International Journal of u- and e- Service, Science and Technology
Vol.8, No. 6 (2015), pp.305-312
http://dx.doi.org/10.14257/ijunesst.2015.8.6.30

Pole Placement Using Genetic Algorithm with Integral Control

Abhishek Pratap Singh and Ashish Patra

Electrical Engineering Department
Madhav Institute of Technology and Science
Gwalior, India
abhishek.pratap28@gmail.com, prof_patra@rediffmail.com

Abstract

This paper presents the application of genetic algorithms to find the optimum feedback gains in pole placement with integral control to get the desired performance in control system. Performances are measured in terms of key parameters like settling time, peak overshoot, undershoot, rise time etc in control system. The major drawbacks of pole placement technique are to get optimum location of poles and we cannot place zeros at desired locations. Dominating concept of poles is enough for placing poles for systems which transfer functions are without zeros. But if there are zeros in transfer function then it is not enough to place pole with dominating pole concept because presence of zero can destroy the dynamic response of control system. In this paper genetic algorithm is used to find the optimum locations of poles to get the desired response and compensate the effect of zeros on transient in case zeros are present in transfer function. GA is a population based algorithm to find the global solution of a problem. This paper compares the unit step response of a plant with presence of zero in transfer function. The same GA based technique is used for finding optimum location of poles to get the desired response for magnetic levitation system. This analysis is well supported by the simulation and experimental results done using MATLAB and Simulink.

Keywords: *Dominating Poles concept, Genetic algorithm, Pole Placement*

1. Introduction

Pole placement by output feedback is separated into pole placement by state feedback and observer pole placement. Since both problems are dual, only the state feedback case is worked out in details. In the single-input case, the pole placement problem has a unique solution. In multi-input case, the solution to the pole placement problem is non-unique. Therefore other specifications in addition to the pole placement can be satisfied. A majority of design technique in control theory is based on the state feedback configuration. Classical method only places the dominant pair of poles to desired location assuming that higher order poles do not affect the second order approximation. Second drawback of classical design is that if parameters of plant change we have to redesign the controller to satisfy our requirements [7].

To overcome these drawbacks it is better to use state space design techniques. Pole Placement also known as full state feedback is one of the state space design technique that places all the poles at predefined desired locations. If plant is type 0 system papers then we have to use pole placement with integrator to remove error [4].

Major problem of Pole Placement technique it does place zeros at desired locations and presence of the zeros does affect the transient response. Second problem is how to decide best possible locations of poles that satisfy our requirement. A general method is dominant pole concept in this method we choose the dominating complex conjugate pole based on desired overshoot and settling time and rest of the poles should be placed five to ten times away from imaginary axis with respect to real part of complex conjugate poles. But this

ISSN: 2005-4246 IJUNESST

method does not consider the effect of zero's on transient response of the plant. To overcome this drawback GA is used to find the best possible location of poles to satisfy our requirement. In control system genetic algorithm consider the zero's effect on response and find the solution as better as possible [5-6].

This paper is only concerned with closed loop system and is targeted to find the amount of internal feedback that will bring the actual performance closer to the intended performance of a system. To achieve this proper combination of feedback, it is necessary to model the system in state space. State space model monitor the internal variables and incorporate a method of feedback into the system.

2. Pole Placement Controlling Using Integrator

In pole placement technique despite using controllers with fixed configuration in the forward path or feedback path, control is achieved by feeding back the state variables through real constant under the restriction that system is completely controllable [2].

Consider a linear dynamic system in the state space form

$$\dot{x} = Ax + Bu \tag{1}$$

$$y = Cx \tag{2}$$

If the system is completely controllable we can use pole placement for stabilizing the system or improving its transient response. Here we represent control signal u as a linear combination of the state variables, that is

$$u = -Kx \tag{3}$$

Where K = [K1 K2 K3 K4] state feedback gain matrix. Now closed-loop system, given by

$$\dot{x} = (A-BK)x \tag{4}$$

$$y = Cx \tag{5}$$

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 we want to control. The basic principle of design of type 1 servo system is to insert an integrator in the feed forward path between the error comparator and the plant as shown in Figure 1.

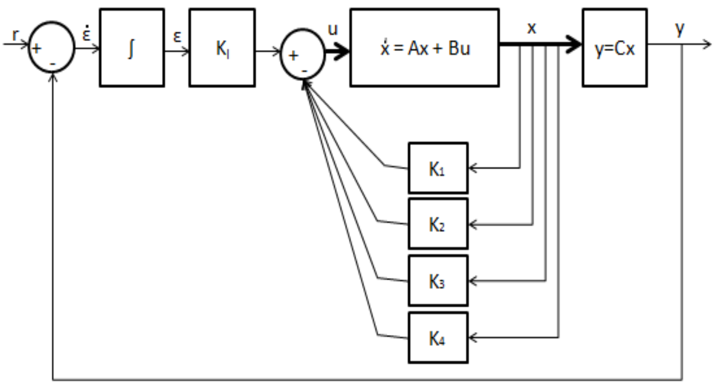

Figure 1. Block Diagram of Pole Placement Controller with Integrator

From Figure 6 we obtain

$$\acute{\varepsilon} = r - y = r - Cx \tag{6}$$

Where x = state vector of the plant and r is reference input. Using the new state variable, new state matrix of the plant is given as,

$$\mathbf{A}_{\mathbf{hat}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{0} \end{bmatrix} \text{ and } \mathbf{B}_{\mathbf{hat}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}$$

Now u becomes

$$u = Kx + K1\acute{\varepsilon} \tag{7}$$

and new state feedback gain matrix

$$Khat = [K - K1] \tag{8}$$

3. Genetic Algorithm

The use of natural evolution method for the optimization of control system has been of interest for the researchers since along time. The control system parameters are considered as the genes of one chromosome and a random population is generated from some of this chromosomes. Then the object function of control system using each chromosome parents is calculated, and then based on population upgrading methods such as roulette wheel mechanism the best chromosome with optimum objective function is generated. Figure 2 shows the flow chart of GA to search the feedback matrix gain which satisfies our requirement. In this algorithm first we create the initial population and check that our requirement is satisfied or not. In case it is not satisfied every time a new population is created by using few percent of champions and rest is from crossover of parents.

In Figure 3 a detailed flow chart is given to generate the initial population for pole placement. First we calculate the dominating complex conjugate poles and rest of the poles are calculated using dominating pole concept with adding some randomness based on order of the system. Using these poles and state space model of system, feedback gain matrix is calculated using either acker or place function of MATLAB. Then all element of this matrix is put for some mutation change.

Now using these feedback gains we transformed the system and using the “step info” function of MATLAB we calculated the specification of system. These specifications are used to calculate the fitness value for that feedback gain matrix. Mutation method describes how and how often new data is injected.

Fitness calculation is based on match method. The calculation is designed to find a K vector that will drive the state space model to within 5% of both user provided criteria. In order to match both criteria, a score is given to each (overshoot and settling time) and the maximum between those is given to the genome.

Genetic algorithm (GA) is a search algorithm that explores the search space in a manner analogous to evolution in nature [1]. It uses probabilistic rules to search for and change the potential solutions in the search space, using a cost function to analyze the fitness of solutions. GA requires the solution to be represented in a way that is analogous to genes so that the processes that bring about a change in the genes (like mutation) can be used. Usually this is done by representing the solutions in a binary format. Initialization, firstly initial solutions are randomly selected from the search space. Selection, during each iteration, a proportion of solutions is selected, based on the fitness function (fitter solutions are more likely to get selected), for breeding the next generation of solutions. The selection is done in a probabilistic manner.

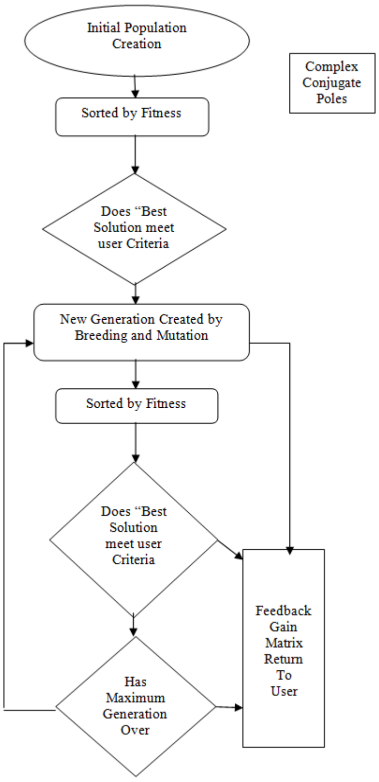
Initial Population Creation
Complex Conjugate Poles
Sorted by Fitness
Does "Best Solution meet user Criteria
New Generation Created by Breeding and Mutation
Sorted by Fitness
Does "Best Solution meet user Criteria
Feedback Gain Matrix Return To User
Has Maximum Generation Over

Figure 2. Flow Chart of GA for Pole Placement with Integral Controller

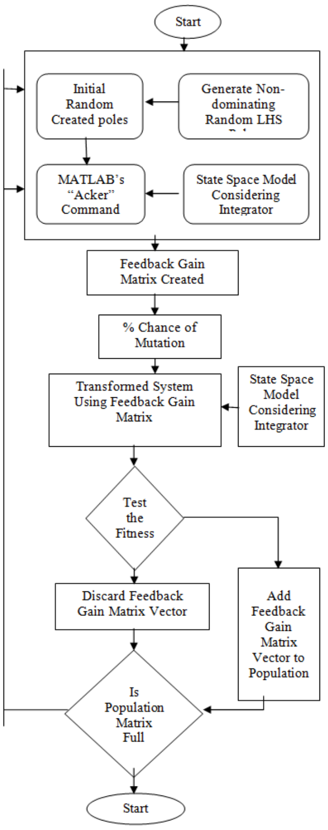
Start
Initial Random Created poles
Generate Non-dominating Random LHS
MATLAB's "Acker" Command
State Space Model Considering Integrator
Feedback Gain Matrix Created
% Chance of Mutation
Transformed System Using Feedback Gain Matrix
State Space Model Considering Integrator
Test the Fitness
Discard Feedback Gain Matrix Vector
Add Feedback Gain Matrix Vector to Population
Is Population Matrix Full
Start

Figure 3. Flow Chart of GA for Initial Population Creation

Fitness calculation is based on match method. The calculation is designed to find a K vector that will drive the state space model to within 5% of both user provided criteria. In order to match both criteria, a score is given to each (overshoot and settling time) and the maximum between those is given to the genome.

$$fitness =$$

$$max\left(\left|\frac{measured\ overshoot - desired\ overshoot}{desired\ overshoot * 0.05}\right| , \left|\frac{measured\ settling\ time - desired\ settling\ time}{desired\ settling\ time * 0.05}\right|\right)$$

$$+10 * measured\ undershoot$$

If fitness value is feasible, feedback gain matrix is added to population along with fitness value otherwise discarded. Once the initial population is created it is sorted best on fitness value in increasing order. If the lowest fitness value satisfy user criteria the process is stop otherwise new generation is created again [3].

4. Simulation and Results

A majority of design technique in control theory is based on the state feedback configuration. For simulation and verification purpose, a type 0 unstable plant with zero used is

$$G(s) = \frac{3s + 4}{s^2 + s - 0.010}$$

The state feedback gain matrix is used to stabilize the system and also drive toward the performance criteria 2.8 second's settling time and 14% overshoot. To get the above performance we calculated the location of poles at -1.392 ± 1.6632i and 1.82 using dominant pole concept which gives the state feedback gain K = [-2.6973 4.2972 13.1895] using either acker or place function of MATLAB. Based on this feedback gain matrix, step response and specifications of system is shown in Figure 4 and given in Table I.

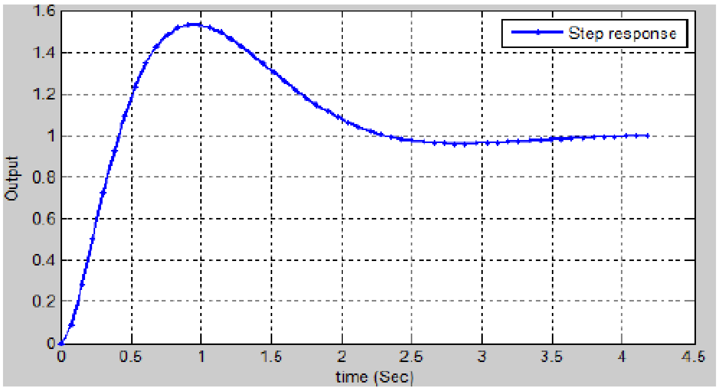

Figure 4. Step Response of System based on Dominant Pole Concept

Table 1. Step Response Parameters of System based on Dominant Pole

Step Response	Rise time	Peak overshoot	Settling time	Under-shoot
	0.27	52.18	3.22	0

This shows that direct calculation does not yield the desired result. The system has two poles, but it also has one zero at S = -1 which is not considered during the calculation of state feedback gain in pole placement.

Now to get the desired response we have to consider the effect of zero on transient response and have to replace the poles at different location. To avoid the effort in hit and trial and get the as better as possible location of poles we use GA. Using GA for performance criteria 2.8 second's settling time and 14% overshoot location of poles -1.82 ± 0.59i and -8.81 is calculated which gives the feedback gain K = [-0.9903 4.1468 -10.7318] Based on this feedback gain matrix, step response and specifications of system is shown in Figure 5 and given in Table II.

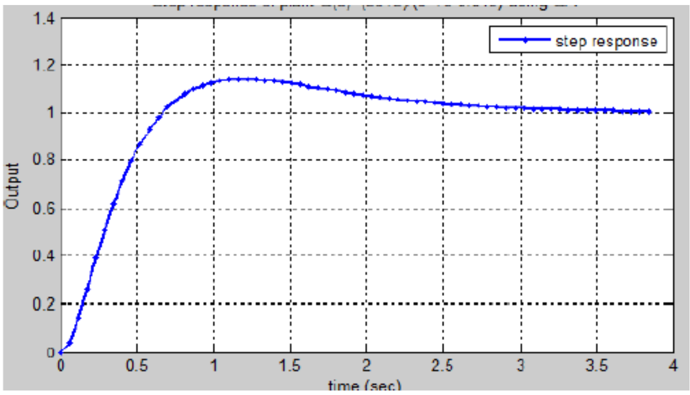

Figure 5. Step Response of System based on GA

Table 2. Step Response Parameters of System based on Genetic Algorithm

Step Response	*Rise time*	*Peak overshoot*	*Settling time*	*Under-shoot*
	0.423	12.256	2.725	0

From the response it is clear that state feedback gain matrix generated by genetic algorithm is better match for desired performance criteria. Though zero is still present but new position of poles has compensated for the zero, and the performance criterion is met. The same GA tuned pole placement technique is used to verify the performance of magnetic levitation system in laboratory which has the following transfer function.

$$G(s) = \frac{2500}{s^2 - 889}$$

It is a type 0 unstable plant so we have to use pole placement with integrator to remove error. For performance criteria 0.2 second's settling time and 10% overshoot location of poles -19.67 ± 26.19i and -129.75 is calculated which gives the feedback gain K = [-2.865 0.0677 -55.718]. Based on this feedback gain matrix, step response and specifications of system is shown in Fig. 6 and given in Table III.

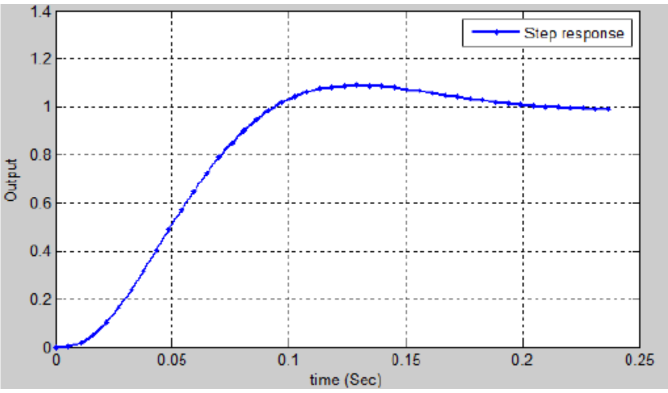

Figure 6. Step Response GA Tuned Magnetic Levitation

Table 3. Step Response Parameters of Magnetic Levitation based on ga

Step Response	*Rise time*	*Peak overshoot*	*Settling time*	*Under-shoot*
	0.48	9.8423	0.9320	0

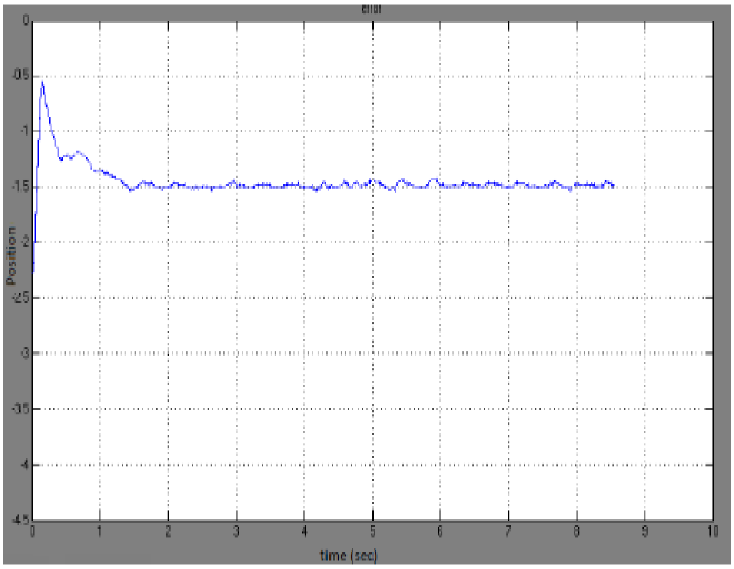

Figure 7. Experimental Response of Magnetic Levitation using GA Tuned Pole Placement with Integral Control

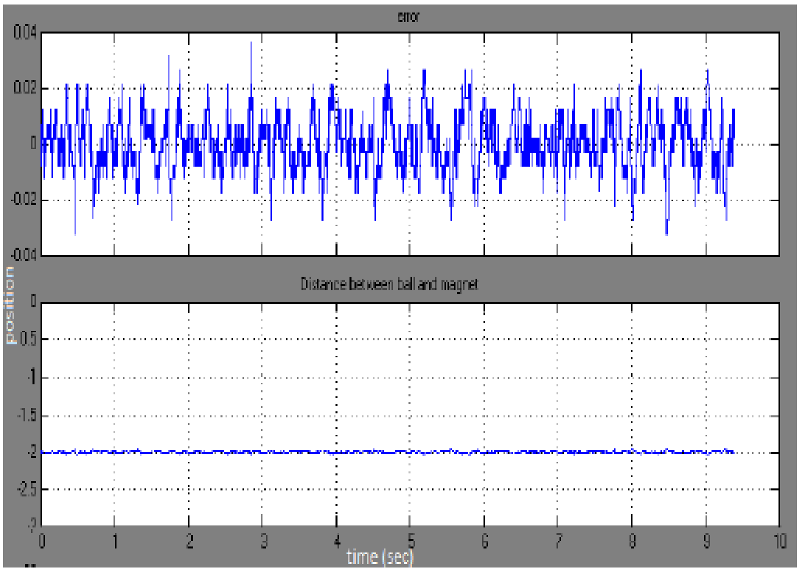

Figure 8. Distance of Ball and Error during Steady State

The same feedback gain matrix is used to check real time performance on googoltech made magnetic levitation system. From experimental results shown in Figure 7 and Figure 8, it verified that the desired performance criteria are satisfied and error is within 2% band.

5. Conclusion

In this paper a comparative study has been performed between GA tuned and dominant pole based pole placement. Results show that the performance of GA tuned pole placement with integral controller is much better in comparison to the conventional approaches. So GA is considered as a powerful tuning technique to find optimum location of poles for state feedback controllers [8].

The main advantage of using GA is that it is considers the effect of zero in the response and place the poles at locations which compensate the effects of zeros to get the desired response. So the main drawback of pole placement that it is unable to place zeros overcome by using GA. The result shown in Figure 5 verified that GA tuned controllers give the desired performances even if zeros are present in transfer function of the plant. The same GA approach can also be used for tuning of other type of controllers parameters like PID controllers.

Acknowledgments

Authors would like to thanks to Dr. Sanjeev Jain, Director, MITS, Gwalior, MP, for promoting this work.

References

[1] M. Salami and G. Cain, "An Adaptive PID Controller Based on Genetic Algorithm Processor," IEE Genetic Algorithms in Engineering Systems, Innovations and Applications Conference, Publication No. 414, 12-14 September, 1995.

[2] Mohd Sazli Saad, Hishamuddin Jamaluddin and Intan Zaurah Mat Darus, " Implementation of PID Controller tunning using Differential Evolution and Genetic Algorithms", International Journal of Innovative Computing, Information and Control, Volume 8, Number 11, November 2012.

[3] Y. J. Cao and Q. H. Wu, "Teaching Genetic Algorithm Using MATLAB", Int. J. Elect. Eng. Educ., Vol. 36, pp. 139–153. Manchester U.P., 1999. Printed in Great Britain.

[4] K. Ogata, Modern Control Engineering, PHI Learning Private Lltd, Rimjhim House, NJ, 1997, 5th Ed., pp. 70–72 and pp. 739–950.

[5] S. N. Sivanandam and S. N. Deepa, "Introduction to Genetic Algorithms", Springer, ISBN 978-3-540-73189-4, SPIN: 12053230.

[6] P. J. Fleming and R. C. Purshouse, "Genetic Algorithm in Control Systems Engineering", CONTROL SYSTEM, ROBOTS AND AUTOMATION- Vol. XVII- *Genetic Algorithms in Control System Engineering.*

[7] M. Gopal, Digital Control and State Variable Method, Tata McGraw Hill, NJ, 2011, 3th Ed., pp.393.

[8] Vishwa Nath, R.Mitra "Genetic Algorithm Tuned Pole Placement with Integral Control" International Journal of Scientific Research Engineering & Technology (IJSRET), ISSN 2278 – 0882 Volume 3 Issue 2, May 2014.