing, vol. 70, pp. 489-501, 2006.
[11] G-B Huang, H. Zhou, X. Ding and R. Zhang.: Extreme learning machine for regression and multiclass classification, IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics,” Vol. 42, No. 2, pp. 513-29, 2013.
[12] R. Rajesh and J. Siva Prakash.: Extreme learning machines - a review and state-of-the-art, International Journal of Wisdom Based Computing, vol. 1, pp. 35–49, 2011.
[13] Huang G-B, Babri HA.: General approximation theorem on feed forward networks, IEEE proceedings of international conference on information, communications and signal processing, ICICS ‘97, Singapore, pp 698–702,(1997).
[14] Huang G-B, Chen L, Siew C-K.: Universal approximation using incremental feed forward networks with arbitrary input weights. Technical Report ICIS/46/2003, School of Electrical and Electronic Engineering, Nanyang Technological University, October, (2003) , Singapore
[15] Serre D.: Matrices: theory and applications. Springer,(2002) , New York.
[16] Rao CR, Mitra SK.: Generalized inverse of matrices and its applications. Wiley,(1971) , New York.
[17] Kennedy, J. and Eberhart, R.: Particle Swarm Optimization, IEEE Int. Conf. Neural Networks,(1995) , pp. 942-1948.
[18] Goldberg, D. E.: Genetic algorithms in search, optimization, and machine learning, MA: Addison Wesley, (1989).
[19] Neha Kapoor and Jyoti Ohri.: Improved PSO Tuned Classical Controllers (PID and SMC) for Robotic Manipulators, International Journal of Modern Education and Computer Science, MECS Publications, vol. 1, pp. 47-52, Jan 2015. DOI: 10.5815/ijmecs.2015.01.07.
[20] S.F. Ding, Y.A. Zhang, X.Z. Xu, and L.N. Bao.: A novel extreme learning machine based on hybrid kernel function, Journal of Computers, vol.8, no.8, pp. 2110–17, 2013.
[21] M. W. Spong and M. Vidyasagar, Robot Dynamics and Control, Wiley-India Edition, New York.
[22] Neha Kapoor and Jyoti Ohri.: Fuzzified PSO-SVM (fP-SVM) Controller for Motion Control of Robotic Manipulator, International Journal of Industrial Systems and Engineering, Inderscience Publications.(Accepted for publication, In Press).
[23] Neha Kapoor and Jyoti Ohri.: A Neural Network Based Novel Approach for Error Optimization in Path Tracking Control of a Robotic Manipulator, Proc. of National Conference Advancements in the Era of Multi Disciplinary Systems, AEMDS-2013, pp. 78-83, April 6-7, 2013. Published by Elsevier-2013. (ISBN: 978-93-5107-057-3).
[24] Neha Kapoor and Jyoti Ohri.: PD-SVM Integrated Controller for Robotic Manipulator Tracking Control, Chinese Journal of Engineering, HindawiPublishing Corporation,vol.2014,ArticleID295210, pages 6, http://dx.doi.org/10.1155/2014/295210. (ISSN: 2314-8063 (Online)).
[25] B. A. 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, pp.1083-183, 1994.
[26] L. Xu and B. Yao.: Adaptive Robust Precision Motion Control of Linear Motors with Negligible Electrical Dynamics, IEEE/ASME Transactions on Mechatronics, vol. 6, no. 4, December 2001, pp. 444-52.