A New LQR Optimal Control for a Single-Link Flexible Joint Robot Manipulator Based on Grey Wolf Optimizer

Navid Razmjooy[1], Mehdi Ramezani[2], Ali Namadchian[1]
1- Department of Electrical Engineering, University of Tafresh, Tafresh, Iran.
Email: navid.razmjooy@hotmail.com (Corresponding author), Ali_namadchian@yahoo.com
2- Department of Mathematics, University of Tafresh, Tafresh, Iran.
Email: ramezani@aut.ac.ir

Received: June 2016 Revised: July 2016 Accepted: August 2016

ABSTRACT:
Flexible manipulators are very commonly used in industries. In this paper a single-link flexible joint robot is modeled firstly by using Euler–Lagrange energy equation. An optimized Linear Quadratic Regulator is employed to control the manipulator. After that, a Linear Quadratic Regulator (LQR) controller is used for optimal control of the manipulator. For optimizing the LQR, the regulator term weighting of the LQR is achieved by using the newly introduced grey wolf optimizer technique. With the optimized LQR controller based on the proposed performance index, it is tried to have a system with the minimum overshoot and settling time. By considering the proposed performance index and comparing with the PSO-based controller as a popular algorithm, the superiority of the proposed LQR controller in improving the stability and performance of the manipulator is shown. The simulations are performed in MATLAB environment and the results confirm the efficiency of the proposed controller.

KEYWORDS: Single-link flexible-joint Manipulator, Linear Quadratic Regulator, Optimal Control, Gray Wolf Optimizer, Particle Swarm Optimization.

1. INTRODUCTION

Nowadays, industrial manipulator robots are used in many branches of manufacturing for tasks such as robotic welding and automated assembly. The flexibility of the manipulator joints is often left un-controlled and un-modeled leading to performance restrictions [1, 2].

This flexibility is made by the gears and belts employed to transmit the actuators generated torque to the links [2]. The natural frequencies of these joints are often relatively low (2–3 Hz), which often synchronize by the trajectory frequency being followed, forcing the operator to wait for any vibrations to decay naturally [3].

Furthermore, in the large scale manipulators, even a relatively small joint flexibility can make considerable vibrations at the manipulator tip, which is highly unfavorable. Hence, understanding and developing the mathematical model for a single link manipulator robot and control the manipulator is an important category for control system engineers.

Several studies about the modeling and control of flexible-joint robotic manipulators are applied [4–8]. Most of the researches tend to control the position of the end-effector in the manipulators robot.

Control techniques which are developed for the present flexible systems, have a lot of limitations in precision and performance [9]. These limitations for the control of the manipulator can be solved by using different control structures.

There are different techniques which can solve this problem and control the single-link manipulators [10]. In 1990, Wang and Liu presented an optimal and robust controller based on H2 technique and a quadratic performance index in frequency domain [10]. Berger and El Maraghy proposed a feedback linearization method for designing a controller in order to stabilize nonlinear modes of the system; LQR is used for the fast term [11].

Feedback linearization method is also utilized for the manipulator robot in [12]. Huang and Chen presented an adaptive sliding controller for a single-link flexible joint robot with mismatched uncertainties [13]. Taghirad and Bakhshi proposed a hybrid controller technique by using a linear H∞ controller for the rigid part in the presence of actuator saturation [14].

The results showed how actuator restrictions enforces performance degradation in the framework. The authors developed the research in [15] by adding an H2 performance index to the cost function for minimizing the amplitude of control attempt. In the presented research, they introduced a mixed H2/H∞

controller.

Drapeau and Wang presented a five bar manipulator with one flexible link and used a closed-loop shaped-input technique in conjunction with a rigid body LQR regulator to control the vibration.

This paper examines the suitability and evaluates the performance of an optimized Linear Quadratic Regulator (LQR) for controlling the manipulator. The LQR controller employs several weighting matrices to achieve the appropriate control force to be implemented to the system. The main contribution of this paper is to design an optimized LQR controller by using the new introduced Gray Wolf Optimizer algorithm to control a single-link flexible manipulator.

This paper will discuss about how to optimal control of a single-link flexible joint robot by a new optimized Linear Quadratic Regulator technique. The rest of the paper is organized as follow: Sect. 2 shows the mathematical modelling of the considered system in details. Section 3 describes the Schur decomposition for reducing the system order and for simplifying the designed controller. Then Sect. 4 has showed a detail description of the required methods including PSO, GWO and LQR techniques for controlling the considered system. Finally, in Sect. 5 results are shown and the paper closes by conclusion in Sect. 6.

2. MATHEMATICAL MODELING FOR THE SINGLE-LINK FLEXIBLE-JOINT MANIPULATOR

By considering a single-link flexible-joint manipulator, we can achieve the mathematical model by the Lagrange equations. The system has a two degree of freedoms with the joint which is fixed to the shaft moved in order to the rotate direction of the motor.

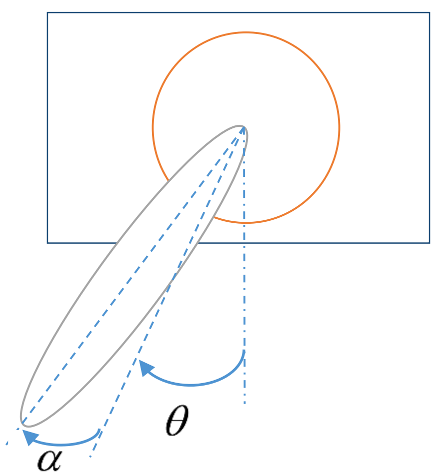

Fig. 1. Single-link flexible-joint manipulator model coordinates for the link

Fig.1 shows the rotation angle (θ) and the oscillation angle of the end effector.
The total energy for the potential and the kinetic energies is equal to:

$$L = K - P \tag{1}$$

Where K and P describe the kinetic and potential energies respectively. Therefore, the Lagrange equations of motion can be described as eq.2.

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\alpha}} - \frac{\partial L}{\partial \alpha} = 0 \qquad \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = r \tag{2}$$

After achieving the Lagrange for the manipulator, we have:

$$J_l\ddot{\theta} + J_l\ddot{\alpha} + K_s\alpha - mgh\sin(\theta+\alpha) = 0 \qquad (J_l + J_h)\ddot{\theta} + J_l\ddot{\alpha} + -mgh\sin(\theta+\alpha) = \tau \tag{3}$$

Here, τ defines the motor generated torque. The torque is obtained by the applied voltage *v* to the armature and illustrates the control input to the system. The relationship between the torque and the applied voltage is:

$$\upsilon = iR_m + K_m K_g \omega \qquad i = \frac{\upsilon}{R_m} - \frac{K_m K_g}{R_m}\omega \tag{4}$$

Where ω is the angular velocity of the motor and R_m is the motor resistance and i defines the armature current. Also K_m and K_g are the constant parameters respectively. Therefore:

$$i = \frac{\tau}{K_m K_g} \qquad \dot{\theta} = \omega \tag{5}$$

And the desired relationship equation is:

$$\tau = \frac{K_m K_g}{R_m}\upsilon - \frac{K_m^2 K_g^2}{R_m}\dot{\theta} \tag{6}$$

By determining the state variables as eq.7.

$$x_1 = \theta \qquad x_2 = \alpha \qquad x_3 = \dot{\theta} \qquad x_4 = \dot{\alpha} \tag{7}$$

The system will be transformed in the form eq.8.

$$\begin{aligned}
\dot{x}_1 &= x_3 \\
\dot{x}_2 &= x_4 \\
\dot{x}_3 &= \frac{K_s}{J_h} x_2 - \frac{K_m^2 K_g^2}{R_m J_h} x_3 + \frac{K_m K_g}{R_m J_h} \upsilon \\
\dot{x}_4 &= -\frac{K_s}{J_h} x_2 + \frac{K_m^2 K_g^2}{R_m J_h} x_3 - \frac{K_m K_g}{R_m J_h} \upsilon - \frac{K_s}{J_l} x_2 + \frac{mgh}{J_l}
\end{aligned} \quad (8)$$

By considering v as the system input (u) and $y = x_1 + x_2$ as the system output, the final form will be as eq.9.

$$\begin{aligned}
\dot{x} &= f(x) + g(x)u \\
y &= h(x)
\end{aligned} \quad (9)$$

The parameters of the system are extracted from the [17] and are summarized in the Table 1:

Table 1. Flexible Joint Robot Manipulator Parameters

Symbol	Description	Value
J_{link}	Inertia of Flexible Manipulator	0.003882 kgm^2
R_m	Motor Resistance	15.5 Ω
K_g	Gear Ratio of Reductor	1.36
K_m	Motor Constant	0.0089 $N/(rad/sn)$
K_s	Flexibility Coefficient of Joint	5.468 N/m
M	Mass of the Flexible Joint	0.03235 kg
G	Gravitional Acceleration	–9.81 N/m
H	Distance to Center of Gravity of Rotational Platform of Flexible Manipulator	0.06 m
J_h	Inertia of Rotational Platform	0.00035 kgm^2

3. LINEARIZATION AND ORDER REDUCTION

After mathematical modeling of the system, for using the LQR mode controller, it is essential to have a linear equivalent for the nonlinear system, by considering the following assumptions:

$$a = \frac{K_s}{J_h};\ b = -\frac{K_m^2 K_g^2}{R_m J_h};\ c = -a = -\frac{K_s}{J_h} \quad (10)$$
$$d = -\frac{K_s}{J_l};\ e = -b = \frac{K_m^2 K_g^2}{R_m J_h}; f = \frac{mgh}{J_l};$$
$$m = -n = \frac{K_m K_g}{R_m J_h}$$

The system is linearized about the equilibrium $(x_{ss}, u_{ss}) = 0 \Rightarrow (x_1, x_2, x_3, x_4) = (0,0,0,0)$. After linearization around the equilibrium points, we have:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 2.23 & 0.06 & 34838.7 \\ 0 & -2.23 & 0.06 & 0 \\ 2.23 & 0.06 & -15623.28 & 0 \\ -2.23 & 0.06 & 4.9 & 941.25 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 2.23 \\ -2.23 \end{bmatrix} v \quad (11)$$
$$y = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

For develop the system controller as a rapid controller, the Schur decomposition is utilized. This algorithm is an order reduction algorithm for simplifying the considered input matrix. The Schur decomposition for a complex square matrix A is a matrix decomposition of the form

$$Q^H A Q = T = D + N \quad (12)$$

where Q defines a unitary matrix, Q^H is its conjugate transpose, and T describes an upper triangular matrix which is the sum of a $D = diag(\lambda_1, \lambda_2, \cdots, \lambda_n)$ (i.e., a diagonal matrix consisting of eigenvalues of λ_i A) and a strictly upper triangular matrix N.The first phase in the Schur decomposition is the Hessenberg decomposition [18-20]. Schur decomposition on an n×n matrix requires $O(n^3)$ operations. Hankel singular values of the system are shown in the Fig.2. As it can be seen, the system can be reduced to 3 orders by very low error (9.2397×10^{-8}).

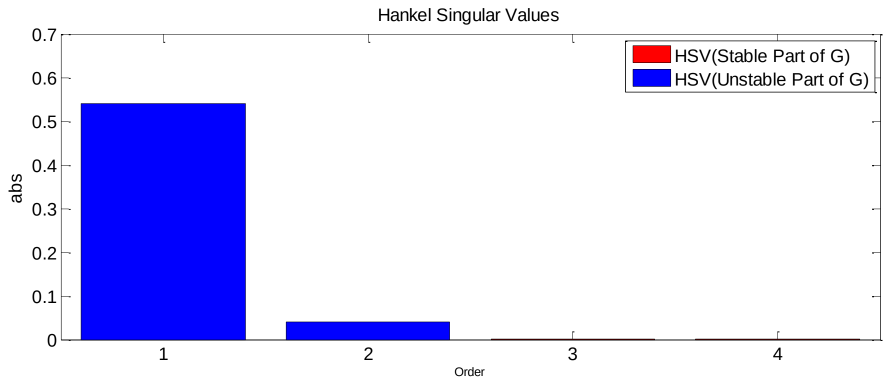

Fig. 2. Hankel singular values for the single-link flexible-joint manipulator

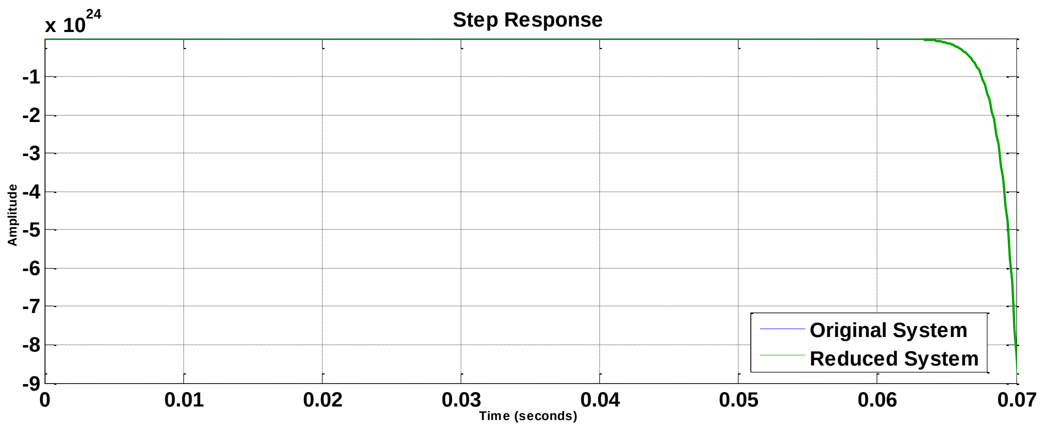

Fig. 3. step response for the Original and the reduced system by the Schur decomposition

After reducing the system by the Schur's decomposition, the 3 orders system is equal to:

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \dot{\hat{x}}_3 \end{bmatrix} = \begin{bmatrix} -996.7170 & 0 & 0 \\ 0 & 849.8620 & -553.1355 \\ 0 & 0 & 91.3865 \end{bmatrix} \times \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix} + \begin{bmatrix} 9.2328 \\ -15.2145 \\ 4.7312 \end{bmatrix} v \tag{13}$$

$$y = \begin{bmatrix} 0 & 5.4862 & 17.6419 \end{bmatrix} \times \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix}$$

From the Fig.3, we can conclude that due to the low error, the step response for the original and the reduced systems are overlapping each other.

By achieving a linear and simple enough system, we are ready to design a controller to balance the single-link flexible joint.

4. MATERIAL AND METHODS

4.1. Particle Swarm Optimization

In the last decades, meta-heuristic algorithms have been employed to solve most of the optimization problems. Particle Swarm Optimization (PSO) algorithm is one of the most popular optimization algorithms which has been presented in 1995 by Kennedy and Eberhart [23].

PSO algorithm is inspired by the social behavior of swarm of fish, bees and other animals [24]. In PSO, optimal solution to a mathematical optimization problem is limited of birds behave in the moment the food pursues, the escape from hunters and the search for mates.

Ordinary PSO algorithm starts with an initial population (swarm) of candidate solution (particles). The particles search throughout the search space due to described formulations. After searching, they move to their own best known position in the search space and the swarm's best known position.

After finding the best position, they will then guide the other particles movements. The searching about the search space is repeated until the satisfactory solution will eventually be detected. In each iteration, the swarm can be adjusted by the following equations:

$$v_i^{t+1} = w.v_i^{t} + c_1 r_1 (p_i^{t} - x_i^{t}) + c_2 r_2 (g_i^{t} - x_i^{t}) \quad (14)$$

$$x_i^{t+1} = x_i^{t} + v_i^{t+1} \quad i = 1,2,...,n \quad (15)$$

Where, n defines the number of particles, w is the weighted inertia, C1 and C2 describe the positive constants, r1 and r2 are two random numbers distributed within the range [0,1], t illustrates the iteration number, Pi is the best position of the ith particle and gi is the best particle among the group members.

According to the eq.19, the particles update their velocity due to the distances and the previous velocity to their current position from both the own best historical position and the best positions of the neighbors in every iteration step, and then they move into the new position given by (20).

4.2. Grey Wolf Optimizer (GWO)

Grey wolf is a new meta-heuristic algorithm which is introduced in 2014 by Mirjalili et al. [25]. GWO algorithm is inspired from the grey wolves' life. The method simulates the social hierarchy and hunting behavior in the grey wolves' society. For mimicking the leadership hierarchy in GWO algorithm, four groups are introduced: alpha, beta, delta, and omega. Also, the GWO has three main steps for hunting, including:

1) Tracking, chasing, and approaching the victim.
2) Pursuing, encircling, and harassing the victim until it stops moving.
3) Attack towards the victim.

GWO algorithm, like other meta-heuristic algorithms, needs a number of parameters to be set, consist of: initialize alpha, beta, and delta, number of search agents, maximum number of iterations, number of sites selected for neighborhood search (out of n visited sites) and the stopping criterion.

For simulating the social hierarchy of wolves until designing GWO, the best solution is considered as the alpha (α). Thereupon, the beta (β) and delta (δ) are introduced as the second and the third best solutions, respectively. The rest of the candidate solutions are considered to be omega (ω).These three wolves lead the other (x) wolves. Afterwards, for simulating the encircling behavior, we have:

$$\vec{D} = \left| \vec{C}.\overrightarrow{X_P}(t) - \vec{X}(t) \right| \quad (16)$$

$$\vec{X}(t+1) = \overrightarrow{X_P}(t) - \vec{A}.\vec{D} \quad (17)$$

Here, t defines the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\overrightarrow{X_P}(t)$ defines the position vector of the victim. The vectors $\vec{A}$ and $\vec{C}$ can be evaluated from the equation eq.18 and eq.19.

$$\vec{X} = 2\vec{a}.\vec{r_1} - \vec{a} \quad (18)$$

$$\vec{C} = 2.\vec{r_2} \quad (19)$$

Where $\vec{a}$ is in the interval from 2 to 0 over the course of iterations and $\vec{r_1}$, $\vec{r_2}$ are random vectors in the range [0, 1].

In GWO, the first three best solutions achieved are stored so far and enforce the omega agents to update their positions according to the position of the best search agents:

$$\vec{D}_\alpha = \left|\vec{C}_1.\vec{X}_\alpha - \vec{X}\right|, \vec{D}_\beta = \left|\vec{C}_2.\vec{X}_\beta - \vec{X}\right|, \vec{D}_\delta = \left|\vec{C}_3.\vec{X}_\delta - \vec{X}\right| \quad (20)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1.(\vec{D}_\alpha), \vec{X}_2 = \vec{X}_\beta - \vec{A}_2.(\vec{D}_\beta), \vec{X}_3 = \vec{X}_\delta - \vec{A}_3.(\vec{D}_\delta) \quad (21)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1(t) + \vec{X}_2(t) + \vec{X}_3(t)}{3} \quad (22)$$

The final solution would be in a stochastic solution within a circle which is described by the positions of alpha, beta, and delta in the search space. In other words, alpha, beta, and delta estimate the prey position and other wolves update their positions randomly around the victim. Pseudo code of the GWO is shown the fig.4.

```
Initialize the grey wolf population Xi (i = 1, 2, ..., n)
Initialize a, A, and C
Calculate the fitness of each search agent
Xα=the best search agent
Xβ=the second best search agent
Xδ=the third best search agent
while (t < Max number of iterations)
    for each search agent
            Update the position of the current search agent
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update Xα, Xβ, and Xδ
    t=t+1
end while
return Xα
```

Fig. 4. Pseudo code of the GWO algorithm [25].

4.3. Linear Quadratic Regulator

Consider the closed-loop input-output transfer function represented by the following state equation:

$$\dot{x} = A\underline{x} + B\underline{u}; \quad (23)$$
$$y = C\underline{x} + D\underline{u}$$

With $x(t) \in R^N$.

The terms *A* and *B* indicate the constant system model parameters. The pair (A, B) is assumed to be controllable.
Suppose that we have sensors to measure the entire state and that we use a controller (regulator):

$$\underline{u} = -K\underline{x} \tag{24}$$

where *K* is state-feedback matrix. The linear quadratic cost function is defined as [22]:

$$J = \int_{t_0}^{t_f} \left(\frac{1}{2}x^T Qx + \frac{1}{2}u^T Ru\right)dt \tag{25}$$

with the control law as eq.26.

$$u = -R^{-1}B^T Px \tag{26}$$

which can be achieved by solving the following algebraic Riccati equation [21]:

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \tag{27}$$

The block diagram of the system with LQR is shown in Fig.5.

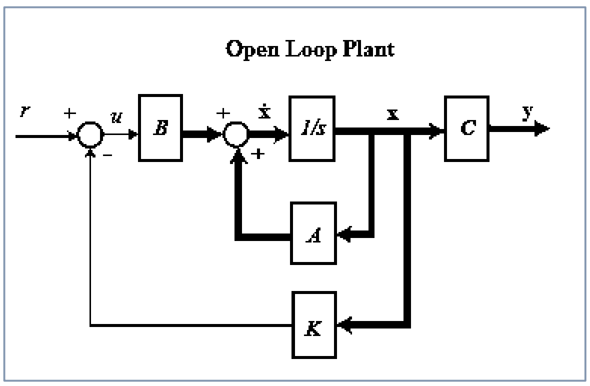

Fig. 5. Block diagram of the LQR controller applied to the system

If the system is controllable (or even just stabilize) and Q is positive-semi definite and R is positive definite, the LQR will result in a stable system with superior gain margin infinite and inferior gain margin of at least (or at most) 0.5 and a phase margin of at least 60 degrees.

5. SIMULATIONS AND RESULTS

Choosing a proper fitness function to get the desired performance aspects like: settling time; overshoot and rise time is important [26]. In this study, we consider a new control technique based on LQR. The main purpose for the proposed fitness is to minimize the following fitness function:

$$J = \int_{t_0}^{t_f} (x^T Qx + u^T Ru)dt + PI \tag{28}$$

and:

$$PI = (10^4 \times OS^2) + (10^4 \times US^2) + (0.0001 \times t_s^{\,2}) \tag{29}$$

Where, Overshoot (OS), Undershoot (US) and settling time is considered for evaluation of the performance index. The proposed function optimized the LQR and it also makes the overshoot decreases, by keeping the settling time in a sensible value.

The fitness function includes two parts (integrals): in the first part $Q = C'\times C$ and the object is to find an optimal value for the term R by utilizing the optimization algorithm. generally, the main task of an LQR controller is to settings of a (regulating) controller governing either a machine or process are found by using a mathematical algorithm that minimizes a fitness function with weighting factors supplied by a human or automatically. In effect, LQR finds those controller settings that minimize the undesired deviations, like deviations from desired altitude or process temperature. In the second part, S is the Overshoot (OS) and Undershoot (US) and t defines the settling time for evaluation of the fitness function. Since, the proposed function makes the overshoot decreases, by keeping the settling time in a sensible value.

GWO is employed for optimizing the proposed fitness function and after that a comparison between GWO and PSO algorithms showed the GWO excellence toward the others. Note that we have 2 constraints: *Q* is positive semi-definite matrix and *R* is positive definite.

Parameters used in GA, PSO and GAPSO algorithms for optimizing the integral controller are presented in the Table 2.

Table 2. GWO and PSO Parameters

Parameters	Value
Number of Search Agent	30
Maximum Iteration	50

By considering the explained restrictions, we can now test how the manipulator will be controlled by the proposed controller. From the Fig.6, it is clear that the GWO has less negative value rather than the PSO. By focusing the GWO based controller in the Fig.7, the final result shows the GWO convergence. Table 2 shows the value for both applied of optimization algorithm on the LQR system. We notice that the settling time is for 2% of final value.

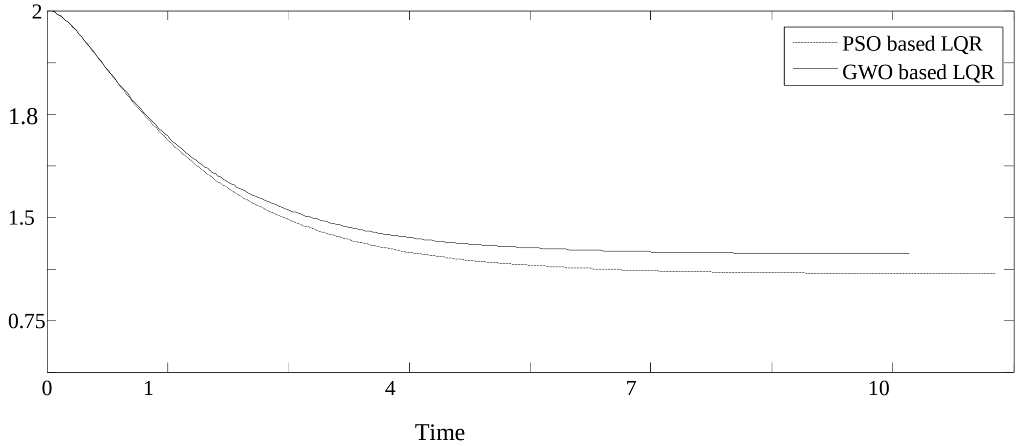

Fig.6. Step response of the single link manipulator robot by PSO (dashed) and GWO (line)

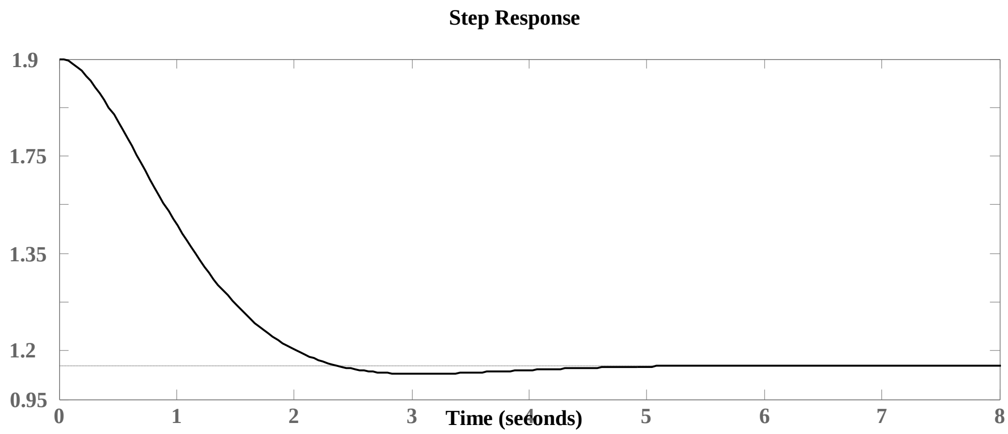

Fig.7. Short step response for the GWO based LQR controller

Table 3. Parameter and cost function value for GWO and PSO based LQR utilized in manipulator

LQR Optimization Method	Value for term 'R'	Min value for Fitness Function
GWO	0.01	0.5
PSO	0.15	1

6. CONCLUSIONS

Selecting inappropriate parameters for the LQR controller will cause an unwanted behavior which may make the system not to be optimal. Since, optimizing the described parameters can develop the security level of the system stability. In this paper, a new optimization technique is used for selection of weighting the regulator to control the single link flexible joint manipulator. The Lagrange technique is utilized to extract the rigid flexible link set up equations of motion, considering the device was a simple spring mass model. The cost of control for the optimized LQR design has been shown to be dependent on the process overshoot, undershoot and settling time.

Therefore, the proposed controller exhibits a more optimal response than a classical LQR controller.
This controller can be performed by means of computer-aided design tools such as MATLAB.

The comparison of performance of the proposed method with a PSO based LQR is done and the results show the presented controller's superiority.

REFERENCES

[1] Rivin, Eugene I. **Mechanical design of robots.** McGraw-Hill, Inc., 1987.

[2] Kelly, Rafael, Victor Santibáñez Davila, and Julio Antonio Loría Perez. **Control of robot manipulators in joint space**. Springer Science & Business Media, 2006.

[3] Eppinger, Steven D., and Warren P. Seering. "**Three dynamic problems in robot force control,"** *in Proc. IEEE International Conference on Robotics and Automation, 1989.*

[4] Ortega, Romeo, Rafael Kelly, and Antonio Loría, **"A class of output feedback globally stabilizing controllers for flexible joints robots."** *IEEE Trans. robotics and automation.* Vol. 11, no. 5, pp. 766-770, 1995.

[5] Tomei, Patrizio. **"A simple PD controller for robots with elastic joints."** *IEEE Trans. Automatic Control*, vol. 36, no. 10, pp. 1208-1213, 1991.

[6] Ulrich, S. and Sasiadek, J.Z., **"Modeling and direct adaptive control of a flexible-joint manipulator",** *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 1, pp. 25–39, 2012.

[7] Ailon, A. and Ortega, R., **"An observer-based set-point controller for robot manipulators with flexible joints,"** *Systems and Control Letters*, vol. 21, pp. 329–335, 1993.

[8] Aldu-Schäffer, A., Ott, C. and Hirzinger, G., **"A unified passivity-based framework for position, torque and impedance control of flexible joint robots,"** *The International Journal of Robotics Research*, vol. 26, no. 1, pp. 23–39, 2007.

[9] Ozgoli, S., and H. D. Taghirad, **"A survey on the control of flexible joint robots,"** *Asian Journal of Control*, vol. 8, no.4, pp. 332-344, 2006.

[10] Wang, W.S. and C.H. Liu, **"Implementation of H2 Optimal Controller for a Single Link FJR,"** *in Proc. 1990 IEEE Int. Conf. Rob. Autom.*, Site??, pp. ??, 1990.

[11] Lin, L.C. and K. Yuan, **"Control of FJRs via External Linearization Approach,"** *J. Rob. Syst.*, 1990.

[12] Berger, R.M. and H.A. ElMaraghy, **"Feedback Linearization Control of Flexible Joint Robots,"** *Rob. Comput. Integr. Manuf.*, vol. 9 (3), pp. 239-246, 1992.

[13] Huang, A.C. and Y.C. Chen, **"Adaptive Sliding Control for Single-Link Flexible-Joint Robot with Mismatched Uncertainties,"** *IEEE Trans. Contr. Syst. Technol.*, vol. 12, no. 5, pp. 770-776, 2004.

[14] Taghirad, H.D. and Gh. Bakhshi, **"Composite Hinf Controller Sythesis for FJRs,"** *In Proc. 2002 IEEE Int. Conf. Intell. Rob. Syst., IROS'02*, pp. 2073-2078.

[15] Ozgoli, S. and H.D. Taghirad, **"Robust Controller for Flexible Joint Robot Using H2/H∞ and Mixed Sensitivity Approaches,"** *Iranian Int. Seminar Mech. Eng.*, Isfahan, Iran,2005.

[16] Drapeau V., Wang D., **"Verification of a closed-loop shaped-input controller for a five-bar-linkage manipulator,"** *in Proc. 1993 IEEE International Conference on Robotics and Automation, pp. 216-221.*

[17] Akyuz IH, Yolacan E, Ertunc HM, Bingul Z. **"PID and state feedback control of a single-link flexible joint robot manipulator,"** *in Proc. 2011 IEEE International Conference Mechatronics* (ICM), pp. 409-414.

[18] Issai Schur. **Neue Begründung der Theorie der Gruppencharaktere.** *Neue begründung der theorie der gruppencharaktere*, pp. 406-432.

[19] Golub, G. H. and Van Loan, C. F. **Matrix Computations.** *3rd ed.*, Baltimore, MD: Johns Hopkins University Press, 1996, pp. 312-314.

[20] Schur, I. **"On the Characteristic Roots of a Linear Substitution with an Application to the Theory of Integral Equations,"** *Math. Ann.*, vol. 66, pp. 488-510, 1909.

[21] Razmjooy, Navid, Ali Madadi, Hamid-Reza Alikhani, and Mahmood Mohseni. **"Comparison of LQR and Pole Placement Design Controllers for Controlling the Inverted Pendulum,"** *Journal of World's Electrical Engineering and Technology* vol. 2322 (2014): p.5114.

[22] Kobayashi, Iwase, Suzuki, Furuta. **Swinging-up and Balancing Control of Double Fututa Pendulum by Using State-Dependent Riccati Equation.** *Graduate School of Systems Engineering***,** Tokyo Denki University.

[23] Razmjooy, Navid, Mohsen Khalilpour, and Mehdi Ramezani. **"A New Meta-Heuristic Optimization Algorithm Inspired by FIFA World Cup Competitions: Theory and Its Application in PID Designing for AVR System,"** *Journal of Control, Automation and Electrical Systems*, vol.27, no.4, pp. 349-472, 2016.

[24] Kennedy J. and Eberhart R. **"Particle swarm optimization,"** *in Proc. 1995 IEEE Int. Conf. Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[25] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. **"Grey wolf optimizer,"** *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.

[26] Zitzler, Eckart, and Simon Künzli. **"Indicator-based selection in multiobjective search,"** *In 2004 International Conference on Parallel Problem Solving from Nature*, pp. 832-842, Springer Berlin Heidelberg.