# APPLICATION OF GENETIC ALGORITHM TO DESIGN PID CONTROLLER FOR POWER SYSTEM STABILIZATION

Seyyed Hossein Hosseini[1] Reza Rahnavard[1, 2] Hamed Kharrati[1]
hosseini@tabrizu.ac.ir rerahnavard@gmail.com hkharrati.eec@gmail.com
[1]*Center of Excellence for Mechatronics, University of Tabriz ,Tabriz,Iran*
[2]*Azarbyjan Regional Electric Power Company, Tabriz, Iran*

*Key words: Power system stabilization-LQR (linear quadratic regulator) - Genetic Algorithm-Fuzzy*

## ABSTRACT

**This paper focuses on the use of advanced techniques in genetic algorithm for solving power system stabilization control problems. Dynamic stability analysis of power system is investigated considering Proportional-Integral-Derivative power system stabilizer for modern power systems. Gain settings of PID-PSS are optimized by minimizing an objective function using genetic algorithm (GA). Dynamic responses are also compared considering PID-PSS and FUZZY-PSS and LQR (Linear Quadratic Regulator)-PSS. Analysis reveals that the proposed PSS gives better dynamic performances as compared to that all of mentioned methods. Controller design will be tested on the power system to prove its effectiveness. All simulations will be carried out using MATLAB® based package for nonlinear simulations of power systems, Power Analysis Toolbox (PAT).**

## I. INTRODUCTION

Power system stabilizers (PSS) must be capable of providing appropriate stabilization signals over a broad range of operating conditions and disturbances. With the increasing electric power demand and need to operate power system in a faster and more flexible manner in the deregulated competitive environment, recent power systems can reach stressed conditions more easily than the past. These cause unstable or poorly damped oscillations that have been observed more often in power systems around the world. In recent years, stabilizing control schemes using intelligent procedures have been proposed. The reason for the lack of stability analysis is due to the complexity of the power systems. Moreover, industry will be reluctant to accept controller design if stability cannot be guaranteed .Consequently, it maybe very difficult to adjust the parameters of the PID controllers via analytical methods. To overcome this problem, using intelligent methods is proposed. Therefore, serious consideration is now being given on the issue of power system stabilization control. A major effort has to be made to improve of power system stabilization. Due to the rapid development of computer technology, the use of optimization tools becomes feasible to help in the implementation of control signals in power system. Genetic algorithm is one of them that have been proved an effective method to solve many difficult problems. This method can be applied directly to various problems, without need to transform them into mathematical formulations [1-2]. High convergence rapid, low computational burden and do not caught in local minima are the illustrious features of this method. To solve power stability problems in modern power systems several modern control strategies have been tested and proposed that seek to track the system conditions close to real time , such as variable structure , self tuning ,artificial intelligence (AI) and fuzzy logic and linear quadratic regulator (LQR) stabilizers. However, practical realization of such power system stabilizers based on modern control theory is very difficult because they need real-time monitoring, measurement or estimation of system variables. In this paper, a classical speed input PID-PSS is proposed. Optimum parameter settings of PID-PSS are obtained by minimizing a cost function using GA. The linearzed model of studied power system consisted of synchronous machine connected to infinite bus bar through transmission line is studied and simulation results are presented and compared with FUZZY-PSS and LQR-PSS. [3-5]

## II.STUDIED POWER SYSTEM MODELING

The linearized model of studied power system consisted of synchronous machine connected to infinite bus bar through transmission line is represented in a block diagram as shown in Figure 1. Its state space formulation can be expressed as follows [6]:

$$\Delta\dot{\delta} = \omega_0 \Delta\omega \tag{1}$$

$$\Delta\dot{\omega} = \frac{1}{M}(-K_1\Delta\delta - D\Delta\omega - K_2\Delta E_q^{'}) \tag{2}$$

$$\Delta\dot{E_q^{'}} = \frac{1}{T_{do}^{'}}(-K_4\Delta\delta - \frac{\Delta E_q^{'}}{K_3} + E_{FD}) \tag{3}$$

$$\Delta\dot{E}_{FD} = \frac{1}{T_A}(-K_AK_5\Delta\delta - K_AK_6\Delta E_q^{'} - \Delta E_{FD} + K_Au) \tag{4}$$

In a matrix form as follows:

$$\dot{X}(t) = AX(t) + Bu(t) \tag{5}$$

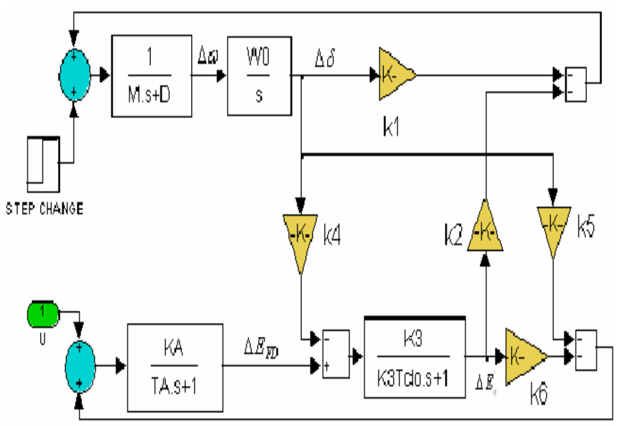


Figure 1. Block diagram of power system under study

Where,

$$A=\begin{bmatrix} 0 & \omega_0 & 0 & 0 \\ \dfrac{-k1}{M} & \dfrac{-D_0}{M} & \dfrac{-k_2}{M} & 0 \\ \dfrac{-k_4}{T'_{d0}} & 0 & \dfrac{-1}{k_3 T'd_0} & \dfrac{1}{T'd_0} \\ \dfrac{-k_A k_5}{T_A} & 0 & \dfrac{-k_A k_6}{T_A} & \dfrac{-1}{T_A} \end{bmatrix} \tag{6}$$

$$B=\begin{bmatrix} 0 \\ 0 \\ 0 \\ \dfrac{k_A}{T_A} \end{bmatrix}, \quad X=\begin{bmatrix} \Delta\delta \\ \Delta\omega \\ \Delta E'_q \\ \Delta E_{FD} \end{bmatrix} \tag{7}$$

## III. GENETIC ALGORITHM

The genetic algorithm (GA) is an optimization and stochastic global search technique based on the principles of genetics and natural selection. A GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the "fitness" (i.e., minimizes the cost function). The method was developed by John Holland (1975) over the course of the 1960s and 1970s and finally popularized by one of his student, David Goldberg (1989) [7-8]. Some of the advantages of a GA are as follows:

- Optimizes with continuous or discrete variables
- Doesn't require derivative information
- Simultaneously searches from a wide sampling of the cost surface
- Deals with the large number of variables
- Optimizes variables with extremely complex cost surfaces (they can jump out of a local minimum)
- Provides a list of optimum variables, not just a single solution
- May encode the variables so that the optimization is done with the encoded variables
- Works with numerically generated data, experimental data, or analytical functions, and
- High convergence rapid.

In the discrete GA, solution point is a binary string of $0$ and $1$ called "chromosome" and number of bits ($N_{bits}$) depends on desired accuracy. The string is included of $n$ variables ($x_1, x_2, \ldots, x_n$), hence the number of bits for each variable is $N_{bits}/n$ called "gene".

A sample solution point with ***8***-bits and two variables (x ,y) is shown in below:

$$\overbrace{1\ \ 0\ \ 1\ \ 1}^{gene1(X)}\ \overbrace{0\ \ 0\ \ 0\ \ 1}^{gene2(Y)}$$

The first **4** bits are related to x and next bits are related to y. To calculate the cost of solution points, they must be decoded at first. Decoded form of the mentioned string is calculated in below:

$$\begin{cases} x \rightarrow 1\times2^3+0+1\times2^1+1\times2^0=11 \\ y \rightarrow 0+0+0+1\times2^0=1 \end{cases} \Rightarrow \quad Cost(x,y)=Cost(11,1)$$

The following sections (A-E) describe GA method and its operators:

***A:*** *The population*

The GA starts with a group of chromosomes known as the population. The population has $N_{pop}$ chromosomes called population size.

***B:*** *Natural selection*

Natural selection is performed on the population by keeping the "most" promising individuals, based on their fitness. In this way, it is possible to keep the size of the population constant, for convenience. First, the $N_{pop}$ costs and associated chromosomes are ranked from lowest cost to highest cost. Then, only the best are selected to continue, while the rest are deleted. The selection rate, $X_{rate}$, is the fraction of $N_{POP}$ that survives for the next step of mating (crossover). The number of chromosomes that are kept each generation is:

$$N_{keep}=X_{rate}\times N_{pop} \tag{8}$$

Natural selection occurs each generation or iteration of the algorithm.

***C:*** *Selection*

In order to replace the deleted chromosomes and keep the population size constant, two chromosomes are selected from the mating pool of $N_{keep}$ chromosomes to produce two new offspring. Pairing takes place in the mating population until $N_{pop}$-$N_{keep}$ offspring are born to replace the discarded chromosomes.

***D*:** *Crossover (Mating)*

Mating is the creation of one or more offspring from the parents selected in the pairing process. The current members of the population limit the genetic makeup of the population. The most common form of mating involves two parents that produce two offspring. A crossover point is randomly selected between the first and the last bits of the parents' chromosomes. First, $parent_1$ passes its binary code to the left of the crossover point to $offspring_1$. In a like manner, $parent_2$ passes its binary code to the left of the same crossover point to $offspring_2$. Next, the binary code to the right of the crossover point of $parent_1$ goes to $offspring_2$ and $parent_2$ passes its code to $offspring_1$ (see Table 1). Consequently, the offspring contain portions of the binary codes of both parents. The parents have produced a total of $N_{pop}$-$N_{keep}$ offspring, so the chromosome population is remained constant ,$N_{pop}$ This method is called Single-Point-Crossover (S.P.C); there is an other type of crossover called Two-Point-Crossover (T.P.C). In T.P.C, two crossover points are randomly selected between the first and the last bits of parents.

Table 1. Process of crossover for two parents

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $parent_1$ | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| $parent_2$ | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| $offspring_1$ | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| $offspring_2$ | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |

***E*:** *Mutation*

Random mutations alter a certain percentage of the bits in the list of chromosomes. Mutation is the second way a GA explore a cost surface. It can introduce traits not in the original population and keeps the GA from converging too fast before sampling the entire cost surface. A single point mutation changes a one to zero, and vice versa. Mutation points are randomly selected from the $N_{pop}\times N_{bits}$ total number of bits in the population matrix ($N_{pop}\times N_{bits}$). Mutations do not occur on the best solutions. They are designed as *elite* solutions destined to propagate unchanged. Such elitism is very common in GAs. The number of bits that must be changed is determined by mutation rate %μ.

$$\text{Mutations}=\mu\times (N_{pop}-1)\times N_{bits} \tag{9}$$

Increasing the number of mutations increases the algorithm's freedom to search outside the current region of variable space. It also tends to distract the algorithm from converging on a popular solution. [9]

Flowchart of the algorithm is shown in figure2.

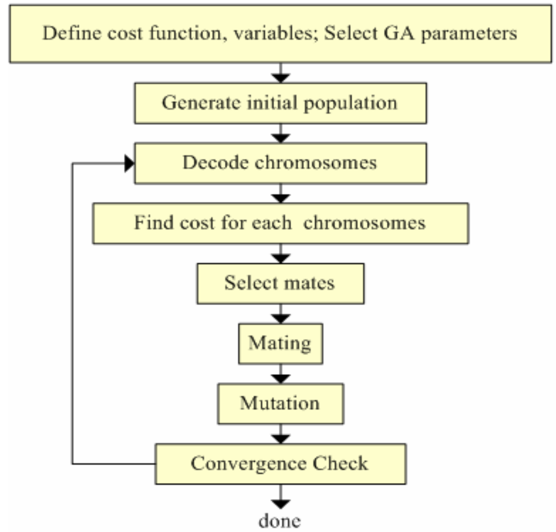


Figure 2. Flowchart of GA

## IV. IMPLEMENTATION GA ON DESIGN OF PID CONTROLLER FOR PSS

The transfer function of a PID controller is described as follows:

$$G_c(s) = k_p + \frac{k_i}{s} + k_d s \tag{10}$$

where $k_p$ ,$k_i$ and $k_d$ are the proportional, integral and derivative gains, respectively. A set of appropriate control parameters $k_p$,$k_i$ and $k_d$ can make a appropriate step change responses of Δω (angular velocity deviation) and Δδ (load angle deviation) that will result in performance criteria minimization. A Performance criterion in the time domain includes the overshoot $M_p$ and settling time $t_s$. Parameters $k_p$ ,$k_i$ and $k_d$ can make a good response that will result in performance criteria minimization. In order to achieve this target, the following cost function is suggested:

$$f(k)=(1-e^{-\beta})(M_p-1) + e^{-\beta}(t_s) \tag{11}$$

Where K is [$k_d$ ,$k_p$ ,$k_i$ ] and β is the weighting factor .If β is set to be smaller than 0.7 the settling time is reduced and if it set to be larger than 0.7 the overshoot is reduced. The discrete GA for searching optimal PID controller parameters is as follows:

At first, the lower and upper bounds of controller parameters are specified and initial population is produced randomly. Each solution point (each chromosome) is a 24-bits string and divided to three sections, each section related to a variable, $k_d$, $k_p$, ki. Total of solutions K (controller parameters) are sent to MATLAB® Simulink® block and on the other hand the values of two performance criteria in the time domain namely $M_p$ and $t_s$

are calculated for each chromosome and cost function is evaluated for each point according to these performance criteria.

Then, natural selection, selection, crossover (mating) and mutation operations are applied to population and the next iteration (generation) is started. At the end of each iteration , program checks the stop criterion. If the number of iterations reaches, the maximum or the stopping criterion is satisfied, records the latest global best solution and stop the algorithm. The best parameters of GA program are selected with trial and error method. Population size of chromosomes and number of generation (Gen.) are 10 and 40, respectively. Mutation rate is selected %5; selection rate is %60 with two-point-crossover.

## V. SIMULATION RESULTS

In this section the best operations of GA and associated parameters are selected, the lower and upper bound of controller parameters are adjusted to 0 and 60, respectively. Then a %5 load disturbance at time 1 second is exerted to under study power system and the GA runs in two modes, S.P.C and T.P.C in order to tune controller parameters, $k_p$ ,$k_d$ ,$k_i$ and weighing factor($\beta$) is adjusted to 0.7.Results of optimal PID parameters in various modes of GA are summarized in Table 2. Convergence curves of them are shown in figure 3.

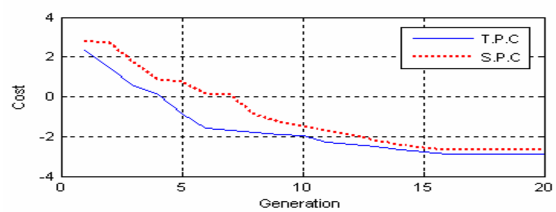

Figure 3. Convergence curves of GA for two types of crossover

Figure 4 shows the angular velocity response of the system without PID-PSS. Figures 5-10 illustrate angular velocity, load angle deviations with GA, LQR and FUZZY PID-PSS, and compare them.

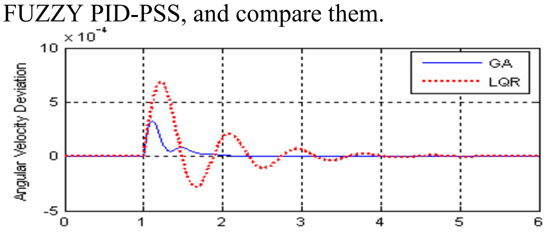

Figure 4. Angular velocity deviation due to 0.05 load disturbance at t=1 second with GA and LQR

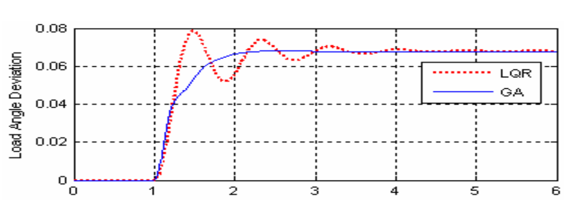

Figure 5. Load angle deviation due to 0.05 load disturbance at t=1 second with GA and LQR

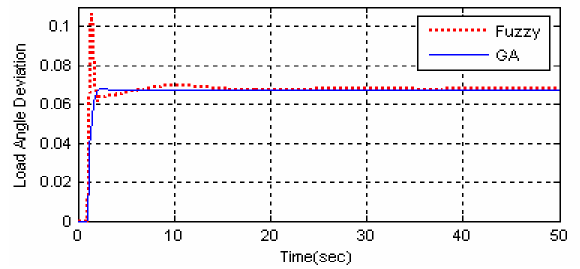

Figure 6. Load angle deviation due to 0.05 load disturbance at t=1 second with GA and FUZZY

Table 2. Optimum PID parameters in various modes

| | Ge n. | $N_{pop}$ | $X_{rate}$ % | µ % | Min. cost | $k_p$ | $k_i$ | $k_d$ |
|---|---|---|---|---|---|---|---|---|
| SPC | 20 | 10 | 50 | 1 | -2.86 | 49.9 | 44.1 | 25.1 |
| | 40 | 20 | 60 | 5 | -2.92 | 38.8 | 34.2 | 15.4 |
| TPC | 20 | 10 | 50 | 1 | -2.89 | 48.7 | 43.8 | 23.2 |
| | 40 | 20 | 60 | 5 | -2.96 | 35.6 | 30.0 | 11.8 |

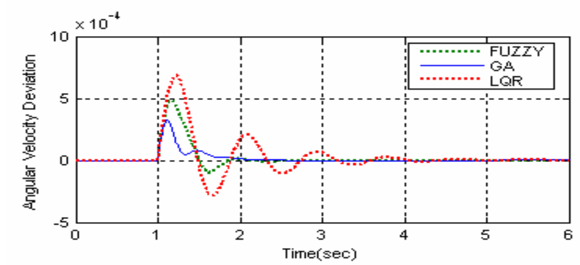

Figure 7. Angular velocity deviation due to 0.05 load disturbance at t=1 second with GA,LQR and FUZZY

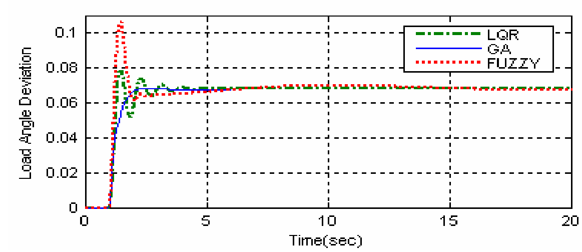

Figure 8. Load angle deviation due to 0.05 load disturbance at t=1 second with GA, FUZZY and LQR

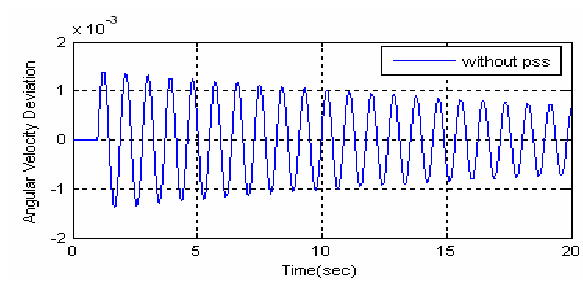

Figure 9. Load angle deviation due to 0.05 load disturbance at t=1 second without controller (PSS)

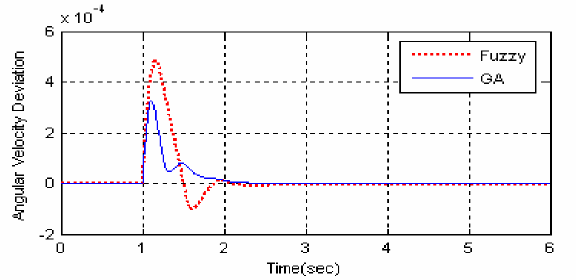

Figure 10. Angular velocity deviation due to 0.05 load disturbance at t=1 second with GA and FUZZY

## VI. CONCLUSION

In this study, proportional-integral-derivative power system stabilizer (PID-PSS) has been proposed for the enhancement of dynamic stability of modern power systems. Gain setting of PID-PSS has been optimized using advanced genetic algorithm by presentation of a new cost function in time domain. The proposed method is implemented on a case study power system and simulation results reveal that GA method gives much better dynamic performances as compared to that of LQR and FUZZY PID-PSS.

## REFERENCES

1. G.N. Taranto, D.M. Falcao, Robust decentralized control design using genetic algorithms in power system damping control, IEE Proc.-Gener. Transm.Distrib, vol. 145, no.1, pp. 1–6, January 1998.
2. Rashidi, M.; Rashidi, F.; Monavar, H.; Tuning of power system stabilizers via genetic algorithm for stabilization of power system, Systems, Man and Cybernetics, 2003. IEEE International Conference on Volume 5, 5-8 Oct. 2003,pp.4649 - 4654 vol.5
3. Ali M. Yousef, M. K. El-Sherbiny, Improvement of Synchronizing and Damping Torque Coefficients Based LQR power System Stabilizer, Electrical, Electronic and Computer Engineering, 2004. ICEEC '04. 2004 International Conference on
5-7 Sept. 2004 ,pp.753 – 758.
4. Lin, Y.J.; Systematic approach for the design of a fuzzy power system stabilizer, Power System Technology, 2004. PowerCon 2004. 2004 International Conference on Volume 1, 21-24 Nov. 2004, pp.747 - 752 Vol.1.
5. Bandal, V.; Bandyopadhyay, B.; Kulkarni, A.M.; Output feedback fuzzy sliding mode control technique based power system stabilizer (PSS) for single machine infinite bus (SMIB) system', Industrial Technology, 2005. ICIT 2005. IEEE International Conference on 14-17 Dec. 2005, pp.341 – 346.
6. X.Lei,H.Huang, S. L. Zheng,D. Z Jiang, Z.W. Sun, Global tuning of power system stabilizer in ,multi-machine systems, Electric Power System Research , Vol.58,2001,pp.103-110.
7. D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company, Inc., 1989.
8. C.L. Karr and D.E. Goldberg, Genetic Algorithm Based Design of an Air-Injected Hydro cyclone", Control 90, Mineral and Metallurgical Processing, 1990, pp. 265-272.
9. D.E.Goldberg and K Sastry (2001). A Practical Schema Theorem for Genetic Algorithm Design and Tuning. In proceedings, Genetic and Evolutionary Computation Conference,pp. 328-335