A GA-Integrated Adaptive Model Reference Controller in Robot Tracking Applications

M. Chakravarti
Indian Institute of Information Technology Design and Manufacturing
Jabalpur, India

A. L. Sivakumar
Indian Institute of Information Technology Design and Manufacturing
Jabalpur, India
sivakumar.iiitdm@gmail.com

***Abstract*—An integrated solution based on adaptive controller ideas is proposed for robotic trajectory tracking problem. The proposal includes a GA-aided adaptive model reference Lyapunov-based nonlinear algorithm for position regulation problem and path planning, respectively. The proposed method designs a simply implementable controller. In addition, the inverse dynamic in tracking problem, which needs an appropriate reference signal, is investigated using genetic algorithm. The proposed approach is evaluated in simulation on the easily accessible six DOF robot model PUMA-560, and the key features of the introduced method are illustrated.**

Keywords—component; Robot, PUMA 560, adaptive model reference controller, path planning, genetic algorithm

I. INTRODUCTION

The major purpose of robot-controlled systems is the tracking of a reference path, which involves the production of an appropriate control signal to make the error between the robot current position and the reference position zero [1-6]. Considering this principle, this paper presents an integrated solution for robot trajectory tracking based on a GA-Adaptive model reference controller, which is implemented on PUMA 560 arm robot modeled in [7-14]. The proposed approach can be used for many other type of industrial robots such as spray painting robot [15], arc welding [16], assembly [17], polishing [18], etc. Adaptive model reference controller that is implemented in previous works such as [19] and [20] is used in the controller design procedure. Furthermore, the GA-aided controllers are also investigated before [21, 22]. Also the problem of path planning using GA algorithms are studied frequently [23, 24], however to the best knowledge of the authors the tracking problem of the serial arm robots is not solved using GA-Adaptive model reference controller which gives a simple implementable controller. The outline of this work is as follows. In the next section we will introduce the dynamic modeling of the PUMA 560 ARM, while Section 3 presents the adaptive model reference controller design algorithm for zero tracking error. Section 4 will introduce the general GA procedure and its application in this paper. Finally, the simulation results will be illustrated.

II. ROBOT MODELING

Fig. 1 shows the schematic structure of PUMA 560 robot arm with up to six degree of motion. This robot has six Electric DC Servos attached to each of the arms from base to the end effector. It is designed to wide range of small parts-handling application such as packaging functions, pharmaceutical and personal care and food industries.

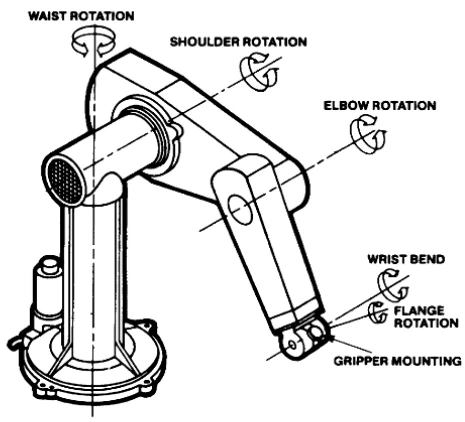

Fig. 1. *schematic structure of PUMA 560 robot arm*

The dynamic model of puma 560 serial robot in matrix form obtained as [1]

$$A(q)\ddot{q}+B(q)[\dot{q}\dot{q}]+C(q)[\dot{q}^2]+g[q]=\Gamma \qquad (1)$$

where $A(q)$ is the $n\times n$ kinetic energy matrix, $B(q)$ is the $\frac{n\times(n-1)}{2}$ matrix of coriolis torques, $C(q)$ is the $n\times n$ matrix of centrifugal torques, $g(q)$ is the n-vector of gravity torques, Γ is the generalized joint force vector And the symbols $[\dot{q}\dot{q}]$ and $[\dot{q}^2]$ are notations for the $\frac{n\times(n-1)}{2}$-vector of velocity products and the n-vector of squared velocities which is given as

$$[\dot{q}\dot{q}] = [\dot{q}_1\dot{q}_2, \dot{q}_1\dot{q}_3, \cdots, \dot{q}_1\dot{q}_n, \cdots, \dot{q}_{n-1}\dot{q}_n]^T, \quad [\dot{q}^2] = [\dot{q}_1^2, \dot{q}_2^2, \cdots, \dot{q}_n^2]^T \tag{2}$$

The procedure used to derive the dynamic model entails four stages:

1. Generation of the kinetic energy matrix and gravity vector. This can be performed by the summations of either Gibbs-Alembert or the Lagrange formulation.

2. Deriving of the kinetic energy matrix elements by combining inertia constants that multiplying variable expressions.

3. Simplification of the centrifugal and coriolis matrix elements in terms of partial derivations of kinetic energy matrix and reduction of these expressions with the relations that hold on these partial derivatives.

4. Formation of the needed partial derivatives, expansion of the centrifugal and coriolis matrix elements in terms of the partial derivatives, and modifying by having inertia constants as in step 2.

Some simplifying assumptions for this analysis consist of the rigid links assumption, link 6 has been assumed to be symmetric, that is $I_{xx} = I_{yy}$, and only I_{xx}, I_{yy} and I_{zz} for all links are considered, that is $I_{xy} = I_{xz} = I_{yz} = 0$.

In the third step the Christoffel symbols of the first kind is used to write the elements of the Coriolis matrix, b_{ij}, and of the centrifugal matrix, c_{ij},

$$b_{ij} = 2\beta^{i,kl} \tag{3}$$

$$c_{ij} = \beta^{i,jj} \tag{4}$$

$$\beta^{i,jk} = \frac{1}{2}\left(\frac{\partial a_{ij}}{\partial q_k} + \frac{\partial a_{ik}}{\partial q_j} - \frac{\partial a_{jk}}{\partial q_i}\right) \tag{5}$$

where $(\dot{q}_k \times \dot{q}_l)$ is the j th velocity product in the $[\dot{q}\dot{q}]$ vector, and where $\beta^{i,jk}$ is the Christoffel symbol. The number of unique non-zero Christoffel symbols required by the PUMA model can be reduced with four equations that hold on the derivatives of the kinetic energy matrix elements from 126 to 39. The first two equations are general, but the last two are specific to the PUMA 560. These equations are:

$$\frac{\partial A_{ij}}{\partial q_k} = \frac{\partial A_{ji}}{\partial q_k} \quad \forall i, j, k \tag{6}$$

$$\frac{\partial A_{ij}}{\partial q_k} = 0 \quad \forall i \geq k, j \geq k \tag{7}$$

$$\frac{\partial A_{ij}}{\partial q_6} = 0 \quad \forall i, j \tag{8}$$

$$\frac{\partial A_{13}}{\partial q_2} = \frac{\partial A_{13}}{\partial q_3} = \frac{\partial A_{12}}{\partial q_3} \tag{9}$$

The reduction of Equation (6) is because of the symmetry of the kinetic energy matrix. Equation (7) obtains because the kinetic energy explained by the velocity of a joint is independent of the configuration of the prior joints. The symmetry of the terminal and sixth link of the PUMA 560 arm results Equation (8). Because the second and third axes of the PUMA arm are parallel equation (9) holds. In step four the mass matrix elements should be differentiated with respect to the configuration variables.

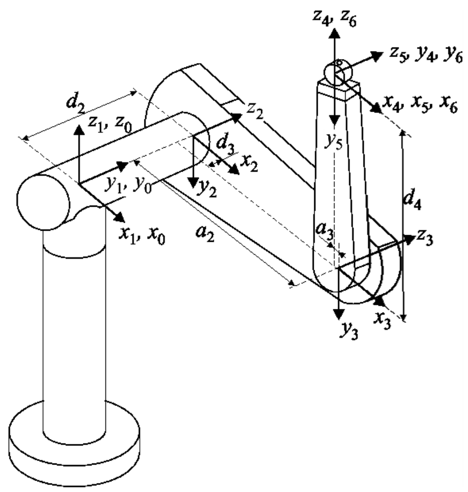

Fig. 2. *The local coordinate systems attached to the PUMA arm*

The mass of links two through six of the PUMA arm are shown in Table 1. The mass of link 1 is ignored because that link was not removed from the base. Because of rotating about its own 2 axis, separate measured mass and inertia terms are not considered for link one. The local coordinate systems are shown in Fig. 2.

Table 1: Link masses (Kg)

Link	Mass
Link 2	17.4
Link 3	4.8
Link 4	0.82
Link 5	0.34
Link 6	0.09
Link 3 with complete wrist	6.04
Detached wrist	2.24

The positions of the centers of gravity are included in Table 2. The dimensions r_x, r_y and r_z refer to the x, y and z coordinates.

Table 2: positions of the links centers of gravity

Link	r_x	r_y	r_z

Link 2	0.068	0.006	-0.0016
Link 3	0	-0.070	0.014
Link 3 with wrist	0	-0.143	0.014
Link 4	0	0	-0.019
Link 5	0	0	0
Link 6	0	0	0.032
Wrist	0	0	-0.064

III. ADAPTIVE CONTROLLER

The adaptation mechanism is used to adjust the parameters in the control law. In Dynamic systems, the adaptation law searches for parameters such that the response of the plant under adaptive control becomes the same as that of the reference model, i.e., the objective of the adaptation is to make the tracking error converges to zero.

As mentioned before, adaptive controller is capable of controlling the dynamic systems with parametric uncertainties, large initial condition errors. Fig. 3 shows the structure of a model reference adaptive controller. As shown in this figure by using two blocks $M(t)$, $K(t)$ the dynamic system under control will be adjustable and simply the tracking problem achieved by minimizing the difference between adjustable system and the reference model.

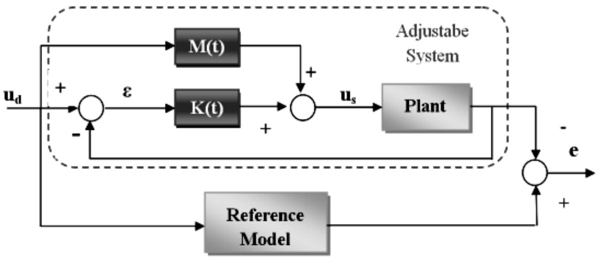

Fig. 3. *The structure of a model reference adaptive controller*

Before finding the appropriate control input, u_s, which is the torque applied to the robot arms, the error signal should be defined as

$$\varepsilon = \theta_d - \theta$$
$$\dot{\varepsilon} = \dot{\theta}_d - \dot{\theta} \qquad (10)$$

where θ and θ_d is actual and desired angular location, respectively. and

$$\theta = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6]^T, \theta_d = [\theta_{d1} \quad \theta_{d2} \quad \theta_{d3} \quad \theta_{d4} \quad \theta_{d5} \quad \theta_{d6}]^T \qquad (11)$$

The structure of adaptive input control torque has the form of

$$\tau(t) = u_s(t) = \begin{bmatrix} K_p & 0 \\ 0 & K_v \end{bmatrix} \begin{Bmatrix} \varepsilon(t) \\ \dot{\varepsilon}(t) \end{Bmatrix} + \begin{bmatrix} M_p & 0 & 0 \\ 0 & M_v & 0 \\ 0 & 0 & M_a \end{bmatrix} \begin{Bmatrix} \theta_d(t) \\ \dot{\theta}_d(t) \\ \ddot{\theta}_d(t) \end{Bmatrix} \qquad (12)$$

where K_p, K_v, M_p, M_v, M_a are adaptive parameters that should chosen properly. So, the control input can be wrote as

$$\tau(t) = u_s(t) = u_s(t) = K_P(t)\varepsilon(t) + K_v(t)\dot{\varepsilon}(t) + M_p(t)\theta_d(t) + M_v(t)\dot{\theta}_d(t) + M_a(t)\ddot{\theta}_d(t) \qquad (13)$$

This control law causes the system to be adjustable. By Substituting eq(6) in eq(1) we can obtain the dynamic model of robot as

$$M\ddot{\theta}(t) + C\dot{\theta} = K_P(t)\varepsilon(t) + K_v(t)\dot{\varepsilon}(t) + M_p(t)\theta_d(t) + M_v(t)\dot{\theta}_d(t) + M_a(t)\ddot{\theta}_d(t) \qquad (14)$$

where θ, $\dot{\theta}$ and $\ddot{\theta}$ is angular location, speed and acceleration, respectively. Substituting eq(3) in eq(7) gives

$$M\ddot{\varepsilon} + (K_v + C)\dot{\varepsilon} + K_p\varepsilon = (M - M_a)\ddot{\theta}_d + (C - M_v)\dot{\theta}_d - M_p\theta_d \qquad (15)$$

This is the dynamic state equation in terms of desired location and error. By defining the state variables as $x_s = [\varepsilon \quad \dot{\varepsilon}]^T$ the state space formulation will be

$$\dot{x}_s = \begin{bmatrix} 0 & I \\ -M^{-1}K_p & -M^{-1}(K_v + C) \end{bmatrix} x_s + \begin{bmatrix} 0 \\ -M^{-1}M_p \end{bmatrix} \theta_d + \begin{bmatrix} 0 \\ M^{-1}(C - M_v) \end{bmatrix} \dot{\theta}_d + \begin{bmatrix} 0 \\ M^{-1}(M - M_a) \end{bmatrix} \ddot{\theta}_d \qquad (16)$$

Introducing a second order system for dynamic reference model for each robot arms gives the freedom of tuning system properties such as overshoot, settling time and steady state error,

$$\ddot{x}_{mi} + 2\xi_i \omega_i \dot{x}_{mi} + \omega_i^2 x_{mi} = 0 \qquad i = 1,2,3 \qquad (17)$$

where ξ_i is damping ratio and ω_i is natural frequency of the robot arm. By defining the reference model state variables as $X_m = [x_{mi} \quad \dot{x}_{mi}]^T \quad i = 1,2,3,4,5,6$ the dynamic state space formulation for reference model will be

$$\dot{X}_m = \begin{bmatrix} 0_{6\times6} & I_{6\times6} \\ diag(-\omega_i^2)_{6\times6} & diag(-2\xi_i\omega_i)_{6\times6} \end{bmatrix} X_m \qquad (18)$$

By defining the adaptive error as $e = x_m - x$ the dynamics of the error function will be obtained as

$$\dot{e} = A_m e(t) + (A_m - A(t))X_s(t) + (B_m - B(t))u(t) \qquad (19)$$

The aim of the controller is to adapt the control law in a way that the dynamic closed-loop system becomes asymptotically stable in the zero equilibrium point. For achieving this aim one should choose the Lyapunov function of this dynamic as [25]

$$V = e^T Pe + tr\left\{(A_m - A(t))^T F_1^{-1}(A_m - A(t))\right\} + tr\left\{(B_m - B(t))^T F_2^{-1}(B_m - B(t))\right\} \qquad (20)$$

where P, F_1 and F_2 are positive definite matrix and V is a PDF. By differentiating Lyapunov function with respect to time one can obtain,

$$\dot{V} = e^T\left(A_m^T P + PA_m\right)e + 2tr\left\{(A_m - A(t))^T\left(PeX_s^T - F_1^{-1}\dot{A}(t)\right)\right\} +$$
$$+2tr\left\{(B_m - B(t))^T\left(Peu^T - F_2^{-1}\dot{B}(t)\right)\right\} \qquad (21)$$

As it assumed A_m is a Hurwitz matrix, so by setting the two last term in eq(14) equal to zero, we will have

$$tr\left\{\left(A_m - A(t)\right)^T \left(PeX_s^T - F_1^{-1}\dot{A}(t)\right)\right\} = 0$$
$$tr\left\{\left(B_m - B(t)\right)^T \left(Peu^T - F_2^{-1}\dot{B}(t)\right)\right\} = 0 \qquad (22)$$
$$\dot{V} = e^T\left(A_m^T P + PA_m\right)e = e^T\left(-Q\right)e$$

where Q is a positive definite matrix and the system is asymptotically stable using Lyapunov stability theorem. So in order to reach an asymptotically stable system we have

$$\left(PeX_s^T - F_1^{-1}\dot{A}(t)\right) = 0$$
$$\left(Peu^T - F_2^{-1}\dot{B}(t)\right) = 0 \qquad (23)$$

By some simplification one have

$$\dot{A}(t) = F_1\, PeX_s^T$$
$$\dot{B}(t) = F_2\, Peu^T \qquad (24)$$

By integrating eq (17) in time, it can easily concluded that

$$A(t) = F_1 P\int_0^t eX_s^T dt + A(0)$$
$$B(t) = F_2 P\int_0^t eu^T dt + B(0) \qquad (25)$$

By nearly long mathematical manipulation, the adaptive laws will obtain as

$$K_p(t) = \alpha\int_0^t ee^T dt + K_p(0)$$
$$K_v(t) = \beta\int_0^t e\dot{e}^T dt + K_v(0)$$
$$m_p(t) = \gamma\int_0^t e\theta_d^T dt + M_p(0) \qquad (26)$$
$$m_v(t) = \delta\int_0^t e\dot{\theta}_d^T dt + M_v(0)$$
$$m_a(t) = \mu\int_0^t e\ddot{\theta}_d^T dt + M_a(0)$$

The parameters $\alpha, \beta, \gamma, \delta$ and μ and the initial conditions of matrices K_p, K_v, m_p, m_v, m_a should be chosen in such way that the tracking error decreases as much as possible.

IV. Genetic Algorithm

There exist heuristic search methods that group individual solutions (states) in sets or multi-sets, referred to as populations, swarms, etc [26]. Such algorithms are based on generating a succession of populations in which each of them should be better than the previous one, that is, should have better individuals. Genetic algorithms (GA) have been proved to suitably work on a wide range of optimization problems from different fields, so this method is used here for finding desired parameters. However, for finding the global best answer to this optimization we must solve a nonlinear optimization due to the robot dynamic model. Solving this optimization problem is the first application of the used genetic algorithm in this paper. However, at the last part of the paper we return to the genetic algorithm to solve the path-planning problem.

Algorithm genetic uses the following procedure toward the best solution of these optimization problems:

a. Initially many individual solutions are arbitrarily generated in the search space to form an initial population. The population size depends on the complexity of the problem and the search space and typically contains several hundreds of solutions covering the entire range of possible solutions.

b. During each generation, a group of the existing population is selected to generate a new breeding. These solutions are selected through evaluation in a fitness function, where better solutions are normally more likely to be selected. Definite selection methods rate the fitness of each solution and choose the best solutions. This procedure is more global in the case of using stochastic function to keep the diversity of the population well studied and large, preventing premature convergence on local optimal solutions. Some algorithms include roulette wheel selection and tournament selection.

c. Next is to generate a second generation of solutions from the selected ones in genetic operators: crossover and mutation. For each new solution to be generated, a pair of "parent" solutions is selected for reproduction from the space selected before. By generating a "child" solution using the aforementioned methods of mutation and crossover, a new solution are produced which normally shares many of the characteristics of its "parents". This procedure continues until the selected population size achieves and a new generation of solutions generated. Although reproduction methods that are based on the use of two or more than two parents lead to better solutions and increase in quality of chromosome. Generally, the average fitness evaluation will increase by this repetitive procedure for each population, since the algorithm tends to inspire the solutions toward populations with higher fitness functions.

d. This repetitive process is repeated until a termination condition has been achieved. Some of the common terminating situations are listed here:

- A solution satisfies minimum criteria
- Fixed number of generations
- Computation time reached

• Combinations of the above

Within robotics, inverse dynamics algorithms are used to calculate the torques that a robot's motors must deliver to make the robot's end-point move in the way prescribed by its current task. It commonly refers to either inverse rigid body dynamics or inverse structural dynamics. Inverse rigid-body dynamics is a method for computing forces and/or moments of force (torques) based on the kinematics (motion) of a body and the body's inertial properties (mass and moment of inertia). Typically, it uses link-segment models to represent the mechanical behavior of interconnected segments, where given the kinematics of the various parts, inverse dynamics derives the minimum forces and moments responsible for the individual movements. In practice, inverse dynamics computes these internal moments and forces from measurements of the motion of external forces such as ground reaction forces, under a special set of assumptions. For this purposes one should solve the kinematic equation for 6 degree of freedom robot to find the torques that should apply on each joint in order to translate the end effector to the prescribed location. In PUMA 560 the transformation matrix using Forward Kinematics (FK) constructed by Denavit-Hartenberg (D-H) formulation from base to the end effector is as follow

$$T_1=\begin{bmatrix} Cos(\theta_1) & 0 & Sin(\theta_1) & 0 \\ Sin(\theta_1) & 0 & -Cos(\theta_1) & 0 \\ 0 & 1 & 0 & 0.67 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_2=\begin{bmatrix} Cos(\theta_2) & 0 & -Sin(\theta_2) & 0.4318Cos(\theta_2) \\ Sin(\theta_2) & Cos(\theta_2) & 0 & 0.4318Sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3=\begin{bmatrix} Cos(\theta_3) & 0 & -Sin(\theta_3) & 0.4318Cos(\theta_3) \\ Sin(\theta_1) & 0 & -Cos(\theta_1) & 0.4318Sin(\theta_3) \\ 0 & -1 & 0 & 0.15005 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_4=\begin{bmatrix} Cos(\theta_4) & 0 & -Sin(\theta_3) & 0 \\ Sin(\theta_4) & 0 & -Cos(\theta_1) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

$$T_5=\begin{bmatrix} Cos(\theta_5) & 0 & -Sin(\theta_5) & 0 \\ Sin(\theta_5) & 0 & -Cos(\theta_5) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_6=\begin{bmatrix} Cos(\theta_6) & 0 & -Sin(\theta_6) & 0 \\ Sin(\theta_6) & 0 & Cos(\theta_6) & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdot T_5 \cdot T_6$$

In which the constants are obtained from the D-H parameters in Table 3.

Table 3. *D-H parameters*

Link	$a(m)$	$\alpha(\deg)$	$d(m)$
1	0	0	0
2	0	-90	0.3435
3	0.4318	0	-0.0934
4	-0.0203	90	0.4331
5	0	-90	0
6	0	90	0

As one can see, there are 6 unknown variables and 3 nonlinear equations to find these variables which cannot be solved analytically. For this reason, an optimization problem of minimizing the error between desired end effector coordinate and the current end effector location constrained by using the maximum energy of each joints from each move to another is defined. This constraint will prevent the undesired jumps, saturation of the actuators and sudden movements of each joint. A constrained genetic algorithm as below is used to solve the addressed nonlinear optimization problem,

$$\begin{aligned} &\min \quad \|K.E\|_2 \\ &\text{Constrained to} \qquad (28) \\ &\|\theta_i - \theta_{di}\|_\infty \le \varepsilon_i \text{ and } \|T\| \le t_i, i=1,2,...,6 \end{aligned}$$

where $\| \ \|_\infty$ and $\| \ \|_2$ define the infinity norm and 2-norm, respectively, and ε_i shows the bound on each joint which is selected as [27] and $t_i, i=1,2,...,6$ selected such as the actuator specification given in Table 4.

Table 4. *Actuator specification*

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Gear Ratio	62.61	107.36	53.69	76.01	71.91	76.73
Maximum Torque(N-m)	97.6	180.4	30	24.2	20.1	21.3
Break Away Torque (N-m)	6.3	5.5	2.6	1.3	1.0	1.2

V. SIMULATION

A. *Definition of reference model system*

An adaptive model reference controller for PUMA 560 has the aim of forcing the actual robot to track the ideal reference model dynamic behavior, so we have to tune the reference model as the ideal model that acts exactly in the way that is desirable. Reference models for different dynamic systems are usually predefined and for robotic systems it is generally a second order system. In order to achieve the desired response in the reference model we choose the system parameters such as ξ, ω in a way that the settling time is less than 1 second and the response does not have any overshoot and undershoot. These constraints achieved by choosing the second order system parameters as below for all of the robot arms:

$$\xi = 0.95, \omega_n = 10^{(rad/s)} \qquad (29)$$

Therefore the transfer functions for all of the reference models are:

$$\frac{\theta_m}{\theta_d} = \frac{100}{s^2 + 19s + 100} \qquad (30)$$

Genetic algorithm for adaptive control Parameters

As shown in eq (19) the optimization problem has 185 unknown optimization states. Each of the matrices K_p, K_v, M_p, M_v, M_a has 36 unknown parameters together with the 5 unknown parameters of $\alpha, \beta, \gamma, \delta$ and μ. Because of very large initial population that is needed for this problem, the suboptimal optimization is solved, instead. This suboptimal problem is constructed by assuming all of the optimization matrices as diagonal ones which decreases the optimization variables to 35 variables. With the genetic algorithm fitness function as Euclidian norm of error between desired states and system responses, the results will be as shown in appendix A. Table 5 indicates

Table 5. *GA-Optimization Parameters*

Population size	generation	fitness scaling
70	20	Rank
selection function	elite count	crossover fraction
stochastic uniform	10	0.8
crossover function	Migration direction	migration fraction
Scattered	forward	0.2

Also, Fig. 4 shows the optimization fitness function per each generation.

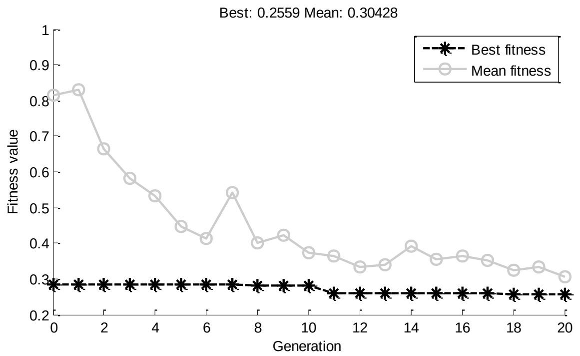

Fig. 4. *Optimization fitness function per each generation*

More than 20 times of the optimization procedure is done to assure that the local minimization was not happened. The optimized adaptive controller and the PUMA 560 model are implemented in MATLAB. Robot parameters and constraints are chosen as the given in Table 1 to Table 4. The closed-loop system response due to step input is shown in fig. 5 and Fig. 6. These results reveal that the error of the tracking reaches to zero in a finite time without any sudden jumping or chattering.

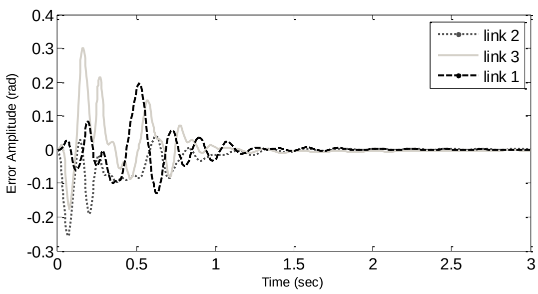

Fig. 5. *Error amplitude for first 3 robot links*

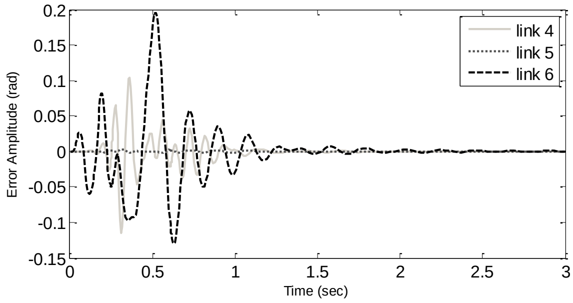

Fig. 6. *Error amplitude for second 3 robot links*

Also, the control inputs are shown in Fig. 7 and Fig. 8 and the results show that the control input is smooth enough to be implemented.

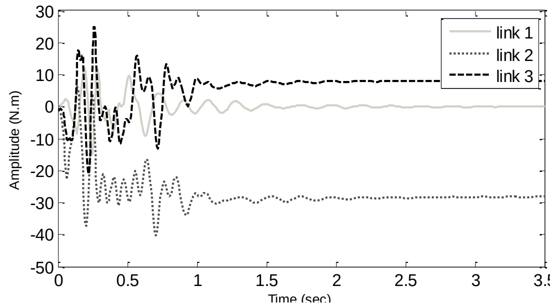

Fig. 7. *The control input for first 3 robot actuators*

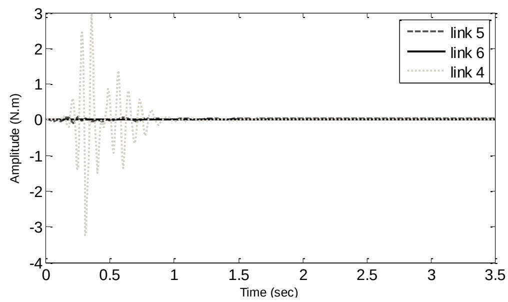

Fig. 8. *The control input for second 3 robot actuators*

These results achieved by assuming more powerful actuators on the robot joints than PUMA 560. By using the traditional actuators of PUMA 560 with torque constraints as Table 4, control inputs as Fig. 9 and Fig. 10 will be achieved.

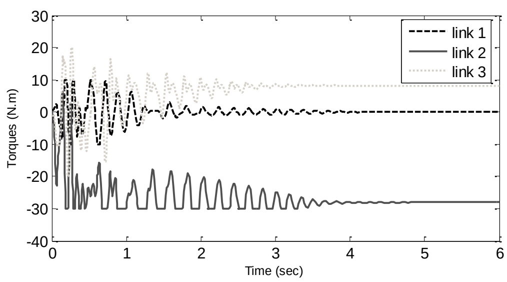

Fig. 9. *The control input for first 3 robot actuators*

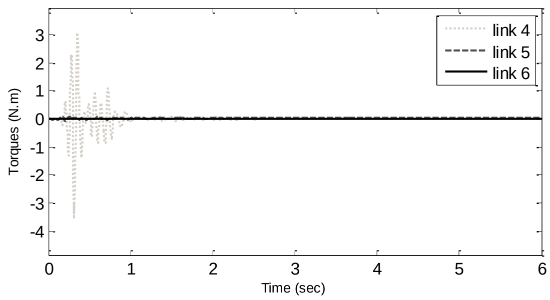

Fig. 10. *The control input for first 3 robot actuators*

For investigating the tracking performance of the robot in more detail, the closed loop system is forced to track a sinusoidal signal as Fig. 11.

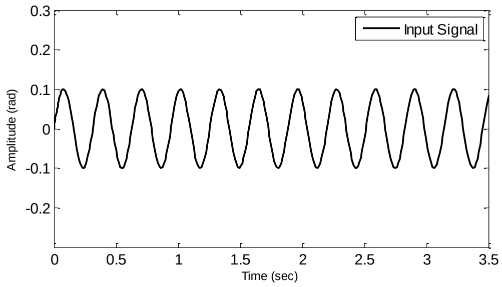

Fig. 11. *Desired signal for tracking test*

And the tracking error signals for this reference signal are shown in Fig. 12 and Fig. 13.

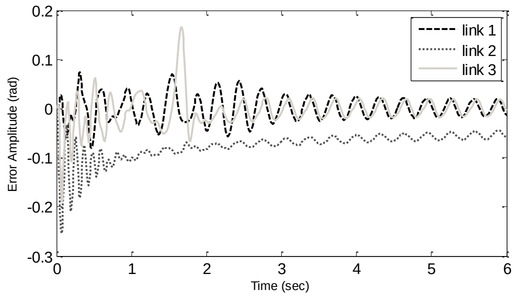

Fig. 12. *The error signal for first 3 robot actuators*

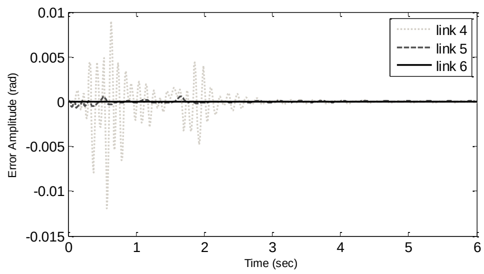

Fig. 13. *The error signal for second 3 robot actuators*

It can be concluded from Fig. 5 and Fig. 12 that the three first actuators from the base have a worse performance than the second three actuators. This may because of more inertia in the first three arms from the basement. Also, Fig. 14 and Fig. 15 show signal by assuming a more powerful actuator in the third joint.

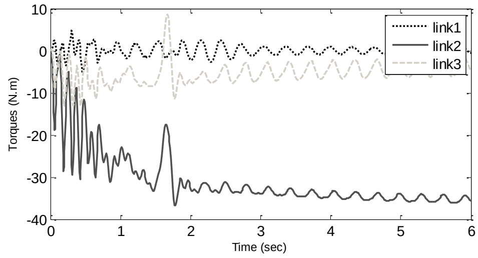

Fig. 14. *The control input signals during the sinusoidal tracking for first 3 robot actuators*

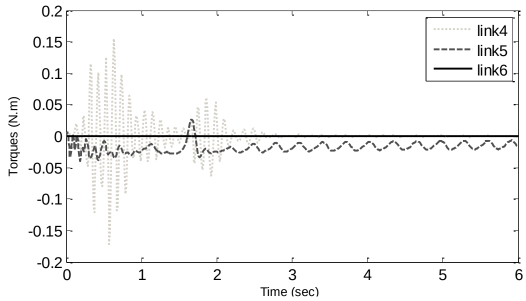

Fig. 15. *The control input signals during the sinusoidal tracking for second 3 robot actuators*

As one can see these control inputs are smooth without any sudden jump. For investigating the system robustness due to the input disturbance a sudden loading is given. For this aim, the system is disturbed by a sudden 0.1sec loading in the time interval $[2.6 \quad 2.7]\text{sec}$ during tracking a step reference signal. The obtained results for tracking error are shown as Fig. 16 and Fig. 17.

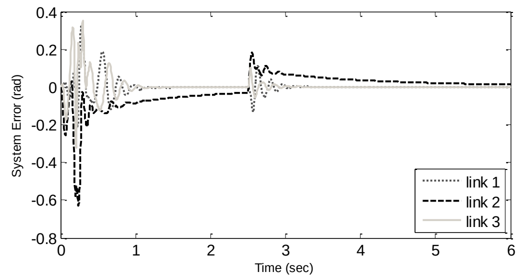

Fig. 16. *The error signal for second 3 robot actuators*

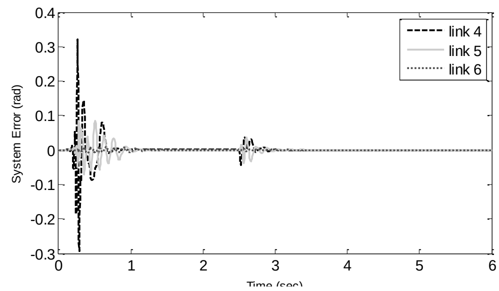

Fig. 17. *The error signal for second 3 robot actuators*

As shown in these figures after the disturbance is removed the system returns to the zero tracking error by less than one second. The control inputs in this case are depicted as fig. 18 and Fig. 19.

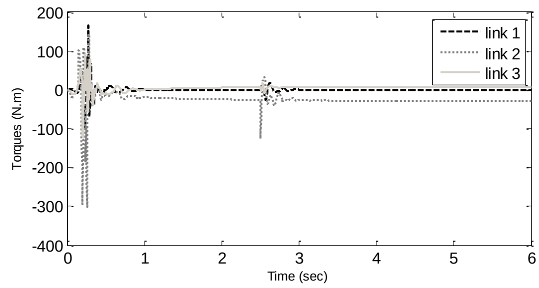

Fig. 18. *The control input for first 3 robot actuators*

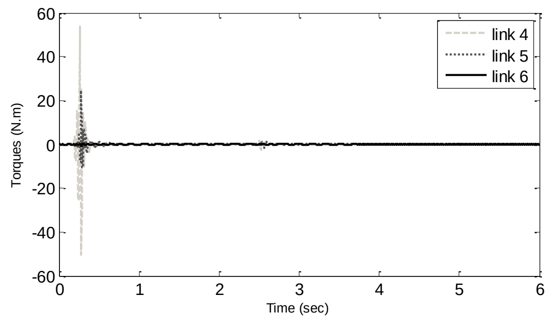

Fig. 19. *The control input for second 3 robot actuators*

For testing closed-loop robot inverse dynamic problem, the GA-Solver on the system output is implemented. This solver gives the system calculated appropriate reference input signal. The desired path is divided to 148 points and between each of these points the GA-solver is activated. In order to show the optimization procedure in this step, the optimization plot between the point 147 to 148 is shown in Fig. 20.

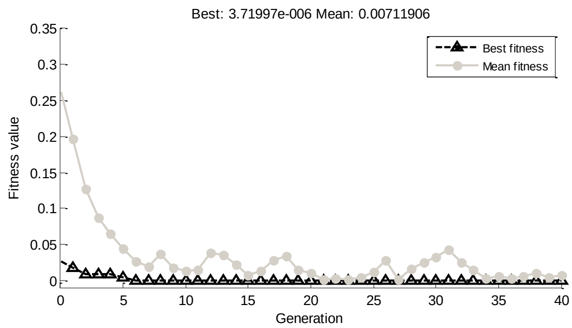

Fig. 20. *The optimization value between the point*

Fig. 21 shows the desired path and the robot path. For this simulation, a proper time interval is considered in each step in order to achieve the steady state response. As it can seen in this figure the system tracks the desired path in an acceptable way.

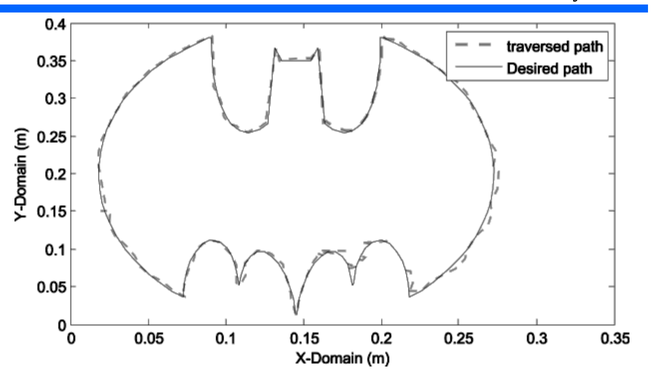

Fig. 21. *Path tracking of the robot*

VI. CONCLUSION

After implementation of the robot dynamic model and the adaptive model reference using MATLAB SIMULINK Lab the performance of the closed-loop system in different working situations such as step and sinusoidal reference input is discussed. In the simulation, the effect of sudden disturbances such as sudden loading which can happen in actual applications is considered and the results are shown. Finally, the results of the genetic algorithm in solving the inverse dynamic problem for path planning is depicted. The performance of the robot in tracking a reference path is investigated. Consequently, it is concluded that the Puma 560 robot with GA-adaptive model reference controller with GA-based path planning has the desired performance in passing through difficult paths.

REFERENCES

[1] J-C. Latombe, “Robot motion planning”, Boston, Kluwer, 1991.

[2] E. Mohammadpour, M. Naraghi, and M. Gudarzi, "Posture stabilization of skid steer wheeled mobile robots." Robotics Automation and Mechatronics (RAM), 2010 IEEE Conference on. IEEE, 2010.

[3] M. Gudarzi, A. Oveisi, and M.M. Mohammadi, "Robust active vibration control of a rectangular piezoelectric laminate flexible thin plate: an LMI-based approach." International Review of Mechanical Engineering Vol. 6, No. 6, pp. 1217-1227, 2012.

[4] M. Gudarzi, and A. Oveisi, "Robust Control for Ride Comfort Improvement of an Active Suspension System considering Uncertain Driver's Biodynamics." Low Frequency Noise, Vibration and Active Control, Vol. 33, No. 3, pp. 317-340, 2014.

[5] M. Gudarzi, and A. Oveisi, "Ride Comfort Performance of Active Vehicle Suspension with Seat Actuator Based on Non-Fragile H∞ Controller”, International Review on Modelling and Simulations (IREMOS), Vol. 8, No. 1, pp. 90-98, 2015.

[6] M. Gudarzi, "µ-Synthesis Controller Design for Seismic Alleviation of Structures with Parametric

Uncertainties." Journal of Low Frequency Noise, Vibration and Active Control Vol. 34, No. 4, pp. 491-511, 2015.

[7] B. Armstrong, O. Khatib, and J. Burdick, "The explicit dynamic model and inertial parameters of the PUMA 560 arm", Robotics and Automation. Proceedings. IEEE International Conference on, pp. 510 – 518, 1986.

[8] A. Oveisi, and M. Gudarzi, "Adaptive sliding mode vibration control of a nonlinear smart beam: A comparison with self-tuning Ziegler-Nichols PID controller." Journal of Low Frequency Noise, Vibration and Active Control Vol. 32, No. 1-2, pp. 41-62, 2013.

[9] A. Oveisi, and M. Gudarzi, "Nonlinear robust vibration control of a plate integrated with piezoelectric actuator." International Journal of Mathematical Models and Methods in Applied Sciences, Vol. 7, No. 6, pp. 638-646, 2013.

[10] A. Oveisi, M. Jarrahi, M. Gudarzi, and M.M. Mohammadi, "Genetic Algorithm and Adaptive Model Reference Controller in Tracking Problem of PUMA 560 Arm Robot." First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM 2013), Sharif University of Technology, Tehran, Iran, February 13-15, 2013.

[11] A. Oveisi, and T. Nestorović. "Robust mixed h2/h8 active vibration controller in attenuation of smart beam" Facta Universitatis, Series: Mechanical Engineering, Vol. 12, No. 3, pp. 235-249, 2014.

[12] B. Mashadi, M. Mahmoudi-Kaleybar, P. Ahmadizadeh, and A. Oveisi, "A path-following driver/vehicle model with optimized lateral dynamic controller", Latin American journal of solids and structures, Vol. 11, No. 4, pp.613-630, 2014.

[13] V. Tavoosi, R. Kazemi, and A Oveisi. "Nonlinear adaptive optimal control for vehicle handling improvement through steer-by-wire system", Journal of Central South University, Vol. 21, No.1, pp. 100-112, 2014.

[14] A. Oveisi, And M. Gudarzi, "Application of auto-tuning ziegler-nichols pid control in chaotic vibration suppression of a beam with nonlinear boundary conditions", Latin American Applied Research, Vol. 45, No.3, pp. 179-184, 2015.

[15] L. Gracia, F. Garelli, and A. Sala, "A path conditioning method with trap avoidance", Robotics and Autonomous Systems, Vol. 60, pp. 862–873, 2012.

[16] Z. Liu, W. Bu, and J. Tan, "Motion navigation for arc welding robots based on feature mapping in a simulation environment", Robotics and Computer-Integrated Manufacturing, Vol. 26, pp. 137-144, 2010.

[17] J. Gomez Ortega, J. Gamez Garcia, S. Satorres Martinez, and A. Sanchez Garcia, "Industrial assembly of parts with dimensional variations. case study: assembling vehicle headlamps", Robotics and Computer-Integrated Manufacturing, Vol. 27, pp. 1001-1010, 2011.

[18] E. Pitschke, M. Schinhaerl, R. Rascher, P. Sperber, L. Smith, R. Stamp, et al., "Simulation of a complex optical polishing process using a neural network", Robotics and Computer-Integrated Manufacturing, Vol. 24, pp. 32-37, 2008.

[19] B. Achili, B. Daachi, Y. Amirat, and A. Ali-Cherif, "A robust adaptive control of a parallel robot", International Journal of Control, Vol. 83, pp. 2107-2119, 2010.

[20] S.S. Ge, and T.H. Lee, "Robust model reference adaptive control of robots based on neural network parametrization", Proceedings of the American Control Conference, pp. 2006-2010, 1997.

[21] V.K. Banga, R. Kumar, and Y. Singh, "Fuzzy-genetic optimal control for robotic systems", International Journal of Physical Sciences, Vol. 6, pp. 204-212, 2011.

[22] Z. Cao, Y. Zhao, and Q. Wu, "Genetic Fuzzy + PI Path Tracking Control of a Nonholonomic Mobile Robot", Chinese Journal of Electronics, Vol. 20, pp. 31-34, 2011.

[23] W.S. Yoo, H.-J. Lee, D.-I. Kim, K.-I. Kang, and H. Cho, "Genetic algorithm-based steel erection planning model for a construction automation system", Automation in Construction, Vol. 24, pp. 30-39, 2012.

[24] J. Wang, B. Ding, G. Miao, J. Bao, and X. Yang, "Path planning of mobile robot based on improving genetic algorithm", Advances in Intelligent and Soft Computing, Vol. 112, pp. 535-542, 2011.

[25] J.-J. Slotine, and W. Li, "Applied Nonlinear Control", Prentice Hall, 1991.

[26] P. Larrañaga, C.M.H. Kuijpers, R.H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators", Artificial Intelligence Review, Vol. 13, pp. 129-170, 1999.

[27] L.W. Khalil, J.M. Dumas, and M. Renaud, "Mathematical and computer models of interconnected mechanical systems", Proc. Symp. Theory and Practice of Robots and Manipulators, Warsaw, Poland, pp. 5-17, 1976.