Designing a 2-DOF Painter Robot and Utilizing FAT-based Adaptive and Neural Network Controllers

1st Melika Salehian
Electrical Engineer departement
Amirkabir Universiy of Technology
Tehran, Iran
electeng@aut.ac.ir

2nd Mohammadhossein Hadian
Electrical Engineering departement
Amirkabir Universiy of Technology
Tehran, Iran
mh7mbb1130@aut.ac.ir

3rd Dr. Iman Sharifi
Electrical Engineer departement
Amirkabir Universiy of Technology
Tehran, Iran
imansharifi@aut.ac.ir

***Abstract*—As time goes by, the usage of robots by human being has been increased significantly, and due to circumstances, it is indeed inevitable. The field of painting can be mentioned as an area that robots have been doing a remarkable progress on it. Considering the accuracy required in this issue, not only the structure design of robot, but also the design of controllers is very crucial. In this article, after expressing the structural features of 2-DOF serial robot, an attempt is made to apply FAT-based adaptive and neural network controllers (that have their own advantages and have been used for different purposes before) on the painting robot. The simulation results are compared and the privilege of controllers are discussed. At the end, several suggestions are proposed in order to improve this project for future applications.**

***Keywords*—serial robot, painter, 2-DOF, neural network, FAT (function approximation technique)**

I. Introduction

The painting task usually consist of three fundamental elements which are: precision, accuracy and beauty. In order to meet these demands, an enormous amount of time and human recourses are needed. Moreover, there is always a limit which eradicate the possibility of using them at their full capacity. Furthermore, working with human workforce leads to faults and errors. To reduce the labor cost and to degrade the failure rate its minimum rate, automated robotic systems for painting process were proposed which meet highly accurate and sustained necessities of manufacturing. In a closer look, robots have significant impacts on different criteria which will be explored in details as this article proceeds:

1. *Reduce cost:*
   Although, at first the initial cost might seem hefty, but in the long run it will benefit you more than you think. Moreover, due to the increasing demands, manufacturers are trying to reduce the cost, so in result, price are decreasing every year.
2. *Improve quality:*
   It is a crystal-clear matter that if you want to stay or improve your position among competitors, you need to work on the quality of your production, especially when it comes to painting. Even the most talented painters cannot ensure matters that a robot can ensure. Matters like consistency and precision.
3. *Increase flexibility:*
   Robots can easily be reprogrammed to perform new jobs. Therefore, why bother yourself by creating a new fixture when a robot can make adjustments by moving a finger? This flexibility not only saves time during but also it reduces the cost. Automation also provides the chance to change materials and colors more frequently, this will allow you to simplify your production planning.
4. *Improve workforce efficiency:*
   Robots do not need to work based on a schedule, and also they do not need to take some day off. As a matter of a fact, they work three shifts of eight hours per day, which is equal to the work of three employees. Hence, one can conclude that labor and overheads, which usually are the largest drivers of savings can be reduced significantly, but that aside, workforce itself also benefits from robots in several tasks. They can perform arduous tasks, which reduces the possibility of on-the-job injuries.

Overall, painter robots are divided in two tiers. The first one is used in industrial applications [1,2] which involves spraying different kinds of parts like cars segments and also painting gigantic building's wall (which is a hazard job for human operator); And the second one is used in creating artistic paintings or schemes. Many studies have been done in order to control arms; but in countless number of these researches, the exact dynamic model of the robot is required for designing the controller torque [3]. Due to the depreciation of the links, there will be some changes in $\boldsymbol{M}$ and $\boldsymbol{C}$ and $\boldsymbol{G}$ matrixes; and also placing the robot in different positions and angles will lead to a divergent $\boldsymbol{G}$ matrix. So, the dynamic is constantly changing this will be troublesome. Therefor control laws that don't have any dependency in model of the robot are usually preferred. In this paper, the main focus is on the classic controllers and neural network controllers, which in classic controllers the adaptive part is specifically investigated thoroughly. Usually for adaptive controllers, first thing that comes to mind is Linear in parameter (LIP) controller. In order to write this controller, one should know and write the regressor matrix ($\boldsymbol{Y}$) very well, which a difficult task is considering that you need to separate parameter vector from the torque matrix if you want to obtain the regressor matrix. Therefore, robust control based on passivity is introduced [4]. Although, the previous issue will be resolved in this way it will generate a new problem. For this specific controller, knowledge about bandwidth of error is required, if not one should find it by trial-and-error task.

Therefore, normally it is not used, and adaptive passivity control is suggested to solve this matter [5]. However, one should bear in mind that it is also based on LIP algorithm, so again the problem of LIP will be occurred. Hence, function approximation technique (FAT) is proposed, which will solve previous issues [6]. Thus, the two main controllers that will be explained in this paper thoroughly will be **FAT** and **neural network controller** [7] (which both of them have the aforementioned advantage).

This paper is organized as follows: In section II, the working mechanism of the whole system and fundamental equations of the robot (forward kinematic, inverse kinematic and the dynamic model of the robot) are described briefly. In section III, there will be discussion on how to obtain principles of the controllers and proving their stabilities. In section IV, showing the results will be presented. Finally, in section V, the conclusion is drawn and suggestion for further developments will be made.

II. ROBOT MODEL

A. Mechanical setup

Two degree of freedom (2-DOF) robot is used in designing the system. The end of the second link will be considered as end-effector (EE) and the pen which is in charge of painting, will be put there. Both of the joints are responsible for finding the position which pen should follow.
For this robot, the links were designed in a way to be identical to each other in the best way possible, so that the utmost usage of work space can be utilized. However, the second link was considered a little bit shorter, in order to prevent weighting problems from happening. Therefore, the robot will not be pulled down by the gravity force. The first motor will be located on the top of the base. There will be one bearing drilled in the upper part in order to move the link and Motor simultaneously. The second motor, will be placed on the top of the second link. Motors which were used in this design are MG996R, and their range is ±180 degree.

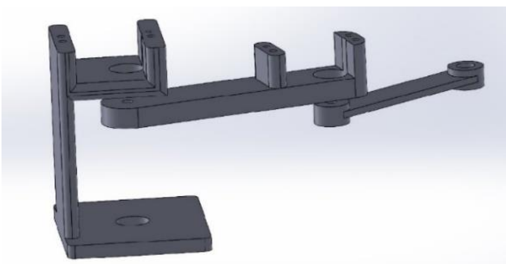

Fig.1-a. Solidwork structure design

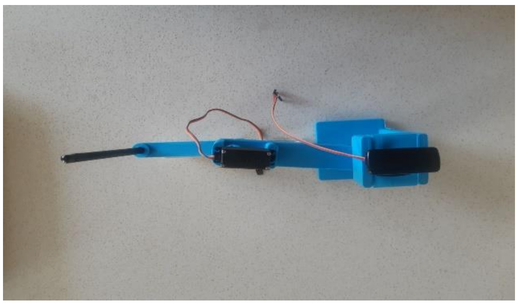

Fig.1-b. The implementation of the Solidwork structure design with 3D printer

B. Forward kinematic:

In order to study the motion of a machine, kinematic analysis will be needed. Parameters like the position, displacement, rotation, velocity and acceleration will be achieved by using forward kinematics. In order to figure out how the joint angles change the position of the end-effector, the Denavit-Hartenberg table should be derived [8]. The equations that determine the position of the EE are as below:

$$X_{EE} = l_1 cos\theta_1 + l_2 cos(\theta_1 + \theta_2) \quad (1)$$
$$Y_{EE} = l_1 sin\theta_1 + l_2 sin(\theta_1 + \theta_2) \quad (2)$$

Where l_1 *and* l_2 are the lengths of the links and θ_1 *and* θ_2 are the joint angles which are presented in the following picture:

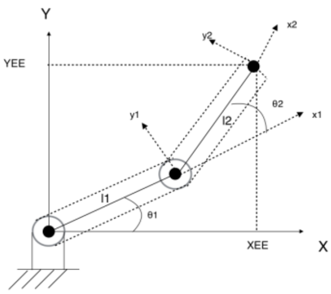

Fig. 2. Kinematic Diagram of a Two-Link Planar Manipulator

	a_i	α_i	d_i	θ_i
i = 1	l_1	0	0	θ_1
i = 2	l_2	0	0	θ_2

Table. 1. Classic Denavit-Hartenberg

C. Inverse kinematic:

The problem of controlling the end-effector in the task space, depends on the joint angles. So inverse kinematic should be derived. This will convert the task space into joint space:

$$X_{EE}^2 + Y_{EE}^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos(\pi - \theta_2)$$

$$\cos(\theta_2) = \frac{1}{2l_1 l_2}(X_{EE}^2 + Y_{EE}^2 - l_1^2 - l_2^2) = B$$

$$\sin(\theta_2) = \pm\sqrt{1 - B^2} = A$$

$$\theta_2 = atan2(A, B) \quad (3)$$

From the equations (1) and (2):

$$cos(\theta_1 + \theta_2) = \frac{1}{l_2}(X_{EE} - l_1 \cos(\theta_1)) \quad (4)$$

$$sin(\theta_1 + \theta_2) = \frac{1}{l_2}(Y_{EE} - l_1 \sin(\theta_1)) \quad (5)$$

From (4) and (5):

$$\frac{1}{l_2^2}(X_{EE} - l_1\cos(\theta_1))^2 + \frac{1}{l_2^2}(Y_{EE} - l_1\sin(\theta_1))^2 = 1$$

$$(X_{EE}^2 + Y_{EE}^2) + l_1^2 - 2l_1(X_{EE}\cos(\theta_1) + Y_{EE}\sin(\theta_1)) = l_2^2$$

Thus:

$$X_{EE}\cos(\theta_1) + Y_{EE}\sin(\theta_1) = \frac{(l_1^2-l_2^2)+(X_{EE}^2+Y_{EE}^2)}{2l_1} = k$$

Taking $\sin(\theta_1) = \alpha$:

$$X_{EE}\sqrt{1-\alpha^2} = k - Y_{EE}\alpha$$

$$(X_{EE}^2 + Y_{EE}^2)\alpha^2 - 2kY_{EE}\alpha + (k^2 - X_{EE}^2) = 0$$

Solving the second-degree equation above will lead to:

$$\alpha = \left[kY_{EE} \pm X_{EE}\sqrt{X_{EE}^2 + Y_{EE}^2 - k^2}\right]/(X_{EE}^2 + Y_{EE}^2)$$

And at the end:

$$\cos(\theta_1) = \pm\sqrt{1-\alpha^2} = \beta$$

$$\theta_1 = \operatorname{atan2}(\alpha, \beta)$$

D. Dynamic:

The dynamic equation of the robot is [9]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = u \tag{6}$$

where $\boldsymbol{M(q)}$ is the n×n inertia matrix, $\boldsymbol{C(q,\dot{q})\dot{q}}$ is a n×1 vector that represent centrifugal and Coriolis forces and $G(q)$ is the n×1 gravity vector. Be cautious that friction's impact was not considered in the modeling. In this article, n is equal to 2 according to the degree the robot's degree of freedom. The exact values of the matrixes are as below:

$$M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$$

$$m_{11} = m_1 l_{c1}^2 + m_1(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}^2 + 2l_1 l_2 cos(q_2)) + I_1 + I_2$$

$$m_{12} = m_{21} = m_2(l_{c2}^2 + l_1 l_2 cos(q_2)) + I_2$$

$$m_{22} = m_2 l_{c2}^2 + I_2$$

where $m_1\ and\ m_2$ are the mass of links, $l_1\ and\ l_2$ are the length of links, $l_{c1}\ and\ l_{c2}$ are distances between the joints and their corresponding links center of mass, and $I_1\ and\ I_2$ are the inertia of the links.

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

$$c_{11} = h\dot{q}_2$$

$$c_{12} = h\dot{q}_1 + h\dot{q}_2$$

$$c_{21} = -h\dot{q}_1$$

$$c_{22} = 0$$

Where:
$h = -m_2 l_1 l_{c2}\sin(q_2)$

$$G = \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix}$$

Since the robot is planar in x-y plane, so the gravitational potential energy relative to the z axis can be considered zero for each of the link's center of masses:

$$P = p_1 + p_2 = 0$$

Then:

$$\varphi_1 = \frac{\partial P}{\partial q_1}$$

$$\varphi_2 = \frac{\partial P}{\partial q_2}$$

So, the $\boldsymbol{G}$ matrix will be:

$$G = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

III. Controllers

A. FAT based adaptive controller

The dynamics of a robot usually is modeled by some nonlinear differential equations, and of course, uncertainties are unavoidable. However, the dynamic model for a robot in real life cannot be achieved as easy as it is in the simulations. Therefore, several robust and adaptive control strategies are suggested. However, function approximation technique (FAT) is a controller that instead of utilizing the exact dynamic of the robot, its estimation will be considered; which will be achieved by putting Fourier series into use. The main idea is to demonstrate system uncertainties by applying combinations of orthonormal vectors with some weighting matrixes.

The tracking error is defined as below:

$$e = q - q_d$$

And the filter error is:

$$r = \dot{e} + \Lambda e$$

As previously mentioned, $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{G}$ are not known. Therefor the considered controller is:

$$u = \widehat{M}\ddot{q}_d - \widehat{M}\Lambda\dot{e} + \hat{C}\dot{q}_d - \hat{C}\Lambda e + \hat{G} - k_d(\dot{e} + \Lambda e)$$

Where k_d is a 2×2 positive definite matrix and all items with hats, are considered to be the estimation of exact quantities. The representation of M, C and G will be employed as follow:

$$\widehat{M} = \widehat{W}_M^T Z_D$$

$$\hat{C} = \widehat{W}_C^T Z_C$$

$\hat{G} = \widehat{W}_G^T Z_G$

Where Z_M, Z_C and Z_G are considered as basis vectors, which are composed of Fourier series estimation. Hence, after that the weighting matrixes which are $\boldsymbol{W_M}$, $\boldsymbol{W_C}$ and $\boldsymbol{W_G}$, will be formed based on basis vectors. It is also a known fact that the accuracy of estimation, depends on the degree you use Fourier series, the higher you go, the closer your stimulation will get. Consequently, the error will converge to zero which is desirable.
Under these conditions and with respect to the (6), the closed loop equation will be:

$$M\dot{r} + Cr + K_d r = \widetilde{W}_M^T Z_M(\ddot{q}_d - \Lambda\dot{e}) + \widetilde{W}_C^T Z_C(\dot{q}_d - \Lambda e) + \widetilde{W}_G^T Z_G$$

Where $\widetilde{W} = \widehat{W} - W$ for all of the $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{G}$ matrixes. Updating laws for the weighting matrixes will be selected, so that the time derivative of the Lyapunov function will be proved to be negative.
In order to accept a controller, the stability of it should be proven. So, the Lyapunov function considered for this controller is:

$$V\big(r, \widetilde{W}_M, \widetilde{W}_C, \widetilde{W}_G\big) = \tfrac{1}{2} r^T M r + \tfrac{1}{2} Trace(\widetilde{W}_M^T Q_M \widetilde{W}_M + \widetilde{W}_C^T Q_C \widetilde{W}_C + \widetilde{W}_G^T Q_G \widetilde{W}_G)$$

Where $\boldsymbol{Q_M}, \boldsymbol{Q_C}$ and Q_G are positive definite matrixes and their dimension was considered 6×6 identity matrix for $\boldsymbol{Q_M}$ *and* $\boldsymbol{Q_C}$ and 3×3 identity matrix for $\boldsymbol{Q_G}$. Moreover, for proving the stability of the controller the derivative of suggested Lyapunov should be taken:

$$\dot{V} = r^T\big[-Cr - K_d r + \widetilde{W}_M^T Z_M\big(\ddot{q}_d - \Lambda\dot{e}\big) + \widetilde{W}_C^T Z_C\big(\dot{q}_d - \Lambda e\big) + \widetilde{W}_G^T Z_G\big] + \frac{1}{2} r^T \dot{M} r + Trace(\widetilde{W}_M^T Q_M \dot{\hat{W}}_M + \widetilde{W}_C^T Q_C \dot{\hat{W}}_C + \widetilde{W}_G^T Q_G \dot{\hat{W}}_G)$$

By utilizing the fact that $\dot{M} - 2C$ is skew symmetric, and also choosing the update laws to be:

$$\dot{\hat{W}}_M = -Q_M^{-1} Z_M r(\ddot{q}_d - \Lambda\dot{e})^T$$

$$\dot{\hat{W}}_C = -Q_C^{-1} Z_C r(\dot{q}_d - \Lambda e)^T$$

$$\dot{\hat{W}}_G = -Q_G^{-1} Z_G r^T$$

Then the derivation of the Lyapunov function will be:

$$\dot{V} = -S^{\dot{T}} K_d s \leq 0$$

In conclusion, by applying this controller, the stability of the robot is guaranteed. And also, it is possible to control the robot without the need to have knowledge about the exact dynamic of the robot.

B. Neural Network Controller:

In this section, neural network (NN) was used to estimate torque input. Hence, a multi-layer perceptron (MLP) with three layers was considered. A sigmoid activation function was used for hidden layer:

$$\sigma(z) = \frac{1}{1 + e^{\alpha z}}$$

And as for output layer, linear activation function was chosen. Moreover, the output function can be written as follow:

$$\hat{f}(x) = \widehat{W}^T \sigma\big(\hat{V}^T X\big) \tag{7}$$

Where $\boldsymbol{\hat{V}}$ is the estimation of the weight matrix for the hidden layer of neural network, $\boldsymbol{\widehat{W}}$ is considered the estimation of output layer's weight matrix, $\hat{f}(x)$ is the output of the network, which is the approximation of applied torque; and *X* is the input of the network:

$$X = [e^T, \dot{e}^T, q_d^T, \dot{q}_d^T, \ddot{q}_d^T, 1]^T$$

So, *X* is a 11×1 vector, and the last element of it is set to be 1 in order to avoid putting error term in the neural network output's equation.
The errors that can be considered for getting feedback from the output of the system are:

$$e(t) = q_d(t) - q(t)$$

$$r = \dot{e} + \Lambda e$$

Where e(t) is the tracking error that is composed of $q_d(t)$ which is the desired trajectory input, and r(t) is the filtered tracking error.
Also, the control torque input was chosen as:

$$\tau = \hat{f} + K_v r \tag{8}$$

The control output, which will be also considered as torque input, will be achieved by combining (7) and (8):

$$\tau = \widehat{W}^T \sigma\big(\hat{V}^T x\big) + K_v r - v \tag{9}$$

Considering the dynamic equation of a robot (6) and (9) will lead to the closed loop equation below:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \widehat{W}^T \sigma\big(\hat{V}^T x\big) + K_v r - v \tag{10}$$

v was assumed zero. The updating rules will be:

$$\dot{\widehat{W}} = F\sigma r$$
$$\dot{\hat{V}} = Gx(diag(\hat{\sigma}'(W^T r)^T)$$

Where *F* and *G* are positive definite matrixes. The Lyapunov function that was chosen to prove stability of controller is:

$$L = \frac{1}{2} r^T M r + \frac{1}{2} trace(\widetilde{W}^T F^{-1} \widetilde{W}) + \frac{1}{2} trace(\widetilde{V}^T G^{-1} \widetilde{V})$$

$$\dot{L} = r^T M \dot{r} + \frac{1}{2} r^T \dot{M} r + trace\left(\widetilde{W}^T F^{-1} \dot{\widetilde{W}}\right) + \frac{1}{2} trace(\widetilde{V}^T G^{-1} \dot{\widetilde{V}})$$

Emerging the closed loop equation (10) with the consideration to v equal to zero, and also the weight updating law will lead to:

$$\dot{L} = -r^T K_v r \leq 0$$

Performance of this controller is based on 3 assumptions:
1) Neural network estimation is working without any error; 2) There will not be any unmodeled disturbance; 3) The stability of weights updating law was not investigated. Therefore, in order to handle these limitations, another neural network controller was proposed which v term is not zero in the control law and the weights updating law will change into:

$$\dot{\widehat{W}} = F\sigma r - F diag(\sigma') V^T x r^T - \kappa F ||r|| W$$

$$\dot{\hat{V}} = Gx(diag(\hat{\sigma}') W^T r)^T) - \kappa G ||r|| V$$

Where κ was considered as a constant with the magnitude of 2. Furthermore, $\boldsymbol{G}$ and $\boldsymbol{F}$ were considered as 11×11 and 10×10 matrix. A robustifying term was added:

$$v(t) = -K_z(||\hat{Z}||_F + Z_M) r$$

Where $\boldsymbol{Z_M}$ is the summation of the norm2 of $\boldsymbol{W}$ and $\boldsymbol{V}$ matrixes when they converge to their final values and $\hat{Z}$ is the concatenation of $\boldsymbol{W}$ and $\boldsymbol{V}$ matrixes. K_z is also another constant that can be changed to get the best tracking performance. The control law of this controller will turn into:

$$\tau = \widehat{W}^T \sigma(\hat{V}^T x) + K_v r \text{-v}$$

The Lyapunov function will remain the same; and the derivation of Lyapunov function will be as below:

$$\begin{aligned}\dot{L} = -r^T K_V r + \frac{1}{2} r^T (\dot{M} - 2V_m) r \\ + trace \widetilde{W}^T \left(F^{-1} \dot{\widetilde{W}} + \hat{\sigma} r^T - (\hat{\sigma}' \hat{V}^T x r^T\right) \\ + trace \tilde{V}^T \left(G^{-1} \dot{\tilde{V}} + (x r^T \hat{V}^T \hat{\sigma}'\right) \\ + r^T (w + v)\end{aligned}$$

By simplifying several equations:

$$\begin{aligned}\dot{L} \leq -||r|| \left[K_{vmin} ||r||\right. \\ \left. + \kappa \left| ||\tilde{Z}||_F (||\tilde{Z}||_F - Z_M) - C_0 - C_1 \right| ||\tilde{Z}||_F \right]\end{aligned}$$

Assume that:

$$K_{vmin} ||r|| + \kappa \left| ||\tilde{Z}||_F (||\tilde{Z}||_F - Z_M) - C_0 - C_1 \right| ||\tilde{Z}||_F] \geq 0$$

Then the stability will be proved:

$$\dot{L} \leq 0$$

IV. Figures and Results

In this section the result of simulations will be brought up and will be explained. In order to obtain the best control gains, several attempts were required, however, the results shown here are the best ones that could be obtained. Note that the results are for various input frequencies so that the functionality of controllers can be compared in different situations and with respect to each other.

A. *FAT based controller rsesult:*

The desired input and the output will be shown as below:

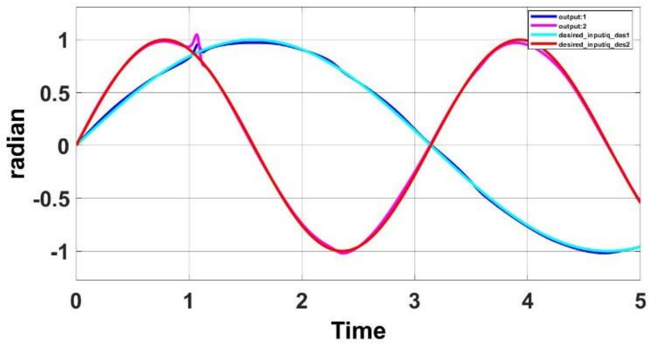

Fig. 3. Desired input and output with 1, 2 rad/s frequencies

The goal is to make the output track the desired input in the best way that is possible, and this work can be done by changing gamma, f_n, $\boldsymbol{Q_G}$, $\boldsymbol{Q_M}$, and $\boldsymbol{Q_C}$. However, you need to be cautious that there is trade of between the best track, and the torque that can be provided by motors. Therefore, considering the fact that MG996R is being used for motor, so the maximum torque available will be ±1. In result, the controller gain was set to be

$$k_d = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

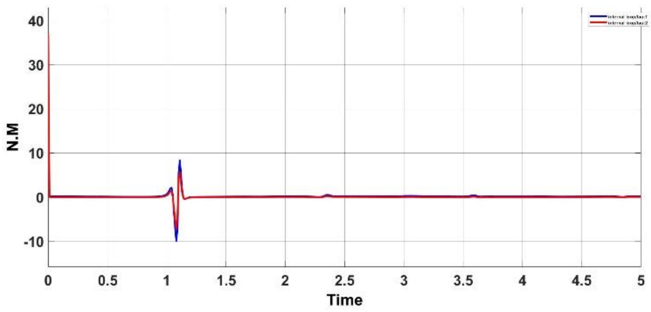

Fig. 4. Controller output (torque input)

In order to fit the controller to a wide range, 0.1 and 8 frequencies were also checked and coefficients were changed accordingly:

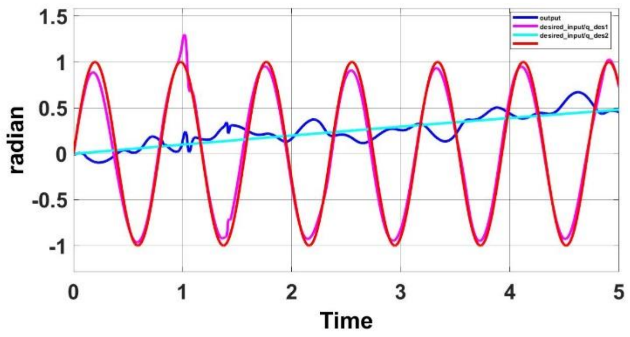

Fig. 5. Desired input and output with 0.1, 8 rad/s frequencies

While it is still an impeccable track there are some overshoots that are visible for both of the outputs

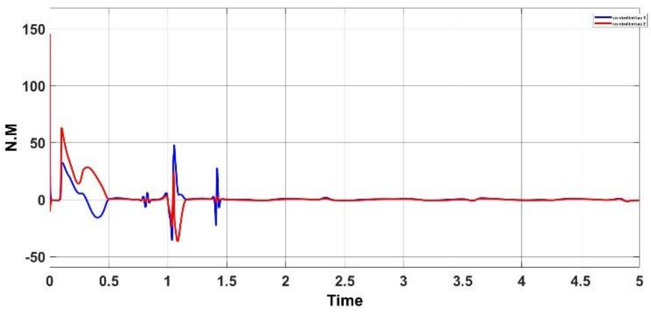

Fig.6. Controller output (torque input)

For optimizing the performance controller's gain had to be changed, so it altered to

$$k_d = \begin{bmatrix} 8 & 0 \\ 0 & 8 \end{bmatrix}$$

Still the torque has some upside and downside hills which can result in failure while working with MG99R motors, because this motor is not capable of delivering this kind of torque.

B. *Neural Network based controller rsesult*

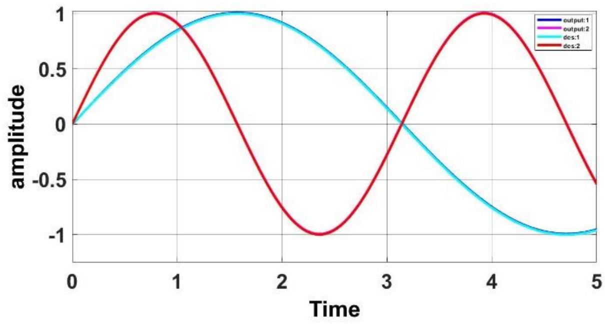

Fig. 7. Desired input and output with 1, 2 rad/s frequencies

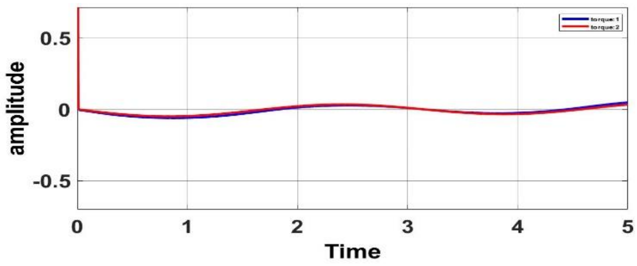

Fig. 8. Controller output (torque input)

The torque input range is between -1 and +1 which is a good feature due to this fact that lots of motors cannot deliver torque outside of this range

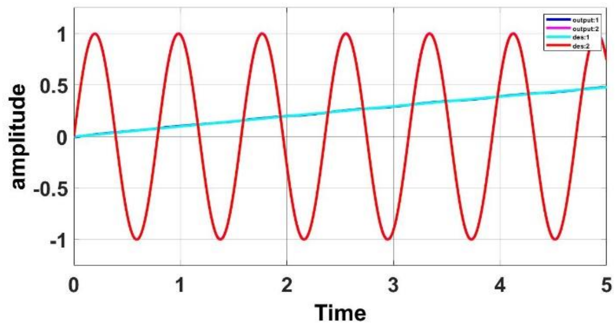

Fig. 9. Desired input and output with 0.1, 8 rad/s frequencies

Based on figure 9 and figure 7 it can be interpreted that even by altering the input frequencies the tracking will be remarkable and stays the same.

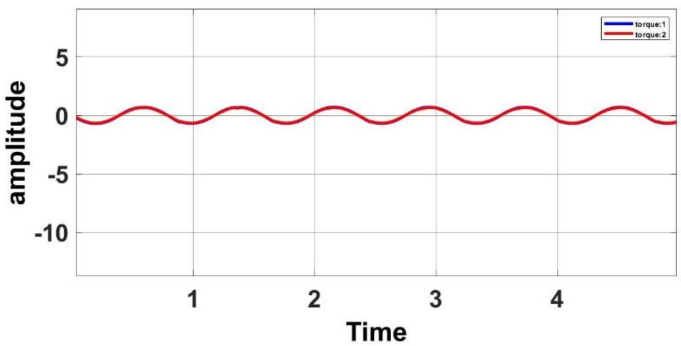

Fig.10. Controller output (torque input)

Moreover, the scale will not change even if the input frequencies were changed which is why it is a good choice for real world implementation.

For these results to be achieved gains were considered as follow: Kv=20, Kz=1.

As it was mentioned before, there is a relationship between the best performance and the ability of robot to deliver what is required for this to happen. The changes will be made to the variables.

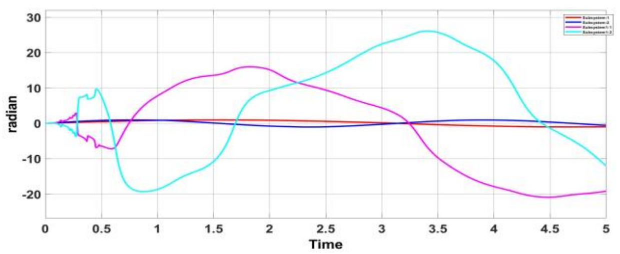

Fig.11. Comparison between controllers output (FAT based controllers)

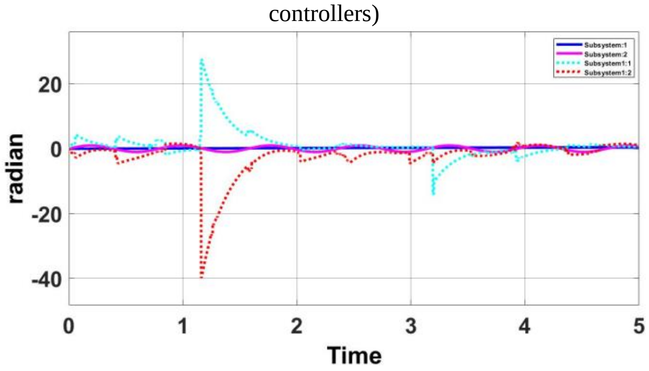

Fig.12. Comparison between Controllers output (neural network based controllers)

The controller with the same coefficients as the paper and also the tuned controller were tried on the system simultaneously, and the results are shown in figure 11 and figure 12.Clearly the performance has improved a lot.

V. Conclusion

In this work, a structure of robot was used for another application. Moreover, two different controller was tested to get the best result. One of them, the classic controller, is FAT-based adaptive controller, and the other one is a neural network controller. In spite of other classic controllers, FAT-based adaptive controller, not only does not need any calculation for regressor matrix, but also it does not require acceleration feedback. This is one of the superiorities of this controller due to the cost of the acceleration sensors. Moreover, the stability of closed loop was investigated thoroughly with the suggested Lyapunov energy like function. Besides, the neural network controller was also implemented, and the stability was proven. It is a known fact that neural network controllers' costs are really high, hence a good reason is required to implement them rather than a classic one. In this paper, it was shown that with changing multiple frequencies, the other controllers lose their ability to track well, so neural network controller was utilized as a solution to this problem. As it was illustrated in figure 7 and 9, the position tracking was remarkable. This also encourages authors to focus their researches in this area for future developments. For example, the proposed system can be enhanced by adding an image processing library such as OpenCV, in order to do both painting and capturing pictures at the same time. Moreover, there are some extra ways that can improve the trajectory planning criteria. For instance, this robot can be used in course of actions which leads to avoiding obstacle simultaneously with the use of neural network controller, but the price is that the calculation require for this controller is expensive.

REFERENCES

[1] M.Vincze, A.Pichler, G.Biegelbauer, K. Häusler, H. Andersen, O. Madsen, M. Kristiansen "Automatic Robotic Spray Painting of Low Volume High Variant Parts" 33rd ISR (International Symposium on Robotics) October 7 – 11, 2002

[2] "Paint robots in the automotive industry–process and cost optimization" ABB Review 4/1996

[3] K. Jankowski "Inverse Dynamics control in Robotics Applications" November 2004

[4] C.Uyulan "Design and stability analysis of a robust- adaptive sliding mode control applied on a robot arm with flexible link" September 2021

[5] M.Mahnoodi, M.Kojourimanesh, M.eghtesad, M.Farid "Adaptive Passivity-Based control of a flexible-joint manipulator subject of collision"Jun 2014

[6] A.C.Huang, S.C.Wu, W.F.Ting "A FAT-based adaptive controller for robot manipulator without regressor matrix: theory and experiments" March 2006, pp205-210

[7] K.Qian "Neural Network controller for Manipulator". MMAE540 Robotics- Class Project Paper. Department of Biomedical Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA

[8] M.W. MSpong, S.Hutchinson,M.Vidyasagar "Robot Modeling And Control' 2006 John Wiley & Sons, Inc.

[9] Introduction to Robotics: Mechanical and Control. Book by John J Craig.

[10] I. kesici, S. biskin, A. eleyan "automatic object painting with SCARA robot using computer vision" Volume 14, Issue 1, p17-22

[11] B. Siciliano, O. Khatib"Springer Handbook of Robotics" Springer-Verlag Berlin Heidelberg 2008

[12] Nopphawan Nurnuansuwan and Aphilak Lonklang "A Prototyping of 2-DOF Robot Arm Using Feedback Control System" 2019 5th International Conference on Control, Automation and Robotics (ICCAR)