Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 125 (2018) 42–49

Procedia Computer Science

www.elsevier.com/locate/procedia

6th International Conference on Smart Computing and Communications, ICSCC 2017, 7-8 December 2017, Kurukshetra, India

Design of Intelligent Hybrid Force and Position Control of Robot Manipulator

Komal Rani[a,b], Naveen Kumar[a,]*

[a]*Department of Mathematics, National Institute of Technology Kurukshetra, Haryana-136119, India*
[b]*Department of Mathematics, Babu Anant Ram Janta College Kaul,Kaithal, Haryana-136021, India*

Abstract

This work considers the Hybrid Force/Position control of robot manipulator in the presence of uncertainties and external disturbances. The proposed controller contains the model based term, Radial Basis Function neural network term plus an adaptive bound part. The Radial basis function neural network is functioning to learn a non linear function with no requirement of off line training. An adaptive bound part is developed to guess the unknown bound on the unmodeled disturbance, neural network reconstruction error and friction term. The Lyapunov function approach is used to the stability of the system. In the end simulations results are presented for two link robot manipulators.

Peer-review under responsibility of the scientific committee of the 6th International Conference on Smart Computing and Communications.

Keywords: *Hybrid Force and Position control; Interection force; RBF neural network; Lyapunov Stability;*

1. Introduction

There are many industrial applications such as grinding, fine polishing, contour following, deburring and assembly in which the robot end effector comes in the contact with the environment. During the implementation of

* Corresponding author. Tel.: +91-9997876560; fax: +0-000-000-0000 .
E-mail address: navindma@gmail.com

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 6th International Conference on Smart Computing and Communications
10.1016/j.procs.2017.12.008

such types of tasks robot contact forces and robot position should be controlled. The Hybrid Force and Position control of robot manipulators has been studied by many researchers. Robert and Craig [1] proposed the hybrid force/ position control scheme in which dynamic decoupling among each of the robot joints is neglected. They designed the controller within the frame work of robot joint control system. Lozano and Brolioto [2] presented an adaptive controller for robot manipulator with redundant degree of freedom. This approach does not require measurement of the joint acceleration and the force first derivative. Yoshikwa and sudou [3] extended the work of Raibert and Craig. They consider the dynamics of manipulator and constraint on the end effector and designed a control algorithm. Kwan [4] designed the robust adaptive controller to control force/motion of the constrained rigid robot together with motor dynamics. In the robust approaches, the controller has fixed structure that produced an suitable performance for dynamics. Kouya et al. [5] has given an adaptive force/position controller by utilizing the strict feedback back-stepping method, depended on passivity. De Queiroz et al. [6] developed adaptive force/position controller for robot manipulator through the constrained motion with no velocity measurement. Cheah et al. [7] developed a motion/ force controller for robot with unsure dynamics and kinematics. Roy and Whitcomb [8] gave an idea of an adaptive force control method by low level position/velocity controller for robot arm in contact with surface of unidentified linear compliance. Filaretov and Zuev [9] created a method by not using force/moment sensor and supplementary device to manage the force used by robot end effect on the environment. Kouya [10] designed a hybrid force/position controller for robot manipulators with uncertainty which comes in the contact of environment. We find that this techniques need the information of complex regression matrices and problems are faced by unmodeled disturbances.

In recent times, neural networks have all-round characteristics for example learning ability and nonlinear mapping due to these it have achieved great popularity among community of control system. The initiative for neural network based control is that it is able to compensate unknown dynamics. Various approaches based on neural network are available in the literature for control of robot manipulators [11–12]. Karayiannidis et al. [13] presented a neuro adaptive controller of robot manipulator interacting with a flat surface when there exists many non parametric uncertainties in the dynamics. Bechlioulis et al. [14] designed an adaptive force/position controller based on neural network with the uncertainties in the dynamical model and bounded disturbances. Kumar et al. [15] designed an adaptive controller by taking into consideration the decomposition of dynamics of robot. Neural network was used to find out the unknown dynamics. Singh and Sukavanam [16] designed the robust adaptive controller based on neural network to achieve the stability of system. Li et al. [17] developed a hybrid position/force control scheme based on adaptive neural network for a constrained reconfigurable manipulator. For reducing the complication of the dynamic model a reduced model for reconfigurable manipulators is designed. Gajhar et al. [18] presented hybrid position/force controller for a constrained robot manipulator. The controller is made of two parts in which one part fulfils the objective of motion and force tracking and another part is used to compensate for the deficiencies of the CT controller.

In this paper, neural networks based hybrid force/position control method is developed for a constrained rigid robot manipulators in the presence of uncertainties. Generally, all hybrid force/position controllers which are based on neural network assume no prior knowledge about the dynamics of the system. Actually, some information is always available about the system dynamics. By using this facts of robot dynamics, we design the controller in which unknown dynamics part is compensated by using the RBF neural network. The whole system attains the stability in the sense of Lyapunov.

The paper is prepared as follow. We present the dynamical form of rigid robot manipulator and its decomposition along with two assumptions and useful properties related to dynamics in Section 2. In Section 3, the RBF neural network plus model based controller is created. The stability testing for the closed-loop error system is presented in Section 4. The simulation results are presented in Section 5.

2. Dynamical Model for Rigid Robot Manipulators

The general equation describing the dynamics of n link revolute rigid robot manipulators with environment contact is given by [16]

$$M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + F_e(\dot{q}) + G(q) + T_d = \tau - \tau_e \tag{1}$$

Where $M(q) \in R^{n\times n}$ is the inertia matrix. $V_m(q,\dot{q}) \in R^{n\times n}$ stands for the centripetal coriolis matrix. $G(q) \in R^n$ is the gravity effect. $F_e(\dot{q}) \in R^n$ is the friction effects. $\tau \in R^n$ represents the torque input vector. $T_d \in R^n$ represents the bounded unknown disturbances and $\tau_e \in R^n$ is the interaction torque which makes contact with environment. The vector $q(t), \dot{q}(t)$ and $\ddot{q}(t) \in R^n$ are the links position, velocity and acceleration respectively. The dynamical equation given by eq. (1) has the useful properties 1 and 2.

Property 1- The matrix $M(q)$ is symmetric as well as positive definite and satisfies the following inequalities

$$\alpha_m \leq M(q) \leq \beta_m, \qquad \alpha_m \ \& \ \beta_m \text{ are positive constant.}$$

Property 2-The matrix $\dot{M}(q) - 2V_m(q,\dot{q})$ is skew- symmetric.

The following assumptions are also useful in the development of the controller:

Assumption 1: For some unknown positive constants b and c, we assume that

$$||F_e(\dot{q})|| = b + c||\dot{q}||$$

Assumption 2: We assume that $||T_d|| \leq d$ for some positive constant d.

The interaction torque $\tau_e \in R^n$ in the joint space is related to interaction force $\bar{F} \in R^m$ at the end-effector by

$$\tau_e = H^T(q)\bar{F} \tag{2}$$

Where Jacobian matrix $H(q)$ belongs to $R^{m\times n}$. The robot kinematic equation is defined by

$$x = f(q) \tag{3}$$

Where the end-effector position vector is $x \in R^m$ and the kinematic transformation for the manipulator is $f(q)$. Derivative of eq. (3) becomes

$$\dot{x} = H(q)\dot{q} \tag{4}$$

In the task space, the deformation of environment i.e. $(x - x_c)$ is related to the interaction force $\bar{F}$ by the relation

$$\bar{F} = \bar{P}(x - x_c) \tag{5}$$

where $\bar{P} \in R^{m\times m}$ is the stiffness matrix which is constant and $x_c \in R^m$ is the point of contact. We rearrange the terms of $\bar{F}$ in above equation and write the new equation as:

$$\bar{F} = \begin{bmatrix} F \\ F' \end{bmatrix} = \begin{bmatrix} P \\ P' \end{bmatrix}(x - x_c) \tag{6}$$

where $F \in R^m$, $F' \in R^{m-s}$, $P \in R^{s\times m}$ where $s \leq m$ and has full rank . $P' \in R^{(m-s)\times m}$ depends on rows of matrix P. Here s is dimension for the subspace of the task space R^m where we have to control forces. The system has n control input which are less than and equal to six. So, we cannot control force as well as position of the robot in all dimensions. Therefore to control the system we control only a small subspace i.e. s forces, $[m-s]$ positions and $[n-m]$ redundant velocities of joints. Therefore dimension of the subspace is $s + [m-s] + [n-m] = n$. Here we assume that $H(q)$ is a full rank matrix.

2.1. Decomposition of dynamical model

The dynamics given by (1) can be decomposed into position, force and the redundant joints [15]. We use the following identities in this process:

$$I = H^+H + H^- \tag{7}$$

$$I = P^+P + P^- \tag{8}$$

Where

$$H^+ = H^T[HH^T]^{-1} \tag{9}$$

$$P^+ = P^T[PP^T]^{-1} \tag{10}$$

$$H^- = I - H^T[HH^T]^{-1}H \tag{11}$$

$$P^- = I - P^T[PP^T]^{-1}P \tag{12}$$

Where H^+ and P^+ are denotes pnerose pseudo inverse, H^- and P^- are projectors.

With eq. (4) and eq. (7), $\dot{q}$ is decomposed as

$$\dot{q} = H^+H\dot{q} + H^-\dot{q} = H^+\dot{x} + H^-\dot{q} \tag{13}$$

Taking eq. (8) and using relation given by (6), $\dot{x}$ can be written as

$$\dot{x} = P^+P\dot{x} + P^-\dot{x} = P^+\dot{F} + P^-\dot{x} \tag{14}$$

Differentiating eq. (4), we have

$$\ddot{x} = H\ddot{q} + \dot{H}\dot{q} \tag{15}$$

Taking eq. (7) with above relation eq. (15), $\ddot{q}$ is written as

$$\ddot{q} = H^+H\ddot{q} + H^-\ddot{q} = H^+(\ddot{x} - \dot{H}\dot{q}) + H^-\ddot{q} \tag{16}$$

Taking derivative of eq. (14), we have

$$\ddot{x} = P^+P\ddot{x} + P^-\ddot{x} = P^+\ddot{F} + P^-\ddot{x} \tag{17}$$

Using eq. (13) to (15), eq. (1) is rewritten as

$$M\left(H^{+}\left[P^{+}\ddot{F} + P^{-}\ddot{x} - \dot{H}\dot{q}\right] + H^{-}\ddot{q}\right) + V_m\left(H^{+}\left[P^{+}\dot{F} + P^{-}\dot{x}\right] + H^{-}\dot{q}\right) + G + F_e + T_d = \tau - \tau_e \quad (18)$$

Now this eq. is clearly decomposition of F, $P^{-}x$ and $H^{-}\dot{q}$ which are force, position and the redundant velocity respectively and similar to eq. (1).

3. Controller Design Based on Neural Network

In this section, our goal is to develop the control torque input τ to attain the desired force and to control the robot end-effector position.

3.1. Error system development

Define the filtered tracking error r as

$$r = H^{+}\left[P^{+}\left(\dot{F} - v_f\right) + P^{-}\left(\dot{x} - v_x\right)\right] + H^{-}\dot{\tilde{q}}) \quad (19)$$

Where

$$v_f = \dot{F}_s - \lambda\tilde{F};\ v_x = \dot{x}_s - \lambda\tilde{x} \quad (20)$$

$$\tilde{F} = F - F_s;\ \tilde{x} = x - x_s;\ \tilde{q} = q - q_s \quad (21)$$

Where $\lambda > 0$ and F_s, x_s, q_s are desired values of F, x, q. Now using eqs. (19) to (21),we write eq. (18) in terms of filtered tracking error r as

$$M\dot{r} = -V_m r + \tau - \tau_e - h(y) - F_e - T_d \quad (22)$$

Where $h(y) = M\left(H^{+}\left[P^{+}\dot{v}_f + P^{-}\dot{v}_x - \dot{H}\dot{q}\right] + H^{-}\ddot{q}_s - Z\right) + V_m(H^{+}\left[P^{+}v_f + P^{-}v_x\right] + H^{-}\dot{q}_s)) + G$ is a nonlinear function. Here $Z = \left(\frac{dH^{+}}{dt}\right)\left(P^{+}\left(\dot{F} - v_f\right) + P^{-}\left(\dot{x} - v_x\right)\right) + \left(\frac{dH^{-}}{dt}\right)\dot{\tilde{q}}$. We write the function $h(y)$ as mixture of two parts corresponding to unknown dynamics and known dynamics as follows

$$h(y) = \hat{h}(y) + \tilde{h}(y) \quad (23)$$

Where $\hat{h}(y) = \hat{M}\left(H^{+}\left[P^{+}\dot{v}_f + P^{-}\dot{v}_x - \dot{H}\dot{q}\right] + H^{-}\ddot{q}_s - Z\right) + \hat{V}_m(H^{+}\left[P^{+}v_f + P^{-}v_x\right] + H^{-}\dot{q}_s)) + \hat{G}$ and $\tilde{h}(y) = \tilde{M}\left(H^{+}\left[P^{+}\dot{v}_f + P^{-}\dot{v}_x - \dot{H}\dot{q}\right] + H^{-}\ddot{q}_s - Z\right) + \tilde{V}_m(H^{+}\left[P^{+}v_f + P^{-}v_x\right] + H^{-}\dot{q}_s)) + \tilde{G}$ and the vector y is given by $y = [r, F, q, \dot{q}]$.

3.2. Radial Basis Function Neural Network

A radial basis function network is a particular type of neural networks. It is utilized for approximating the unknown part of dynamics equation. A RBF network with centres c_i and width σ_i for approximation of unknown part is defined as follows:

$$\tilde{h}(y) = W^{T}\varsigma(y) + \varepsilon(y) \quad (24)$$

where $W \in R^{N\times n}$ symbolizes the matrix of weights of neural network, $\varepsilon(\cdot) : R^{3n+1} \rightarrow R^{N}$ represents the neural network reconstruction error which depends on *N*. For large value of *N*, we have $||\varepsilon(y)|| \leq \varepsilon_N$ for some $\varepsilon_N > 0$ and $\varsigma(\cdot) : R^{3n+1} \rightarrow R^{N}$ is a smooth basis function defined element wise as

$$\varsigma_i(y) = \exp\left(-\frac{||y - c_i||^2}{\sigma_i}\right) \qquad i=1,2,\ldots,N \quad (25)$$

N denotes the no. of nodes of neural network.

Now substituting the value of $h(y)$ from eq. (23) in eq. (22) and using eq. (24), we get

$$M\dot{r} = -V_m r + \tau - \tau_e - \hat{h}(y) - W^{T}\varsigma(y) - \varepsilon(y) - F_e - T_d \quad (26)$$

3.3. Adaptive bound

Considering the assumption 1 and 2 and taking upper bound ε_N of neural network reconstruction error, we have

$$||F_e + T_d + \varepsilon(y)|| \le b + c||\dot{q}|| + d + \varepsilon_N \tag{27}$$

We define the adaptive bound β, which can be written in the following form

$$\beta = b + c||\dot{q}|| + d + \varepsilon_N = Q^T(||\dot{q}||)\phi \tag{28}$$

where $Q \in R^k$ is known vector of joint velocity and $\phi \in R^k$ is a parameter vector, k is fixed positive integer. To achieve the desired goal, we design the controller τ as

$$\tau = \tau_e + \hat{h}(y) - K_d r + \widehat{W}\,\varsigma(y) - \left(\frac{\hat{\beta}}{||r||}\right) r \tag{29}$$

where $K_d = K_d{}^T \in R^{n\times n}$ is a gain matrix, $\hat{\beta} = Q^T\hat{\phi}$, $\hat{\phi}$ and $\widehat{W}$ are approximated values of ϕ and W respectively provided by tuning algorithm.

Using the controller given by eq. (29) and eq. (26), the error dynamics is written as

$$M\dot{r} = -(K_d + V_m)r - \widetilde{W}^T\varsigma(y) - \left(F_e + T_d + \varepsilon(y)\right) - \left(\frac{\hat{\beta}}{||r||}\right) r \tag{30}$$

where $\widetilde{W} = W - \widehat{W}$.

4. Stability Analysis

Here we show the stability of the dynamic system using Lyapunov function.

Theorem: If the robot dynamic is presented by eq. (1), the control input to the system is given by eq. (29) and adaptation laws for neural network weights and adaptive bounds are selected by eq. (31) and eq. (32) respectively, then the error in force, position and redundant velocity i.e. $\tilde{F}$, $(P^- \tilde{x})$, $H^-\dot{\tilde{q}}$ respectively converges to zero when $t \to \infty$.

$$\dot{\hat{W}} = -\Gamma_W\varsigma(y)\,r^T \tag{31}$$

$$\dot{\hat{\phi}} = \Gamma_\phi Q||r|| \tag{32}$$

where $\Gamma_W = \Gamma_W{}^{-1} \in R^{N\times N}$ and $\Gamma_\phi = \Gamma_\phi{}^{-1} \in R^{k\times k}$ are positive definite matrices.

Proof: Let the Lyapunov candidate be

$$L = \frac{1}{2}r^T M r + \frac{1}{2}tr\left(\widetilde{W}^T\Gamma_W{}^{-1}\widetilde{W}\right) + \frac{1}{2}tr\left(\tilde{\phi}^T\Gamma_\phi{}^{-1}\tilde{\phi}\right) \tag{33}$$

where $\widetilde{W} = W - \widehat{W}$ and $\tilde{\phi} = \phi - \hat{\phi}$

Differentiating the Lyapunov function L, we have

$$\dot{L} = \frac{1}{2}r^T\dot{M}r + r^T M\dot{r} + tr\left(\widetilde{W}^T\Gamma_W{}^{-1}\dot{\widetilde{W}}\right) + tr\left(\tilde{\phi}^T\Gamma_\phi{}^{-1}\dot{\tilde{\phi}}\right) \tag{34}$$

Now using the eq. (30) and the facts $\dot{\widetilde{W}} = \dot{\hat{W}}$, $\dot{\tilde{\phi}} = -\dot{\hat{\phi}}$, we get the following expression

$$\dot{L} = \frac{1}{2}r^T\dot{M}r + r^T\left(-K_d r - V_m r - \widetilde{W}^T\varsigma(y) - \left(F_e + T_d + \epsilon(y)\right) - \left(\frac{\hat{\beta}}{||r||}\right)r\right) - tr\left(\widetilde{W}^T\Gamma_W{}^{-1}\dot{\hat{W}}\right) - tr\left(\tilde{\phi}^T\Gamma_\phi{}^{-1}\dot{\hat{\phi}}\right) \tag{35}$$

By using the tuning algorithm given by eq. (31) and eq. (32), the eq. (35) becomes

$$\dot{L} = \frac{1}{2}r^T\dot{M}r - r^T K_d r - r^T V_m r - r^T\widetilde{W}^T\varsigma(y) - r^T\left(F_e + T_d + \epsilon(y)\right) - \left(\frac{\hat{\beta}}{||r||}\right)r^T r + tr\left(\widetilde{W}^T\Gamma_W{}^{-1}\Gamma_W\varsigma(y)r^T\right) - tr\left(\tilde{\phi}^T\Gamma_\phi{}^{-1}\Gamma_\phi Q||r||\right) \tag{36}$$

$$\dot{L} = \frac{1}{2}r^T\left(\dot{M} - 2V_m\right)r - r^T K_d r - r^T\left(F_e + T_d + \epsilon(y)\right) - \left(\frac{\hat{\beta}}{||r||}\right)||r||^2 - tr\left(\tilde{\phi}^T Q||r||\right) \tag{37}$$

Using property 2, we have

$$\dot{L} = -r^T K_d r - r^T\left(F_e + T_d + \epsilon(y)\right) - \hat{\beta}||r|| - tr\left(\tilde{\phi}^T Q||r||\right) \tag{38}$$

Using eq. (27) and eq. (28), we find the relation of the form given by eq. (39)

$$\left\|r^T\left(F_e + T_d + \epsilon(y)\right)\right\| = ||r||\,||F_e + T_d + \epsilon(y)|| \le ||r||\beta = ||r||Q^T\phi = ||r||Q^T\left(\tilde{\phi} + \hat{\phi}\right) \tag{39}$$

Taking norm on both sides and using eq. (39), we have

$$\dot{L} \le -r^T K_d r + ||r||Q^T\left(\tilde{\phi} + \hat{\phi}\right) - \hat{\beta}||r|| - tr\left(\tilde{\phi}^T Q||r||\right)$$

$$\dot{L} \leq -r^T K_d r \tag{40}$$

This implies $\dot{L} \leq 0$. Also we have $L > 0$, this proves the stability of the system so that $r, \widetilde{W}$ and $\widetilde{\phi}$ are bounded. Hence $\widehat{W}, \widehat{\phi}$ are bounded.

Using eq. (6) in eq. (19), we have

$$r = H^{+}\left[P^{+}\left(P\dot{x} - v_f\right) + P^{-}\ \ (\dot{x} - v_x)\right] + H^{-}\dot{\tilde{q}}) \tag{41}$$

Using eq. (8), we have

$$r = H^{+}\left[\dot{x} - P^{+}v_f - P^{-}\ v_x)\right] + H^{-}\dot{\tilde{q}}) \tag{42}$$

$$r = H^{+}\dot{x} - H^{+}\left(P^{+}v_f + P^{-}\ v_x\right) + H^{-}\dot{q} - H^{-}\dot{q}_s) \tag{43}$$

using eq. (13), we have

$$r = \dot{q} - J^{+}\left(P^{+}v_f + P^{-}\ v_x\right) - H^{-}\dot{q}_s) \tag{44}$$

Since we have considered that the joints are revolute, therefore x, F are bounded and this implies v_f and v_x are bounded given by eq. (20). Now q_s is also bounded. So, we have $\dot{q}$ is bounded. Using Property 1 and taking derivative of r given by eq. (19), we find that $\dot{r}$ is also bounded since the joint are revolute. Hence, $r \to 0$ as $t \to \infty$.

Now, P and H are full rank matrix and bounded. Multiplying eq. (19) by PH, we have

$$PHr = \dot{F} - v_f \to 0, \quad \text{since } r \to 0. \tag{45}$$

Multiplying eq. (19) by H, we get

$$Hr = P^{-}(\dot{x} - v_x) \to 0 \tag{46}$$

Now from eqs. (19), (44) and (45), we have

$$H^{-}\dot{\tilde{q}} \to 0 \tag{47}$$

Now from eq. (45), eq. (46) and using eq. (20) and eq. (21), we get the input terms $\dot{\tilde{F}} + \lambda\tilde{F}$ and $P^{-}\dot{\tilde{x}} + \lambda P^{-}\tilde{x}$ are converging to zero whose outputs are considered as $\tilde{F}$ and $P^{-}\tilde{x}$ respectively. Hence, we have the result that the errors $\tilde{F} \to 0$, $P^{-}\tilde{x} \to 0$ and $H^{-}\dot{\tilde{q}} \to 0$.

5. Simulations Studies

In this section, a two link robot manipulator is used to perform simulation whose mathematical expression can be found in [15]. Let m_1, m_2 and l_1, l_2 are the masses and lengths of link 1,2 respectively whose values are defined as $m_1 = 15.61\ kg, m_2 = 11.36\ kg, l_1 = 0.432m, l_2 = 0.432m, g_0 = 9.8m/s^2$.

We take the contact force $F = \kappa(x - x_{1e})$ in the x_1 direction where $\kappa = 10^4\ and\ x_{1e} = 0.61m$ and the desired trajectory $x_{2s} = 0.5[1 - \exp(-t)]$ in x_2 direction.

The force $F_s = 20(1 - 0.5\sin t)$.

Also, $P^{+} = 1/\kappa\,, P^{-} = [0\ 0\ ; 0\ 1], H^{+} = H^{-1}\ and\ H^{-} = 0$.

The controller parameters are $\lambda = 100, K_d = 200I_2$.

The friction term is taken as $F_e(\dot{q}) = 5.3\dot{q}_1 + 2.4\dot{q}_2$ and unknown disturbance $T_d = 80[\cos(2t)\ \sin(2t)]^T$.

We take ten nodes for RBF neural network. The centre of the Gaussian type function are taken from $[-2, 2]^{10}$ and width σ_i are taken to be 0.2.The matrices in the weight updates law are $\Gamma_W = 10I_{10}\ and\ \Gamma_\phi = 5I_4$. The simulation is exposed for 15 secs. The presentation of the proposed controller is obvious from the figs. 1-5. The fig.1 shows the trajectory tracking of the end effector of the robot manipulator. It is clear from fig. 2 that error of force tracking quickly converging to zero. The fig. 3 gives the trajectory of the desired force and the actual force. The joint trajectory and joint velocity trajectory tracking is given by fig. 4 and fig. 5 respectively.

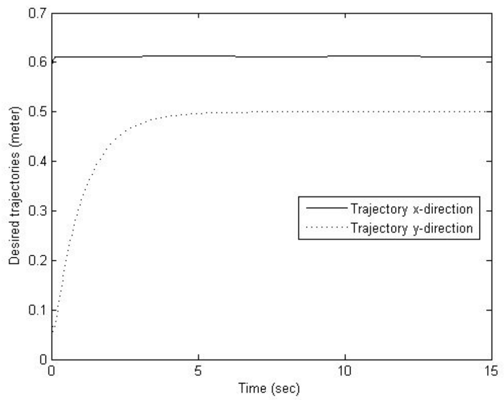

Fig. 1. End- effector trajectory tracking .

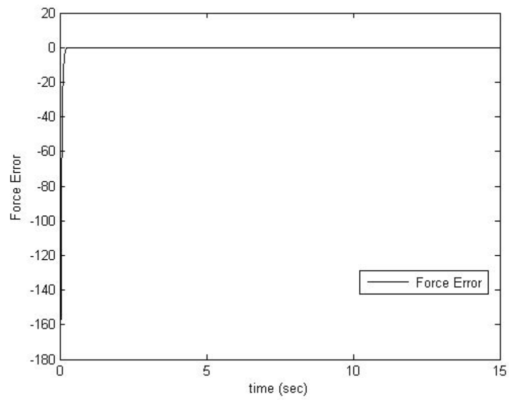

Fig. 2. Force tracking error .

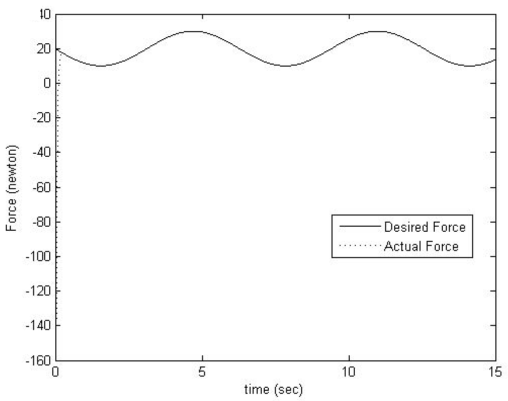

Fig. 3. Desired and actual force tracking.

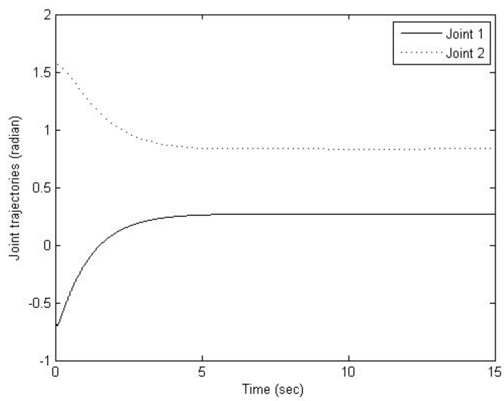

Fig. 4. Joint trajectory with proposed controller.

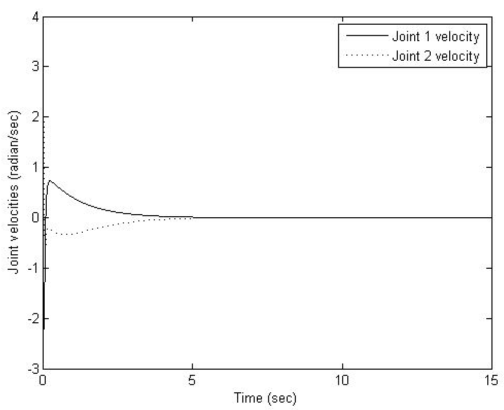

Fig. 5. Joint velocity.

6. Conclusion

We have created the controller which combines the model based techniques and RBF neural network based learning approach to enhance the performance of controller. The controller learn the unstructured and structured uncertainties of the dynamics in the online way. The RBF is utilized to find out the unknown dynamic part of the system and the adaptive bound part estimates the bounds on unmodeled dynamics and neural network approximation error. The whole system is shown stable taking Lyapunov candidate, made by weighting matrices. At last the simulation results are produced for the case of two link robot manipulator.

References

[1] M.H. Raibert, and J.J. Craig. (1981) “Hybrid Position/Force Control of Manipulators.” ASME Journal of Dynamical Systems, Measurement and Control **103(2)**: 126-133.

[2] R. Lozano, and B. Brogliato. (1992) “Adaptive hybrid position/force control of redundant manipulators.” IEEE Trans Autom Control **37(10)**: 1501–1505.

[3] T. Yoshikawa, and A. Sudou. (1993) “Dynamic Hybrid Position/Force Control of Robot Manipulators: Online-Estimation of Unknown Constraint.” IEEE Transactions on Robotics and Automation **9(2)**: 220-226.

[4] C.M. Kwan. (1996) “Robust Adaptive Force/Motion Control of Constrained Robots.” IEE Proc.-Control Theory and Applications **143(1)**: 103-109.

[5] D.N. Kouya, M. Saad, and L. Lamarche. (2002). “Backstepping Adaptive Hybrid Force/Position Control for Robotic Manipulators.” Proc. of the American Control Conference 4595-4600.

[6] M.S. De Queiroz, H. Jun, and D.M. Dawson. (1997) “Adaptive Position/Force Control of Robot Manipulators without Velocity Measurements: Theory and Experimentation.” IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics **27(5)**: 796-809.

[7] C.C. Cheah, Y. Zhao, and J.J.E. Slotine. (2006) “Adaptive Jacobian Motion and Force Control for Constrained Robots with Uncertainties.” Proc. of the International Conference on Robotics and Automation 2226-2231.

[8] J. Roy, and L.L. Whitcomb. (2002) “Adaptive force control of position/velocity controlled robots: theory and experiment.” IEEE Trans Robot Autom **18(2)**: 121-137.

[9] V.F. Filaretov, and A.V. Zuev. (2008) “Adaptive force/position control of robot manipulators.” Proceedings of the 2008 IEEE/ASME international conference on advanced intelligent mechatronics 96–101.

[10] D.N. Kouya. (2008) “Adaptive hybrid force-position control of a robotic manipulator.” 2nd European computing conference 323–331.

[11] F.L. Lewis, S. Jagannathan, and A. Yesildirek. (1999) “Neural network control of robot manipulators and nonlinear systems.” Taylor and Francis, UK.

[12] F.L. Lewis, A. Yesildirek, and K. Liu. (1996) “Multilayer neural net robot controller with quaranted tracking performance.” IEEE Trans Neural Netw **7(2)**: 388–399.

[13] Y. Karayiannidis, K. Rovithakis, and Z. Doulgeri. (2007) “Force/Position Tracking for a Robotic Manipulator in compliant contact with a surface using Neuro-adaptive Control.” Automatica **43(7)**: 1281-1288.

[14] C.P. Bechlioulis, Z. Doulgeri, and G.A. Rovithakis. (2010) “Neuroadaptive force/position control with prescribed performance and guaranteed contact maintenance.” IEEE Trans Neural Network **21(12)**: 1857–1868.

[15] N. Kumar, V. Panwar, N. Sukavanam, S.P. Sharma, and J.H. Borm. (2011) “Neural network based hybrid force/position control for robot manipulators.” Int J Precis Eng Manuf **12(3)**: 419–426.

[16] H.P. Singh, and N. Sukavanam. (2013) “Stability analysis of robust adaptive hybrid position/force controller for robot manipulators using neural network with uncertainties.” Neural Computing and Applications **22(7–8)**: 1745–1755.

[17] Y. Li, G. Wang, B. Dong, and B. Zhao. (2015) “Hybrid position–force control for constrained reconfigurable manipulators based on adaptive neural network.” Advances in Mechanical Engineering **7(9)**: 1–10.

[18] M.H. Ghajar, M. Keshmiri, and J. Bahrami. (2017) “Neural-network-based robust hybrid force/position controller for a constrained robot manipulator with uncertainties.” Transactions of the Institute of Measurement and Control 1–12.