

Fuzzy-neural network compensator for Robot manipulator controlled by PD-like fuzzy system

Prof. Dr. Turki Y. Abdalla^{*}, Dr. Basil H. Jasim^{**}

^{*}Department of computer engineering, college of engineering

^{**}Department of electrical engineering, college of engineering
University of basrah

Abstract

In this paper, high tracking performance control structure for rigid robot manipulator is proposed. PD-like Sugano type fuzzy system is used as a main controller, while fuzzy-neural network (FNN) is used as a compensator for uncertainties by minimizing suitable function. The output of FNN is added to the reference trajectories to modify input error space, so that the system robust to any change in system parameters. The proposed structure is simulated and compared with computed torque controller. The simulation study has showed the validity of our structure, also showed its superiority to computed torque controller.

د. تركي يونس^{*}

د. باسل هاني جاسم^{**}

^{*}قسم هندسة الحاسبات، كلية الهندسة

^{**}قسم الهندسة الكهربائية، كلية الهندسة

جامعة البصرة

الخلاصة

في هذا البحث، قُدمت منظومة سيطرة عالية الاداء في متابعة إشارة مرجعية معينة وذلك لذراع الروبوت. أُستُخدم النظام المضطرب شبيه PD من النوع سيكون لتصميم المسيطر الاساسي، بينما أُستُخدمت شبكة مضطربة-عصبية كموازن لعدم الوثوقية والثبات في عناصر النظام وذلك من خلال التصغير الامثل لدالة معينة. أُخرجات الموازن تضاف الى الاشارة المرجعية لتصغير مساحة الخطأ المدخل لجعل نظام السيطرة اكثر قوة في مواجهة التغيرات في عناصر النظام. عملت محاكاة لمنظومة السيطرة المصممة وكذلك فورنت من خلال المحاكاة مع متحكم العزم المحسوب. نتائج المحاكاة أثبتت صلاحية المنظومة المصممة واداءها العالي وكذلك أفضليتها على متحكم العزم المحسوب.

1. Introduction

Nowadays, in systems and cybernetics engineering, intelligence is the most important terminology. In robotics research, intelligence is the key issue for the next generation of robot [1].

Fuzzy and neural network control systems are two main strategies to deal with intelligence in control area. Fuzzy control theory can be defined as an intelligent method to transfer human expressions into numeral expressions for a dynamical system [2,3]. Human experiences and knowledge can be transferred to the dynamical system by converting linguistic expression into numerical values. The performance of control system can be degraded if fuzzy rules incorrectly designed by unexperted operator. However, unexperted operator of fuzzy system can use a standard or generic PD-like or PI-like fuzzy controller. Although, these controllers work well with some systems, in most cases fuzzy rules need to be modified based on system characteristics to get better results. Then, it is very true to say that designing perfect fuzzy rules is nearly impossible.

Even the optimal fuzzy rules have been designed, the performance of the system is not always guaranteed until the controller is robust enough to deal with those system parameter's variations and outer disturbances [1]. Therefore, designing fuzzy rules is not

easy at all [4]. This is one of disadvantages of fuzzy systems.

To solve this problem, adaptive techniques have been introduced into the framework of a fuzzy controller system structure. Designed fuzzy rules, and may be other parameters of fuzzy system, are adaptively adjusted according to changes in system parameters and disturbances [6]. The shape of membership functions is adjusted by modifying a center and a width of these functions to minimize the error of the system. The other method to improve fuzzy controlled system performance is to use neural networks or fuzzy-neural networks to compensate for uncertainties by minimizing the system error. One possibility of compensation is to add compensating signal to the control input level for fuzzy-controlled system; this will improve the performance of the system by achieving the inverse dynamics control which will lead to improving defuzzification process. The other possibility is to add the compensating signal to the reference trajectories to modify input error space so that we can achieve the same effect of modifying fuzzy rules [1].

In this paper, the last method has been used to design high specifications control system for robot manipulator. Fuzzy-neural network will be used as a compensator for uncertainties in a fuzzy-controlled robot manipulator.

New structure has been designed and computer simulation has been used to verify the validity and performance of this structure. Results curves and numerical values of simulation were compared with computed torque controller.

2. Robot manipulator dynamic equations

The dynamic equations of motion of rigid robot manipulator are:-

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (1)$$

In which q, \dot{q}, \ddot{q} are the $(n \times 1)$ vectors of position, velocity and acceleration of joints respectively. $M(q)$ is the $(n \times n)$ matrix of inertia of robot manipulator, $C(q, \dot{q})\dot{q}$ is the $(n \times 1)$ vector of coriolis and centrifugal forces. $g(q)$ is the $(n \times 1)$ vector of gravity forces. And τ is the $(n \times 1)$ vector of the input torque to be computed by the controller.

3. Control system

In the control structure proposed in this paper, fuzzy system is designed to work as a main controller in the structure, and FNN is used as a compensator for uncertainties of the manipulator. The outputs of FNN system or the compensating signal is added to the reference trajectories. By this method, input error space is modified so we can obtain the same effect of modifying fuzzy rules, and better tracking performance can be achieved.

The inputs to the FNN system are the modified error e_m and its derivatives. The

block diagram of the overall system is depicted in Fig.(1).

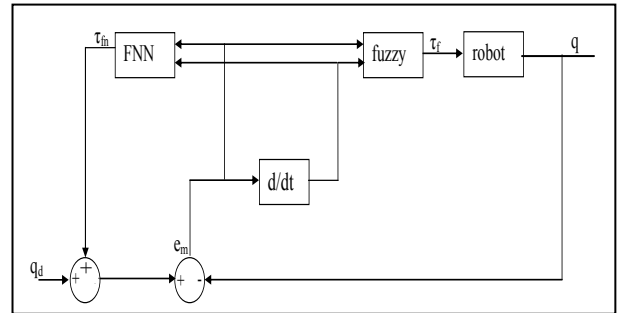


Fig.(1) Block diagram of proposed structure.

From Fig.(1)

$$e_m = q_d + \tau_{fn} - q \quad (2)$$

The fuzzy controller used in this structure is a PD-like Sugano. Fuzzy system output can be expressed by the following equation:-

$$\tau_f = k_1 e_m + k_2 \dot{e}_m \quad (3)$$

Where, k_1, k_2 are the dynamic gains of fuzzy controller.

Substituting eq.(2) into eq.(3) yields:-

$$\begin{aligned} \tau_f &= k_1 (q_d + \tau_{fn} - q) + k_2 (q_d + \tau_{fn} - q) \\ \tau_f &= k_1 e + k_2 \dot{e} + k_1 \tau_{fn} + k_2 \dot{\tau}_{fn} \end{aligned} \quad (4)$$

The function to be minimized is selected as

$$L = k_1 e + k_2 \dot{e} \quad (5)$$

Substituting (5) into (4) and rearranging yields:-

$$L = \tau_f - (k_1 \tau_{fn} + k_2 \dot{\tau}_{fn}) \quad (6)$$

3.1 Fuzzy controller design

PD-like fuzzy control system is the system adopted as a main controller. Fig.(2) shows the membership functions for the two inputs for the controller (the error e_m and its derivative \dot{e}_m) and Fig.(3) shows the

membership functions of the output.

Table(1) shows the rules of the system.

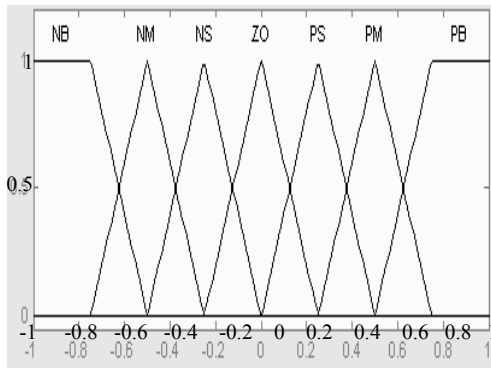


Fig.(2) scaled membership functions of fuzzy system inputs.

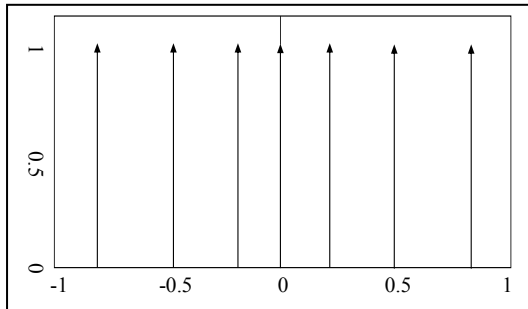


Fig.(3) scaled output membership functions

Table(1) rules of fuzzy system.

| \dot{e} \ e | NB | NM | NS | ZO | PB | PM | PS |
|---------------|----|----|----|----|----|----|----|
| NB | Nb | Nb | Nb | Nb | Nm | Ns | Zo |
| NM | Nb | Nb | Nb | Nm | Ns | Zo | Ps |
| NS | Nb | Nb | Nm | Ns | Zo | Ps | Pm |
| ZO | Nb | Nm | Ns | Zo | Ps | Pm | Pb |
| PB | Nm | Ns | Zo | Ps | Pm | Pb | Pb |
| PM | Ns | Zo | Ps | Pm | Pb | Pb | Pb |
| PS | Zo | Ps | Pm | Pb | Pb | Pb | Pb |

3.2 FNN design

The structure of FNN used is shown in

Fig.(4)

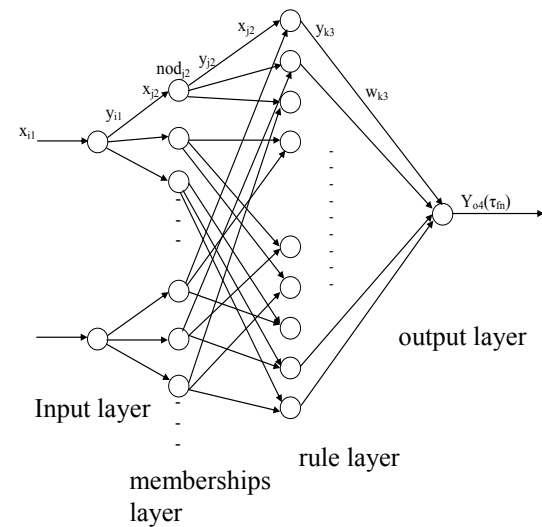


Fig.(4) FNN used as a compensator.

In this network:

- layer1 is the input layer in which

$$y_{i1} = x_{i1}$$

where $x_{11} = e_m$ and $x_{21} = \dot{e}_m$

- layer2 is the memberships layer, where each node perform a membership function. The function used is the Gaussian function. For general jth node

$$nod_{j2} = -\frac{(x_{i2} - m_{ij})^2}{(\sigma_{ij})^2}, \quad j=1,2,\dots,p$$

And

$$y_{j2} = \exp(nod_{j2})$$

Where, m_{ij} is the mean and σ_{ij} is the standard deviation of the Gaussian function. p is the total number of the linguistic with respect to the input node.

- Layer3 is the rule layer. The number of inputs for each node is two, and multiplication process is taken place in node to form the output of node (the rule). For general kth node

$$y_{k3} = nod_{k3} = \prod_j x_{j3}$$

- Layer4 is the output layer. Where single node performing summation action on weighted terms of its inputs.

$$y_{o4} = nod_{k4} = \sum_k w_{k4} x_{k4}$$

The FNN in the structure composed of n FNN networks, where n is the number of joints of robot manipulator. The inputs to FNN1 are e_{m1} and \dot{e}_{m1} where e_{m1} and \dot{e}_{m1} are the modified error and the derivative of modified error corresponding to the first link, e_{m2} and \dot{e}_{m2} are the input to FNN2, and so on. The output of each FNN is single line corresponding to the specified link.

The objective function for any FNN is selected to be:-

$$J = \frac{1}{2} L^2 \quad (7)$$

Now, we should find the update laws for the adjusted parameters of FNN^s w_{k4} , m_{ij} and σ_{ij} . The FNN^s are adjusted in online and backpropagation algorithm is used for that purpose.

In backpropagation, the parameters are adjusted using the gradient of J with respect to the parameter, i.e.:

$$\Phi(k+1) = \Phi(k) - \mu \frac{\partial J}{\partial \Phi(k)}$$

Where, Φ is the adjusted parameter and μ is constant representing the rate of learning. $\Phi(k+1)$ represents the current value of Φ and $\Phi(k)$ is the previous value of Φ . The partial derivative $\frac{\partial J}{\partial \Phi(k)}$ is the gradient of J with respect to Φ .

The gradient of J with respect to w_{k4} is:-

$$\frac{\partial J}{\partial w_{k4}} = \frac{\partial J}{\partial L} \frac{\partial L}{\partial \tau_{fn}} \frac{\partial \tau_{fn}}{\partial w_{k4}} \quad (8)$$

And this can be proved to be:-

$$\frac{\partial J}{\partial w_{k4}} = -L(k_1 y_{k3} + k_2 \dot{y}_{k3}) \quad (9)$$

Then, the update law for the weights w_{k4} can be written as:

$$w_{k4}(r+1) = w_{k4}(r) + \mu L(k_1 y_{k3}(r) + k_2 \dot{y}_{k3}(r)) \quad (10)$$

To find the update laws of the membership function parameters (m_{ij} and σ_{ij}), we should first find the gradient of J with respect to these parameters and as follow:-

$$\frac{\partial J}{\partial m_{ij}} = \frac{\partial J}{\partial L} \frac{\partial L}{\partial \tau_{fn}} \frac{\partial \tau_{fn}}{\partial y_{k3}} \frac{\partial y_{j2}}{\partial nod_{j2}} \frac{\partial nod_{j2}}{\partial m_{ij}}$$

And, this can be proved to be:-

$$\frac{\partial J}{\partial m_{ij}} = -L \left[\sum_k (k_1 y_{k3} + k_2 \dot{y}_{k3}) w_{k4} \right] \frac{2(x_{i2} - m_{ij})}{(\sigma_{ij})^2} \quad (11)$$

Using similar procedure the gradient of J with respect to σ_{ij} can be found to be:

$$\frac{\partial J}{\partial \sigma_{ij}} = -L \left[\sum_k (k_1 y_{k3} + k_2 \dot{y}_{k3}) w_{k4} \right] \frac{2(x_{i2} - m_{ij})^2}{(\sigma_{ij})^3} \quad (12)$$

Then, the update laws for m_{ij} and σ_{ij} can be written as

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \mu \frac{\partial J}{\partial \sigma_{ij}}(k) \quad (13)$$

$$m_{ij}(k+1) = m_{ij}(k) - \mu \frac{\partial J}{\partial m_{ij}}(k) \quad (14)$$

And $\frac{\partial J}{\partial m_{ij}}, \frac{\partial J}{\partial \sigma_{ij}}$ are as stated by equations (11) and (12) respectively.

4. Simulation study

A two link robot manipulator has been used as an example to study the proposed structure performance, and to compare it with other known structure.

The dynamics model equations for the robot manipulator used in the simulation are as stated by eq.(1), i.e:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau$$

In which,

$$M(q) = \begin{bmatrix} m_{11}(q) & m_{12}(q) \\ m_{12}(q) & m_{22}(q) \end{bmatrix}$$

$$C(q, \dot{q})\dot{q} = \begin{bmatrix} -B_{12}(q_2) - 2B_{12}(q_2)\dot{q}_1\dot{q}_2 \\ B_{12}(q_2)\dot{q}_2^2 \end{bmatrix}$$

$$g(q) = \begin{bmatrix} \gamma_1(q)g \\ \gamma_2(q)g \end{bmatrix}$$

Where,

$$m_{11}(q) = (m_1 + m_2)r_1^2 + m_2r_2^2 + J_2 + 2m_2r_2r_1 \cos(q_2)$$

$$m_{12}(q) = m_2r_2^2 + m_2r_2r_1 \cos(q_2)$$

$$m_{22}(q) = m_2r_2^2 + J_2$$

$$B_{12}(q) = m_2r_2r_1 \sin(q_2)$$

$$\gamma_1(q) = (m_1 + m_2)r_1 \cos(q_2) + m_2r_2 \cos(q_1 + q_2)$$

$$\gamma_2(q) = m_2r_2 \cos(q_1 + q_2)$$

Where, m_1, m_2 are the mass of the first and second links of the manipulator respectively, r_1 and r_2 are the lengths of these links, J_1 and J_2 are their inertias.

The nominal values of these parameters are selected to be:

$$m_1=0.5\text{kg}, m_2=1.5\text{Kg}, r_1=1\text{m}, r_2=0.8\text{m}, J_1=5\text{Kg.m and } J_2=5\text{kg.m.}$$

Simulation study included designing computed torque or inverse dynamic controller for the purpose of comparing our controller with this controller which is classified as a good controller for robot manipulators.

In computed torque controller, the control input is the inverse dynamics of the dynamic model of robot manipulator. i.e,

$$\tau = \hat{M}(q)u + \hat{C}(q, \dot{q})\dot{q} + \hat{g}(q)$$

$$\text{And, } u = \ddot{q}_d + k_v \dot{e} + k_p e$$

Where, $\hat{M}(q), \hat{C}(q, \dot{q})$ and $\hat{g}(q)$ are the estimated matrices of $M(q), C(q, \dot{q})$ and $g(q)$ respectively, q_d is the desired trajectories. If no uncertainties exist, then $\hat{M}(q) = M(q), \hat{C}(q, \dot{q})\dot{q} = C(q, \dot{q})\dot{q}$ and $\hat{g}(q) = g(q)$, and an ideal control is obtained.

4.1 Performance measures

Several performance measures have been selected to compare between the two

control structures, the proposed structure and computed torque controller. These performance measures are:

-Maximum steady state error (msse):-

The maximum error when the system reaches the steady states.

-Mean square error (mse):- The mean square error in a specified time range. This factor can be an indication of the amount of error. Mathematically,

$$mse = \frac{1}{T} \int_0^T |e_i|^2 dt$$

Where, T is the time of executing the task of robot manipulator. In our simulation 30s is assumed as a time for our task.

Settling time (ts):- The time required for the system to reach steady state.

-Mean square of applied torque (msat):-

This factor is an indication of the power consumed by joints drives. Mathematically,

$$msat = \frac{1}{T} \int_0^T |torque|^2 dt$$

It is assumed that the maximum allowed torque, which is determined by torque limitation of joint drives, is 300Nm. Then, the applied torque is not allowed to exceed this limit.

4.2 Simulation results

Matlab programming code has been used to simulate the systems. Runge kutta method is used to solve the differential equations of the dynamic model of the robot manipulator with 0.001s as an integration time step.

Several cases have been taken to include uncertainties, which included assumption that robot manipulator handle variable payloads. Also, disturbance of some form has been added, and it is assumed to take the following general form:

$$D = \begin{bmatrix} b q_1 \dot{q}_1 \sin(t) \\ b q_2 \dot{q}_2 \cos(t) \end{bmatrix} \quad (15)$$

Where, b is constant.

The desired trajectories are selected as:

$$q_d = \begin{bmatrix} 0.7 \sin(t) \\ 0.5 \cos(t) \end{bmatrix}$$

The following initial conditions are selected:

$$q = [0.2 \quad 0.2]^T$$

$$\dot{q} = [0 \quad 0]^T$$

Case1

In this case, we try to study the effect of robot manipulator's parameters uncertainties. The main source of parameter uncertainties is the payload variations which vary the link masses.

A deviation of link1 (m_1) and link2 (m_2) from the nominal values is assumed.

The link1 mass (m_1) is assumed to be 0.4Kg. A payload variations of four cases (0.5,1.5,3.5 and 5.5)Kg are assumed. The link2 mass (m_2) is effected directly by payload variations because the payload is attached directly to link2. Therefore, m_2 will be (2,3,5 and 7)Kg for payload variations respectively.

Figs.(5) and (6) show the tracking error of q_1 and q_2 respectively for computed torque controller, for the four values of m_2 .

Figs.(7) and (8) show the tracking error of q_1 and q_2 for the controller proposed in this paper for the four values of m_2 .

Table(2) contains the performance measures for $m_2=3\text{Kg}$. while, table(3) contains the performance measures for $m_2=7$.

Table(2) performance measures of the simulated systems for $m_2=3\text{Kg}$.

| | Msse(rad) ($\times 10^{-4}$) | | Mse(rad) ($\times 10^{-3}$) | | Msat(Nm) | | ts(s) |
|----------|--------------------------------|--------|-------------------------------|--------|----------|--------|-------|
| | Joint1 | Joint2 | Joint1 | Joint2 | Joint1 | Joint2 | |
| Cmp. trq | 80 | 48 | 0.6 | 0.2 | 2542 | 419.6 | 2 |
| Our str. | 1 | 0.4 | 0.3 | 0.1 | 2827 | 429.6 | 1.5 |

Table.(3) performance measures of the simulated systems for $m_2=7\text{Kg}$.

| | msse(rad) ($\times 10^{-4}$) | | mse(rad) ($\times 10^{-3}$) | | msat(Nm) | | ts(s) |
|----------|--------------------------------|--------|-------------------------------|--------|----------|--------|-------|
| | Joint1 | Joint2 | Joint1 | Joint2 | Joint1 | Joint2 | |
| Cmp. trq | 290 | 147 | 1.4 | 0.3 | 12277 | 2135 | 2.5 |
| Our Str. | 2.5 | 1.5 | 0.5 | 0.1 | 13406 | 2369 | 2 |

Case2

In this case, the effect of disturbances is studied. The disturbance described by eq.(15) is adopted.

The parameter b is varied to obtain different disturbance effect. Three different values of b have been taken; these values are 50,100 and 150. For each value of b , the systems have been simulated. The parameters m_1 and m_2 have been fixed at 0.4Kg and 3Kg respectively.

Figs.(9) and (10) show the tracking error of q_1 and q_2 respectively for computed torque controller.

Figs(11) and (12) show the tracking error of q_1 and q_2 respectively for our structure.

Table(5) contains the performance measures for $b=100$.

Table(4) performance measures of the simulated systems for $b=100$.

| | msse(rad) ($\times 10^{-4}$) | | mse(rad) ($\times 10^{-3}$) | | msat(Nm) | | ts(s) |
|----------|--------------------------------|--------|-------------------------------|--------|----------|--------|-------|
| | Joint1 | Joint2 | Joint1 | Joint2 | Joint1 | Joint2 | |
| Cmp. trq | 100 | 135 | 0.6 | 0.3 | 2476.4 | 611.3 | 2 |
| Our Str. | 2.2 | 4 | 0.3 | 0.1 | 2786.9 | 619 | 1.8 |

5. Conclusion

Simulation has showed the validity and also, superiority of the proposed structure on computed torque controller. Due to efficient online training, the tracking performance and robustness of the proposed structure are better than that of computed torque. Also, it has showed that the proposed structure has good features including high tracking performance and robustness and other features. The following points can be concluded from simulation study:-

-From figures and tables of case1 and case2, it can be said that the proposed structure shows very good tracking performance, and it has better tracking performance than computed torque controller. Also, it is better in other performance measure.

-From figures and tables of case1, it can be said that the proposed structure are very robust against parameters variations, where it keep very high performances with high parameter variations.

-From figures and table of case2, one can notice that the proposed structures have high robustness against disturbances.

References

- [1] D. H. Song, Y. Eom and S. Jong "Comparison studies of two neural networks compensation techniques for standard PD-like fuzzy controlled robotic manipulator", pp. 3178-3183, The 3dth annual conference of the IEEE industrial electronics.
- [2] T. H Lee and S. S. Ge, "Intelligent control of mechatronic systems", pp 646-659, IEEE Symposium on intelligent control, 2003.
- [3] M. E. Magana and F. Holzapfel, "Fuzzy-logic control of an inverted pendulum with vision feedback", pp. 165-170, IEEE Trans.on education, Vol. 41, No.2, may 1998.
- [4] D.Driankov, H. Hellendoom, and M. Reinfrank "An introduction to fuzzy control", Springer, 1996.
- [5] T. H. Hung, M. F. Yen, and H. C. Lu, "A PI-like fuzzy control implementation for the inverted pendulum system", pp. 218-222, IEEE Conference on intelligent processing systems, 1997.
- [6] L. X. wang, "Adaptive fuzzy systems and control", Prentice Hall, 1994.

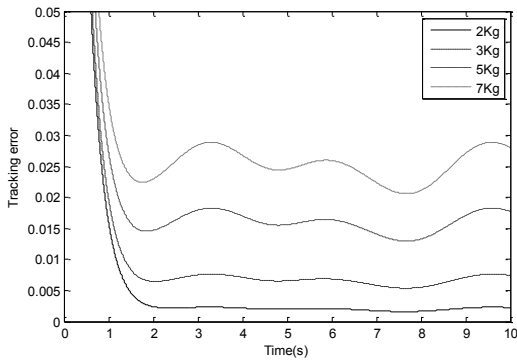


Fig.(5) Tracking error of joint1(q_1) of computed torque for case1.

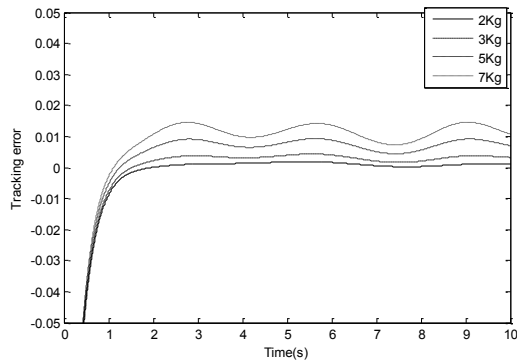


Fig.(6) Tracking error of q_2 of computed torque for case1.

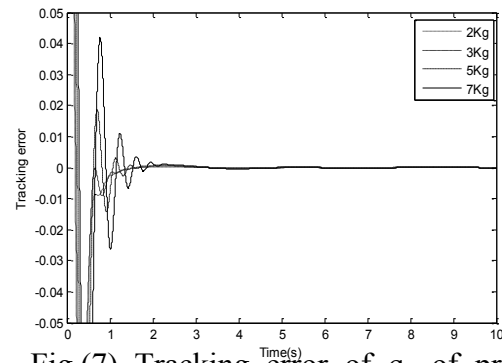


Fig.(7) Tracking error of q_1 of proposed controller for case1.

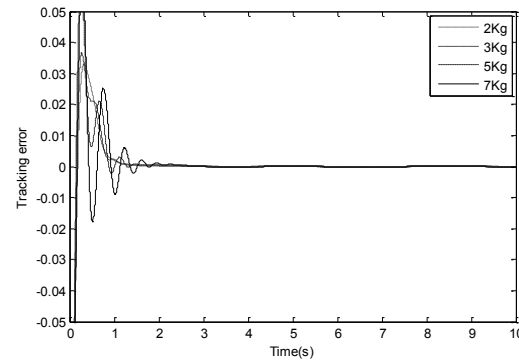


Fig.(8) Tracking error of q_2 of proposed controller for case1.

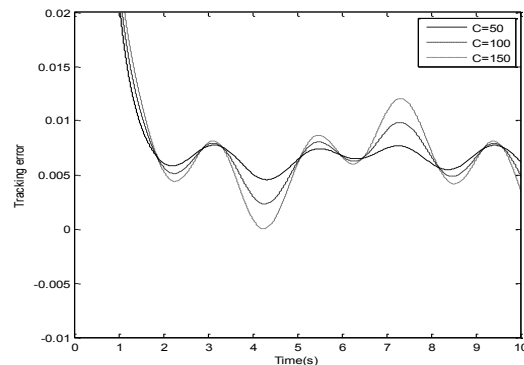


Fig.(9) Tracking error of q_1 of computed torque for case2.

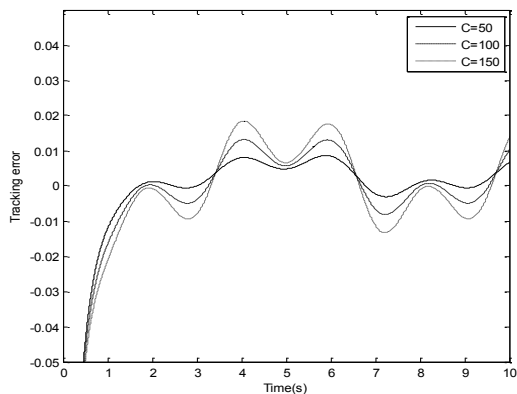


Fig.(10) Tracking error of q_2 of computed torque for case2.

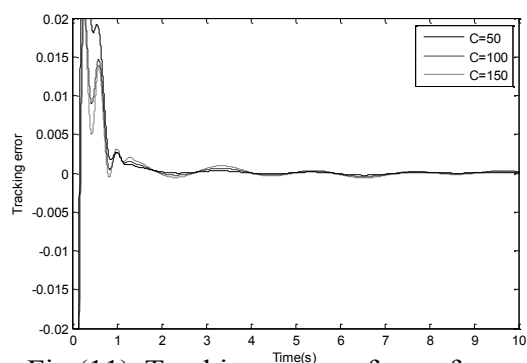


Fig.(11) Tracking error of q_1 of proposed controller for case2.

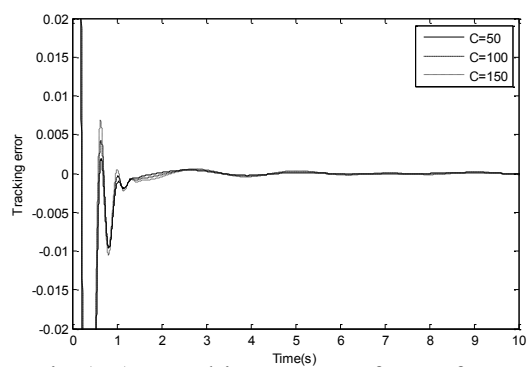


Fig.(12) Tracking error of q_2 of proposed controller for case2.