

The Design of Pole Placement With Integral Controllers for Gryphon Robot Using Three Evolutionary Algorithms

Somayyeh Nalan-Ahmadabad and Sehraneh Ghaemi

Abstract—In this paper, pole placement with integral control for a Gryphon robot are designed using three optimum approaches. The applied approaches are Shuffled Frog Leaping (SFL), Genetic Algorithm (GA) and Firefly algorithms (FA). The design goal is to set up the transient response parameters (overshoot (o.s) and settling time(t_s)) to the desired values and to decrease the volatilities. An objective function of these indexes is defined and minimized by applying SFL, GA and FA. After optimization of the objective function, the optimal parameters for pole placement with integral control are adjusted. Simulation results show that GA has a remarkable effect on decreasing the amount of volatilities and Setting up t_s to the desired value while SFL algorithm performs better on adjusting o.s to its desired value and FA is better on decreasing of the volatilities.

Index Terms—Robotics, pole placement with integral control, gryphon, shuffled frog leaping, genetic algorithm, firefly algorithm.

I. INTRODUCTION

The feedback design method is very prevalent rather than other control methods in nature and technology. Feedback modifies the dynamics of system and decreases the system sensitivity to the model uncertainty in control processes [1], [2]. Pole placement is one of the state space design technique that places all the poles at the predefined desired locations. If plant is type 0 system then we have to use pole placement with integrator to remove error [3].

Major problem of pole placement technique is that it places zeros at the desired locations and the presence of zeros effect on the transient response. Second problem is how to decide best possible locations of poles that satisfies our requirement. For solving these problems, several new methods are recommended. These methods, by having the biological or social inspiration as a common theme, are accomplished within evolutionary algorithms such as Binary and Continuous Genetic algorithms [4], Ant Colony Optimization [5], Continuous and Discrete Particle Swarm Optimization [6], [7] and different types of Honey Bee Colony Algorithms [8], [9].

In this approach, working on Gryphon robot which includes 5 joints, we must design five pole placement with integral controls; each for one joint. Thus we must first verify the pole placement with integral control designation problem as an optimization problem with an objective function which is supposed to be minimized by adjusting the three

performance indexes, i.e. maximum o.s, t_s and the volatilities of step response. Finally we will minimize the defined objective function for each joint by applying SFL, GA and FA.

The paper is organized in six sections. Section II includes a brief description of the system under-investigate. Section III reviews the utilized intelligent algorithms and in Section IV, the problem formulation is explained. Section V presents the simulation results and analysis them and Section VI concludes the paper.

II. A BRIEF DESCRIPTION OF SYSTEM

As shown in Fig. 1, Gryphon is a robot with 5 revolute joints. Thus this robot benefits from 5 degrees of freedom. The first three joints which are called shoulder, elbow and wrist respectively, are supposed to determine the position of the end-effector while the last two joints are responsible for the orientation of the end-effector. This robot is controlled by four micro-processors, one of which is to determine the position of the axis, and two others are the motor controllers and the last one is supposed to support the others and communicates with the host computer. Each joint is moved by a stepper motor while there is the feedback on the corresponding encoder to realize the closed-loop control. In this robot the utilized gear ratio is high enough to assume that all joints are independent from one another.

TABLE I: TRANSFER FUNCTIONS OF GRYPHON JOINTS

JOINT NUMBER	TRANSFER FUNCTION
JOINT 1	$-.0005s + .1012$
JOINT 2	$.0448s^2 + .5581s - .0126$ $-.0031s + .1012$
JOINT 3	$.0221s^2 + .6761s - .0125$ $.0005s + .1013$
JOINT 4	$.0210s^2 + .3968s - .0130$ $-.0022s + .1005$
JOINT 5	$.0096s^2 + 1.4940s - .0051$ $.0276s + .1005$
	$.0539s^2 + 1.4210s - .0059$



Fig. 1. Joint model for Gryphon.

Manuscript received April 10, 2015; revised January 6, 2016.

The authors are with the Control Engineering Department, University of Tabriz, Tabriz, Iran (e-mail: nalansomayyeh@gmail.com, ghaemi@tabrizu.ac.ir).

Table I presents the transfer functions of the joints, which have been obtained in “The Research Lab of Robotics” in the

University of Tabriz.

III. THE UTILIZED INTELLIGENT ALGORITHMS

A. The Shuffled Frog Leaping Algorithm (SFL)

In SFL, the population consists of a set of frogs (solutions) that is partitioned into subsets referred to as memeplexes. The different memeplexes are considered as different cultures of frogs, each performs as a local search. Within each memeplex, the individual frogs hold ideas, that can be influenced by the ideas of other frogs, and evolve through a process of memetic evolution. After a defined number of memetic evolution steps, ideas are passed among memeplexes in a shuffling process [10]. The local search and the shuffling processes continue until the defined convergence criteria is satisfied [11].

For S -dimensional problems (S variables), a frog i is represented as $X_i = (x_{i1}, x_{i2}, \dots, x_{is})$. Afterwards, the frogs are sorted in a scending order according to their fitness. Then, the entire population (P) is divided into m memeplexes, each containing n frogs (i.e. $P = m \times n$). In this process, the first m frogs go to m memeplex respectively, the second m frogs go to m memeplex respectively too and etc.

Within each memeplex, the frogs with the best and worst fitnesses are identified as X_b and X_w , respectively. Also, the frog with the global best fitness is identified as X_g . Then, a process is applied to improve only the frog with the worst fitness (not all frogs) in each cycle. Accordingly, the position of frog with the worst fitness is adjusted as follows:

$$\text{Change in frog position } (D_i) = \text{rand} \times (X_b - X_w) \quad (1)$$

$$\text{New position } X_w = \text{current position } X_w + D_i ; |D_i| \leq D_{\max} \quad (2)$$

where rand is a random number between 0 and 1; and D_{\max} is the maximum allowed change in a frog's position.

If this process produces a better solution, then the worst frog is replaced with better solution. Otherwise, the calculations in Eqs (1) and (2) are repeated but with respect to the global best frog (i.e. X_b is replaced by X_g). If no improvement becomes possible in this case, then a new solution is randomly generated to replace that frog. The calculations then continue for a specific number of iterations [11].

B. Genetic Algorithm (GA)

The genetic algorithms were introduced in 1975 by Holland. It is an optimization method that models the evolution of a population in a particular environment. Each member in population, called chromosome, is a solution for the problem and algorithm tries to improve the quality of solutions. To this end, it uses some operators to act on the chromosomes of population in a predefined number of iterations. These operators are called selection, crossover and mutation.

In this paper we utilize the rollet-wheel selection method. In Fernandes and Rose (2001) some other methods are explained. The crossover operator that is used is Heuristic crossover [12]. For mutation some genes in the population are selected randomly with a predetermined mutation rate. Then these genes are replaced with randomly chosen

variables in the search space.

C. Firefly Algorithm (FA)

Firefly Algorithm was developed by Yang ([13], [14]) and it is based on the flashing characteristics of fireflies. Firefly Algorithm is as follows:

Objective function $f(x)$, $x = (x_1, \dots, x_d)^T$

Initialize a population of fireflies $x_i (i = 1, 2, \dots, n)$

Define light absorption coefficient γ

while ($t < \text{Max Generation}$)

for $i = 1 : n$ all n fireflies

for $j = 1 : i$ all n fireflies

Light intensity I_i at x_i is determined by $f(x_i)$

if ($I_j > I_i$)

Move firefly i towards j in all d dimensions

end if

Attractiveness varies with distance r via $\exp[-\gamma r^2]$

Evaluate new solutions and update light intensity

end for j

end for i

Rank the fireflies and find the current best

end while; Post process results and visualization.

The movement of firefly i is attracted to another more attractive (brighter) firefly j is determined by

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + \alpha \varepsilon_i^t \quad (3)$$

where β_0 is the attractiveness at $r = 0$, the second term is due to the attraction, while the third term is randomization with the vector of random variables ε_i being drawn from a Gaussian distribution. The distance between any two fireflies i and j at x_i and x_j can be Cartesian distance $r_{ij} = \|x_i - x_j\|_2$ or l_2 -norm. For most cases in our implementation, we can take $\beta_0 = 1$, $\alpha \in [0, 1]$, and $\gamma = 1$.

For a nonlinear optimization problem with equality and inequality constraints, a common method is the penalty method. The idea is to define a penalty function so that the constrained problem is transformed into an unconstrained problem. Now we can define

$$\Pi(x, \mu_i, v_j) = f(x) + \sum_{i=1}^M \mu_i \varphi_i^2(x) + \sum_{j=1}^N v_j \psi_j^2(x) \quad (4)$$

where $\mu_i \geq 1$ and $v_j \geq 0$ which should be large enough, depending on the needed quality. As we can see, when an equality constraint is met, its effect or contribution to Π is zero. However, when it is violated, it is penalized heavily as it increases Π significantly.

IV. PROBLEM FORMULATION TO DESIGN POLE PLACEMENT WITH INTEGRAL CONTROL

In pole placement technique, control is achieved by feeding back the state variables through real constant under the restriction that system is completely controllable [15].

Consider a linear dynamic system in the state space form

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \quad (5)$$

If the system is completely controllable we can use pole placement for stabilizing the system or improving its

transient response. We represent control signal u as a linear combination of the state variables, that is

$$u = -Kx \quad (6)$$

where $K = [K_1 \ K_2]$ is state feedback gain matrix. Now closed-loop system, is obtained as follow:

$$\begin{cases} \dot{x} = (A - BK)x + Bu \\ y = Cx \end{cases} \quad (7)$$

Pole placement controller is not able to eliminate error if plant has no integrator (type 0 plants) because it feed only proportional and derivative of state variable. The basic principle of design of type 1 servo system is to insert an integrator in the feed forward path before the plant as shown in Fig. 2. From Fig. 2 we obtain

$$\dot{\varepsilon} = r - y = r - Cx \quad (8)$$

where x is the state vector of plant and r is the reference input. So the new state matrix and u becomes,

$$\hat{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \text{ and } u = -Kx + K_I \varepsilon \quad (9)$$

where ε is output of integrator. New state feedback gain matrix is:

$$\hat{K} = [K \ -K_I] = [K_1 \ K_2 \ -K_I] \quad (10)$$

For all joints, the state feedback gain matrix is used to stabilize the system and also drive toward the performance

criteria $t_s = 2.5, o.s = 15\%$. To this purpose, we do not generate initial population randomly in all algorithms on the contrary first we calculate the dominating complex conjugate poles considering desired t_s and $o.s$.

Then rest of poles -nondominant poles- are calculated using dominating pole concept with adding some randomness based on the order of system. Using all poles and the state space model of system, the feedback gain matrix is calculated. Then all element of this matrix are put for some mutation change. Now using these feedback gains we calculated the specification of system. These specifications are used to calculate the fitness value for that feedback gain matrix. Fitness function is considered as follows:

$$\text{fitness} = \max \left(\left| \frac{\text{maximum o.s} - \text{desired o.s}}{\text{desired o.s} * 0.05} \right|, \left| \frac{\text{measured } t_s - \text{desired } t_s}{\text{desired } t_s * 0.05} \right| \right) +$$

10 * measured amplitude of volatilities

V. SIMULATION RESULTS

For all of algorithms, 30 runs, have been carried out to have the average answer which is more reliable. The parameter evaluation for each utilized algorithm is shown in Table II.

Tables III, IV and V respectively present the simulation results, k_1, k_2 and k_I values of SFL, GA and FA algorithm for all of the five joints. The step responses for each of the five motors are shown on Fig. 3 to Fig. 7.

TABLE II: THE PARAMETER EVALUATION FOR EACH UTILIZED ALGORITHM

SFL	Population size=50	The number of memplexes=10	Memetic evolution steps before shuffling=10
GA	Population size=40	Selection rate= 0.5	Mutation rate=0.15
FA	Population size=40	Light absorption coefficient=1	$\beta_0=2, \alpha=0.2$

TABLE III: SIMULATION RESULTS APPLYING SFL ALGORITHM

Joints	K_1	K_2	K_I	o.s	t_s	volatility
Joint 1	28.8320	62.0782	-4.0778	14.7063%	2.5489	2.9100%
Joint 2	33.4760	76.0274	-21.5544	15.0000%	2.5014	2.0000%
Joint 3	30.9294	69.3706	-10.6973	15.0557%	2.3765	3.2501%
Joint 4	22.0221	42.3040	-150.1186	15.0019%	2.5010	5.8910%
Joint 5	38.7766	67.3095	-109459	14.7285%	2.3384	1.6702%

TABLE IV: SIMULATION RESULTS APPLYING GA ALGORITHM

Joints	K_1	K_2	K_I	o.s	t_s	volatility
Joint 1	31.6282	71.2549	-3.3688	14.9994%	2.5000	2.8305%
Joint 2	25.8431	53.4011	-24.1567	15.0000%	2.5000	2.0000%
Joint 3	34.9450	62.1672	-4.0636	14.0110%	2.6648	2.5801%
Joint 4	27.8382	59.5434	-148.0281	14.5586%	2.4900	1.8811%
Joint 5	39.1412	66.7262	-9.9519	15.0258%	2.4462	1.8804%

TABLE V: SIMULATION RESULTS APPLYING FA ALGORITHM

Joints	K_1	K_2	K_I	o.s	t_s	volatility
Joint 1	29.2948	50.0000	0.5000	13.9619%	2.7709	2.0405%
Joint 2	30.1211	68.3904	-23.6012	15.5775%	2.3131	4.2616%
Joint 3	33.7807	73.0614	-9.8013	12.4576%	2.2838	2.0500%
Joint 4	-148.1210	25.3615	50.2684	14.2662%	2.6669	2.9517%
Joint 5	29.4992	48.7147	-13.8040	15.4754%	2.4601	1.9825%

TABLE VI: THE AVERAGE VALUES FOR COMPARISON

	Average maximum o.s – desired o.s	Average measured t_s – desired t_s	Average volatility
SFL	0.1246	0.0673	3.1443
GA	0.2914	0.0457	2.2344
FA	1.0734	0.1762	2.6573

The simulation results show that GA has a remarkable effect on decreasing the amount of the volatility and Setting up t_s to the desired value because considering the average value of the 5 joints according to Tables III to V, it is obviously seen in Table VI that the average absolute difference between the t_s and its desired value and the average volatility is far less in GA in comparison to FA and SFL. While the performance of SFL on decreasing the amount of absolute difference between o.s and its desired value is better than two other algorithms. Also the SFL algorithm compared to FA performs better on controlling t_s and the FA algorithm is better on decreasing of the volatility.

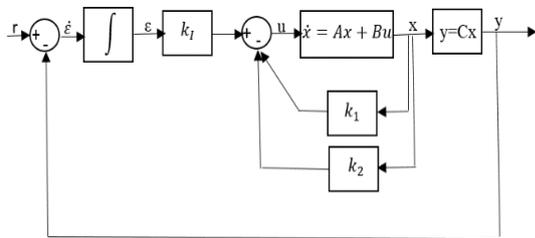


Fig. 2. Block diagram of pole placement controller with integrator.

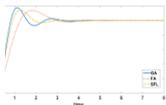


Fig. 3. Step response applied on motor 1.

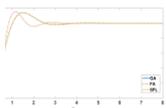


Fig. 4. Step response applied on motor 2.

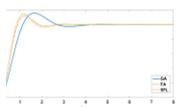


Fig. 5. Step response applied on motor 3.

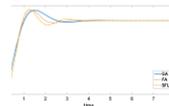


Fig. 6. Step response applied on motor 4.

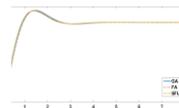


Fig. 7. Step response applied on motor 5.

VI. CONCLUSION, DISCUSSION AND FUTURE WORKS

This research begins by introducing the Gryphon robot as case-study. Then pole placement with integral control were designed for the joints of a gryphon robot using three evolutionary algorithms, i.e. the SFL, GA and FA. Which, unfortunately, is not fast enough to be utilized in on-line applications. The results of applied simulations are reported in the included tables and figures.

Comparing the contents of tables and figures leads to introducing the GA strong enough in setting up of t_s and reduction of the volatility. On the other hand the SFL algorithm acts better on adjusting the o.s. Also the act of FA algorithm is good in decreasing of the volatility.

Designing pole placement with integral controls with other intelligent algorithms and changing the present controller to an on-line one may be our future field study.

REFERENCES

- [1] C. J. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*, New York: Macmillan Publishing Company, 1992, pp. 52-218.
- [2] T. M. Chen, "A pole-placement design for linear time-invariant siso feedback systems," *Journal of Agricultural Machinery*, vol. 12, pp. 25-36, March 2003.
- [3] K. Ogata, *Modern Control Engineering*, 5th ed. PHI Learning Private Ltd, Rimjhim House, NJ, 1997.
- [4] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, John Wiley, 2004.
- [5] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, 2004.
- [6] R. C. Eberhart and Y. Shi, "Particle swarm optimization: Developments, applications and resources," IEEE, 2001.
- [7] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE International Conference on*

Systems, Man and Cybernetics, 'Computational Cybernetics and Simulation', vol. 5, pp.4104–4108, 1997.

- [8] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization:artificial bee colony," *Journal of Global Optimization*, vol. 39, issue 3, pp. 459–471, 2007.
- [9] D. Karoboga and B. Basturk, "On the performance of artificial bee colony (abc) algorithm," *Applied Soft Computing*, vol. 8, issue 1, pp. 687–697, 2008.
- [10] S. Y. Liong, "Atiquzzaman Md. optimal design of water distribution network using shuffled complex evolution," *J Inst Eng*, vol. 44, pp. 93–107, 2004.
- [11] M. M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *J Water Resour Plan Manage*, vol. 129, no. 3, pp. 210–25, 2003.
- [12] Global Optimization Toolbox. [Online]. Available: <http://www.mathworks.com/help/toolbox/gads/f6174dfi10.html>
- [13] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, UK, 2008.
- [14] X. S. Yang, "Firefly algorithms for multimodal optimization," *Lecture Notes in Computer Science*, vol. 5792, pp. 169–178, Springer-Verlag, Berlin, 2009.
- [15] M. S. Saad, H. Jamaluddin, and I. Z. M. Darus, "Implementation of pid controller tuning using differential evolution and genetic algorithms,"

International Journal of Innovative Computing, Information and Control, vol. 8, no. 11, November 20.



algorithms.

S. Nalan-Ahmadabad received her B.Sc. and M.Sc. degrees in electrical engineering (control systems) from University of Tabriz, Tabriz, Iran, in 2007 and 2010, respectively. Now, she is a Ph.D. student in electrical engineering (control systems) in University of Tabriz, Tabriz, Iran. Her research interests include robotics, fuzzy choquet integral, neural networks and intelligent



research interests include soft computing, genetic algorithm and human behavior modeling.

Sehraneh Ghaemi received her B.Sc. degree in power engineering from Amirkabir University of Technology, Tehran, Iran, in 2000 and the M.Sc. and Ph.D. degrees in control engineering from University of Tabriz, Tabriz, Iran, in 2003 and 2010, respectively. She is now an assistant professor in the Faculty of Electrical and Computer Engineering, University of Tabriz. Her research interests include soft computing, genetic algorithm and human behavior modeling.