

Research Article

Stabilizing of Subspaces Based on DPGA and Chaos Genetic Algorithm for Optimizing State Feedback Controller

**M. Hosseinpour,¹ P. Nikdel,¹ M. A. Badamchizadeh,¹
and M. A. Poor²**

¹ *Control Engineering Department, Faculty of Electrical and Computer Engineering,
University of Tabriz, Tabriz 5166614776, Iran*

² *Department of Electrical and Computer Engineering, University of Alberta,
Edmonton, AB, Canada T6G 2V4*

Correspondence should be addressed to M. Hosseinpour, m.hosseinpour88@ms.tabrizu.ac.ir

Received 17 May 2012; Accepted 2 October 2012

Academic Editor: Jui-Sheng Lin

Copyright © 2012 M. Hosseinpour et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The main purpose of the paper is to optimize state feedback parameters using intelligent method, GA, Hermite-Biehler, and chaos algorithm. GA is implemented for local search but it has some deficiencies such as trapping into a local minimum and slow convergence, so the combination of Hermite-Biehler and chaos algorithm has been added to GA to avoid its deficiencies. Dividing search space is usually done by distributed population genetic algorithm (DPGA). Moreover, using generalized Hermite-Biehler Theorem can find the domain of parameters. In order to speed up the convergence at the first step, Hermite-Biehler method finds some intervals for controller, in the next step the GA will be added, and, finally, chaos disturbance will help the algorithm to reach a global minimum. Therefore, the proposed method can optimize the parameters of the state feedback controller.

1. Introduction

One of the first applications of state-space approach in state feedback is to shift the system eigenvalues. According to Kalman et al. [1], J. Bertram in 1959 was the first person who discovered that if the realization of the system is in the controllable mode, characteristic equation can be used with any equation of state feedback and the poles can be located in the appropriate places [2]. Three years later in 1962, Rosenbrock used state feedback for shifting system eigenvalues to obtain better response. As a fact, design of state feedback is used for stabilizing and improving the performance of systems [3].

In this paper, genetic algorithm with distributed parameters is used to obtain the parameter of state feedback controller for unstable systems with time delay. To do so, the theory of Hermite-Biehler is used in order to calculate stable areas of controller in DPGA. As mentioned above, this theory can be applied to the stable systems. As a result, in the first step, FOPDT, which is an unstable system, must be stabilized with proportional or other types of controllers as an inner loop. In the next step, the stable state feedback parameters are found by genetic algorithm to optimize the main system [4].

In DPGA, the search space is divided into several subspaces. Genetic algorithm is done in each of these subspaces to achieve the best solution. Then by comparing the best answers of subspaces, the best subspaces is selected for next division. Therefore, the performance of algorithm is improved by decreasing the search space [4]. Then in the new division the space is divided into several subspaces that each one does not cross others.

The paper is organized as follows. A brief review of genetic algorithm is described in Section 2 and combination of DPGA and chaos GA is presented in Section 3. An experiment that is designed by combination of DPGA and chaos algorithm is proposed in Section 4 and finally we draw conclusion of this method in Section 5.

2. A Brief Review of Genetic Algorithm

A genetic algorithm is a method to search an optimal solution and genetic algorithm is a kind of evolutionary computing, which is part of artificial intelligence. This algorithm is not random search algorithm because of special properties. The main idea of this method is inspired by biological processes of living organisms. The nature of successive generations, according to the law of "adaptation for survival" selects creatures. This law, first, was raised by Darwin who suggested that genetic algorithm can be used to solve real world problems [5]. The first idea of evolutionary computation, in 1960, is proposed by Rechenberg and this theory was later used by other researchers and has been developed. Since the basic principle of genetic algorithms was presented by Holland and colleagues at the University of Michigan in 1962, then in 1975, their mathematical foundation was published in a book in the Netherlands with the name *Adaptation in natural and artificial systems* [6]. In 1992, Koza used genetic algorithm to perform specific tasks in the program and called this method evolutionary programming [7]. Its goal is to evaluate the performance of algorithm in order to find the answer to any problem. In recent years, genetic algorithms have been usually used in optimization problems. Important property of this algorithm is its robustness, so that a flexible balance exists between performance and the characteristics required for survival in the environment.

3. Combination of DPGA and Chaos GA

3.1. Hermite-Biehler Theorem

In this section we consider $\delta(s) = \delta_0 + \delta_1 s + \dots + \delta_n s^n$ as a real polynomial of transfer function of degree n and define $\delta(s)$ as $\delta(s) = \delta_e(s^2) + s\delta_o(s^2)$, where $\delta_e(s^2)$, $\delta_o(s^2)$ are the components of $\delta(s)$ made up of even and odd powers of s for $\omega \in R$, $\delta(j\omega) = p(\omega) + jq(\omega)$ where $p(\omega) = \delta_e(-\omega^2)$, $q(\omega) = \omega\delta_o(-\omega^2)$ [8].

Let $\omega_{e_1}, \omega_{e_2}, \dots$ and $\omega_{o_1}, \omega_{o_2}, \dots$ denote the nonnegative real zeros of $\delta_e(-\omega^2)$ and $\delta_o(-\omega^2)$ are arranged in ascending order. Then $\delta(s)$ is Hurwitz stable if and only if

all zeros of $p(\omega)$, $q(\omega)$ are real and distinct and $0 < \omega_{e_1} < \omega_{o_1} < \omega_{e_2} < \omega_{o_2} < \dots$. Let $0 = \omega_0 < \omega_1 < \dots < \omega_{m-1}$ be the real, nonnegative, distinct finite zeros of $q(\omega)$ with odd multiplicities. Also define $\omega_\infty = \infty$, then

$$\sigma(\delta(s)) = \begin{cases} \left\{ \text{sgn}[p(\omega_0)] - 2 \text{sgn}[p(\omega_1)] + \dots + (-1)^{m-1} 2 \text{sgn}[p(\omega_{m-1})] \right. \\ \left. + (-1)^m 2 \text{sgn}[p(\omega_m)] \right\} \cdot \gamma & \text{if } n \text{ is even,} \\ \left\{ \text{sgn}[p(\omega_0)] - 2 \text{sgn}[p(\omega_1)] + 2 \text{sgn}[p(\omega_2)] \right. \\ \left. + \dots + (-1)^{m-1} 2 \text{sgn}[p(\omega_{m-1})] \right\} \cdot \gamma & \text{if } n \text{ is odd,} \end{cases} \quad (3.1)$$

where $\gamma = (-1)^{l-1} \text{sgn}[q(k_p, \infty)]$ and $\sigma(\delta(s)) =$ number of open left half plane zeros of $\delta(s)$ –number of open right half plane zeros of $\delta(s)$ [9].

3.2. Chaos Algorithm

Chaos algorithm is used in nonlinear systems. This algorithm is named due to chaotic disturbance that avoid being trapped in local minimum.

Integrating with chaos algorithm, genetic algorithm uses its dynamic characteristic as well as its random property for fast convergence and finding optimal solution. In the following, we discuss chaos in detail.

We use the following logical equation for the production of initial variables:

$$X_{n+1} = \mu X_n (1 - X_n) \quad n = 0, 1, \dots, N, \quad (3.2)$$

where $\mu = 4$, ($X_0 \neq 0.25, 0.5, 0.75$), $0 < X_0 < 1$.

Chaos variables are very sensitive to initial values; little change in initial values causes completely changes in chaotic variables [10].

Chaos algorithm has characteristics of high efficiency such as regularity, ergodicity, and stochastic properties and uses chaotic variables for optimal search. Using chaos algorithm in GA forms new method which helps to avoid trapping into local minimum and results in a fast convergences to a global optimal solution.

3.3. Stable Region for Unstable FOPDT

FODPT is described by the transfer function $G(s) = (ke^{-\theta s})/(Ts - 1)$ that is usually normalized by the delay time ($L = \theta/T$) and when it is applied to the system; $G(s)$ is as below:

$$G(s) = \frac{e^{-Ls}}{s - 1}. \quad (3.3)$$

Controller with closed-loop characteristic equation will be as follows:

$$\delta(s) = D(s) + k_p N(s). \quad (3.4)$$

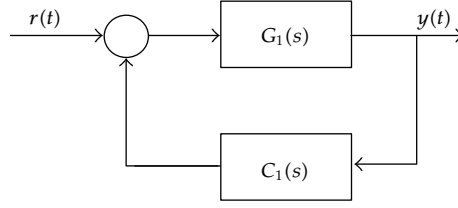


Figure 1: Controller of inner loop.

There are many classical methods to determine the value of k_p that $\delta(s)$ is Hurwitz such as technique of root locus, Nyquist stability criterion, and the Routh-Hurwitz criterion. Since the first two methods are graphical methods and fail to provide analytical characteristic of values of stable k_p , the Routh-Hurwitz criterion is chosen according to its ability in analytical solution. However, using this method for the stabilization process of the FOPDT does not always have a desired value of k_p . To obtain stable range of k_p , a structure with two controllers in [11] has been adopted.

In Figure 1, $C_1(s)$ is a controller that is shown in (3.5). The purpose of controlling system is stabilization and the main feedback controller is applied to the overall system which is shown in Figure 1.

$$C_1(s) = \begin{cases} k_{p1} & L \leq 1 \\ k_{p1}(1 + T_{d1})s & 1 < L < 2. \end{cases} \quad (3.5)$$

3.4. Stable Domain of the State Feedback Controller by DPGA

The controller design is in the following form:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \quad u(t) = -Kx(t) + r, \end{aligned} \quad (3.6)$$

where r is a reference input and

$$K = [k_1, k_2, \dots, k_n], \quad (3.7)$$

where K is state feedback controller with variable range of the overall transfer function. So the closed-loop characteristic polynomial can be obtained as follows:

$$\zeta(s) = \zeta_e(s) + s\zeta_o(s). \quad (3.8)$$

Equation (3.8) is the transfer function of the system and ζ_e and ζ_o contained the controller parameters and the closed-loop characteristic polynomial, and $\zeta(s, K)$ is Hurwitz. By considering Hermite-Biehler Theorem, $\zeta_e(w)$ and $\zeta_o(w)$ can be obtained to have a k_i in the odd powers and these k_i will not depend on k_j in the even powers.

3.5. Procedures of DPGA and Chaos GA

Step 1. Size of population and length of chromosome are selected and the objective function is chosen as follows [12]:

$$J_{IAE} = \int_0^t |e(t)| dt, \quad J_{ITAE} = \int_0^t t |e(t)| dt, \quad (3.9)$$

$$J = J_{IAE} + J_{ITAE},$$

where $e(t)$ is the closed loop error and the fitness function of each gene is calculated as follows:

$$F = \frac{1}{1 + J}. \quad (3.10)$$

Step 2. Assume $k_j = 0$, stable region of k_i , is obtained based on Routh-Hurwitz criterion [4].

Step 3. The number of search subspace (N_z) is chosen and the scan space is $(k_{i_{\max}} - k_{i_{\min}})/N_z$ (for $i = 1, \dots, r$).

Step 4. In each subspace, k_i s are selected and for every k_i suitable domain of k_j are obtained (for $j = 1, \dots, l$) and the biggest region of k_j are chosen as the search subspace.

Strings u_t are chosen in a way that (3.11) is satisfied as follows:

$$n - \sigma(N') = \begin{cases} \{t_0 - 2t_1 + (-1)^{m-1}2t_{m-1} + (-1)^m t_m\} \gamma & \text{for } m' + n \text{ even} \\ \{t_0 - 2t_1 + (-1)^{m-1}2t_{m-1}\} \gamma & \text{for } m' + n \text{ odd,} \end{cases} \quad (3.11)$$

and the ranges of k_j s are determined by solving the following inequalities (for $t = 0, 1, \dots, m$):

$$\begin{aligned} \zeta_e(w_t) &> 0 & \text{if } t_t = 1, \\ \zeta_e(w_t) &< 0 & \text{if } t_t = -1. \end{aligned} \quad (3.12)$$

Step 5. Chaos algorithm is applied in the following steps.

Step 5.1. Generation of initial population according to (3.2).

Step 5.2. Continuing the steps as standard genetic algorithm.

Step 5.3. Selection of top 10 to 15 percent of the population.

Step 5.4. Repetition of Steps 2 and 3 to achieve desired condition.

Step 6. The parameters of genetic algorithm are generated for every subspace such as selection, crossover, and mutation.

Step 7. If the end condition is met, the best solution among subspaces is selected.

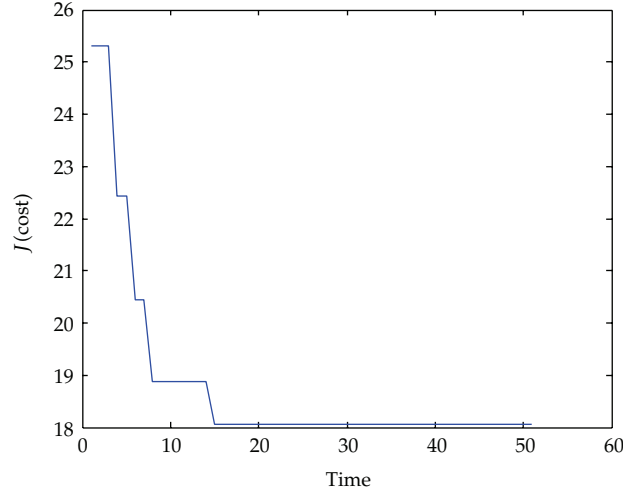


Figure 2: Convergence in DPGA and chaos GA.

Table 1: The parameters of inner loop of $G_1(s)$.

System	Inner loop	Result
$G_1(s)$	$k_{p1}(1 + T_{d1})s$	$k_{p1} = 0.119, T_{d1} = 0.169$

4. Simulation Results

The following examples illustrate that combination of DPGA and chaos algorithm has a good response. For this algorithm, the number of bits for any variable is 8, and number of iterations equals to 50 and $N_z = 3$.

The transfer functions of the plants are given as follows:

$$G_1(s) = \frac{e^{-3.5s}}{3s - 1}, \quad (4.1)$$

$$G_2(s) = \frac{4s + 1}{(s + 1)^3}.$$

The results of inner loop of $G_1(s)$ are shown in Table 1 and the ranges of state feedback controller are represented in Tables 3 and 5. The search subspaces of DPGA and chaos GA based on Hermite-Biehler Theorem are shown in Tables 2 and 4. The transfer functions of systems using Integrated Time Absolute Error (ITAE) as performance index have small overshoots, short rise times, and fast transition process according to ITAE [12]. Therefore, the fitness function is chosen to have a good result. Figures 2 and 4 show the change of fitness values. Figures 3 and 5 show that combination of DPGA and chaos algorithm for feedback controller has a better result than standard GA and chaos.

Table 2: The search subspaces of $G_1(s)$ with Hermite-Biehler Theorem.

Feedback controllers parameters		
$-10 < k_1 < -0.04763$	$-10 < k_1 < -0.04763$	$-5 < k_1 < -0.04763$
$-3.351 < k_2 < -0.013$	$-6.668 < k_2 < -3.351$	$-6.668 < k_2 < -10.026$
$-10 < k_3 < -0.02733$	$-10 < k_3 < -0.02733$	$-10 < k_3 < -0.02733$

Table 3: The parameters of controller for $G_1(s)$.

Method	Feedback controller			Cost
	k_1	k_2	k_3	
GA	-4.101	-3.213	-8	36.76
Chaos GA	0.571	-2.921	-5.693	25.27
DPGA + chaos GA	-0.045	-0.335	-0.027	18.12

Table 4: The search subspaces of $G_2(s)$ with Hermite-Biehler Theorem.

Feedback controllers parameters		
$-5 < k_1 < 5$	$-5 < k_1 < 5$	$-5 < k_1 < 5$
$-1.768 < k_2 < 1.555$	$1.555 < k_2 < 5.778$	$5.778 < k_2 < 10$
$-1.024 < k_3 < 10$	$-1.024 < k_3 < 10$	$-1.024 < k_3 < 10$

Table 5: The parameters of controller for $G_2(s)$.

Method	Feedback controller			Cost
	k_1	k_2	k_3	
GA	-1.9	-2.451	-1.352	71.9724
Chaos GA	-3.267	-4.653	-1.552	69.6884
DPGA + chaos GA	-4.113	-5.875	-0.953	52.3323

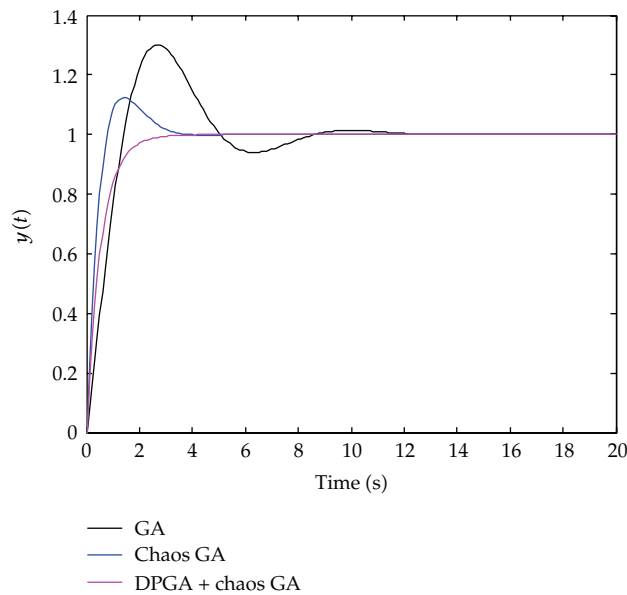


Figure 3: Comparison of three methods for $G_1(s)$.

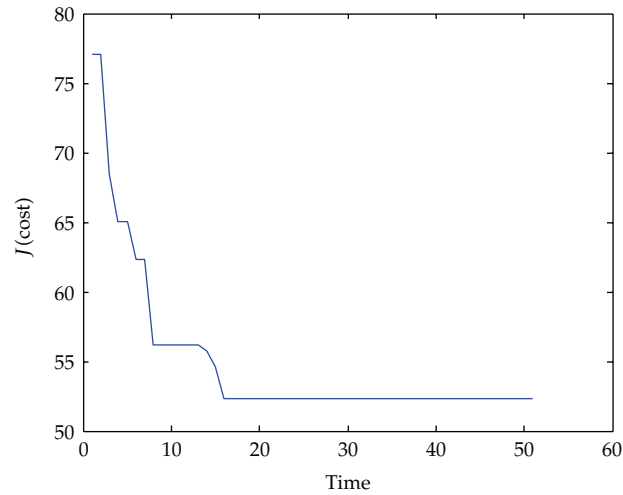


Figure 4: Convergence in DPGA and chaos GA.

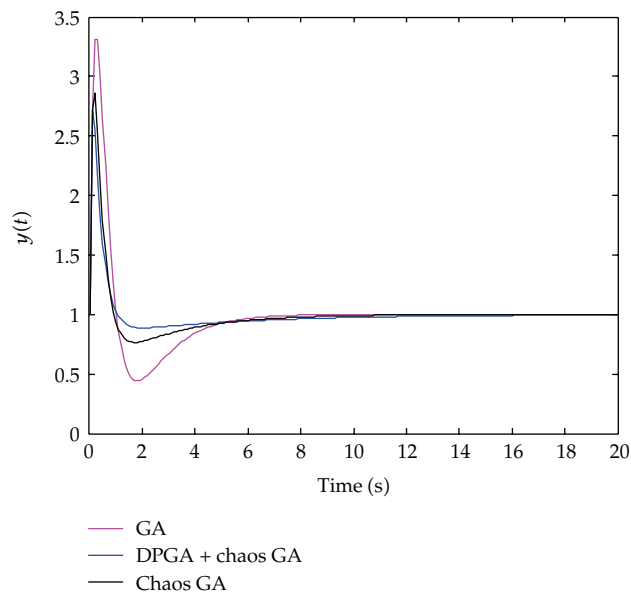


Figure 5: Comparison of three methods for $G_2(s)$.

5. Conclusion

A distributed population genetic algorithm with the chaos algorithm has been presented to optimize state feedback controller and this algorithm is used for that stable and unstable FODPT processes. In the first step unstable FODPT is stabilized by the inner loop and then the feedback controller is designed. A chaos genetic algorithm takes the advantage of chaotic variables to make the search of optimal values more effective and faster. In other words, chaos genetic algorithm integrates powerful global searching capability of the GA with that

of powerful local searching capability of the chaos. The results show that the combination of DPGA and chaos algorithm has a good performance such as speed of convergence.

References

- [1] R. E. Kalman, P. L. Falb, and M. A. Arbib, *Topics in Mathematical System Theory*, McGraw-Hill Book, New York, NY, USA, 1969.
- [2] K. Ogata, *Modern Control Engineering*, Prentice Hall PTR, 2001.
- [3] H. Rosenbrock, "Distinctive problems of process control," *Chemical Engineering Progress*, vol. 58, pp. 43–50, 1962.
- [4] R. L. Haupt, S. E. Haupt, and J. Wiley, *Practical Genetic Algorithms*, Wiley-Interscience, Hoboken, NJ, USA, 2nd edition, 2004.
- [5] T. Jili, W. Ning, and H. Xiongxiang, "Stabilization subspaces based DPGA for optimizing PID controllers," in *Proceedings of the 27th Chinese Control Conference (CCC'08)*, pp. 38–42, Kunming, China, July 2008.
- [6] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press, Boston, Mass, USA, 1992.
- [7] W. Banzhaf, J. Koza, C. Ryan, L. Spector, and C. Jacob, "Genetic programming," *IEEE Intelligent Systems and Their Applications*, vol. 15, pp. 74–84, 2000.
- [8] A. Roy and K. Iqbal, "PID controller design for first-order-plus-dead-time model via Hermite-Biehler theorem," in *Proceedings of the IEEE American Control Conference*, vol. 6, pp. 5286–5291, June 2003.
- [9] G. J. Silva, A. Datta, and S. P. Bhattacharyya, *PID Controllers for Time-Delay Systems*, Control Engineering, Birkhäuser, Boston, Mass, USA, 2005.
- [10] T. Wu, Y. Cheng, J. Tan, and T. Zhou, "The application of chaos genetic algorithm in the PID parameter optimization," in *Proceedings of the 3rd International Conference on Intelligent System and Knowledge Engineering (ISKE'08)*, pp. 230–234, Xiamen, China, November 2008.
- [11] C. Xiang and L. A. Nguyen, "Control of unstable processes with dead time by PID controllers," in *Proceedings of the 5th International Conference on Control and Automation (ICCA'05)*, vol. 2, pp. 703–708, June 2005.
- [12] Y. Tian and H. Q. Vu, "A multiple population genetic algorithm and its application in fuzzy controller," in *Proceedings of the 3rd International Symposium on Intelligent Information Technology and Security Informatics (IITSI'10)*, pp. 729–733, Jingtangshan, China, April 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

