

Chaos Suppression Using Genetically Optimized PID Control of the 4-D Novel Hyperchaotic Vaidyanathan System

Guessas Laarem

Department of Electronics
University Ferhat Abbas Setif 1
Setif, Algeria
l_guessas@univ-setif.dz

Abiza Merbouha

Department of Electronics
University Ferhat Abbas Setif 1
Setif, Algeria
liali87@yahoo.fr

Abstract—In this paper, a dynamical analysis of the novel hyperchaotic system with four parameters is presented. Genetically optimized proportional integral and derivative (PID) controllers were designed and applied for the chaos suppression of the 4-D novel hyperchaotic system, by varying the genetic algorithms (GA) options to view the impact factor on the optimized PID controllers. The use of the final optimized PID controllers ensures less time of convergence and fast speed chaos suppression.

Keywords—hyperchaos; chaos suppression; PID control; optimization; genetic algorithm

I. INTRODUCTION

The study of chaotic and hyper chaotic attractors arising in nonlinear dynamical systems has received much attention as it has potential applications in many branches of science and engineering. Chaotic systems are characterized as nonlinear dynamical systems, are irregular, aperiodic, unpredictable and have sensitive dependence on the initial conditions [1-4]. A hyperchaotic system is mathematically defined as a chaotic system having more than one positive Lyapunov exponents implying that its dynamics are expanded simultaneously in many different directions. Thus, a hyperchaotic system has more complex dynamical behaviors than a chaotic system [5]. There are many well-known hyperchaotic systems [6-10] and the 4-D novel hyperchaotic Vaidyanathan system [11]. PID control, in which parameters are optimized by genetic algorithm, is used in chaos suppression and synchronization for chaotic and hyper chaotic systems [11-16]. In this paper, the qualitative properties of the novel hyperchaotic system are described. The Lyapunov exponents of the system are having two positive signs showing that the system is hyperchaotic.

II. DYNAMICAL ANALYSIS OF THE NOVEL 4-D HYPERCHAOTIC SYSTEM

In this work, the Vaidyanathan novel hyperchaotic system is described by (1) [7]:

$$\dot{X} = f(X) = \begin{cases} \dot{x} = a(y - x) + w \\ \dot{y} = cy - xz \\ \dot{z} = xy - bz + y \\ \dot{w} = xz + rw \end{cases} \quad (1)$$

where x, y, z, w are the variable states, and a, b, c, r are positive constant parameters. In this work, we show that the system (1) is hyperchaotic when we take parameters as:

$$a = 39, b = 3, c = 22, r = 1 \quad (2)$$

For numerical simulations, we take the initial values of the hyperchaotic system as:

$$\begin{aligned} x(0) &= 0.2, & y(0) &= 0.2 \\ z(0) &= 0.2, & w(0) &= 0.2 \end{aligned} \quad (3)$$

Figures 1, 2 show the 3-D projections of the novel 4-D hyperchaotic system (1) on (x, y, z) , (x, y, w) space projections, respectively.

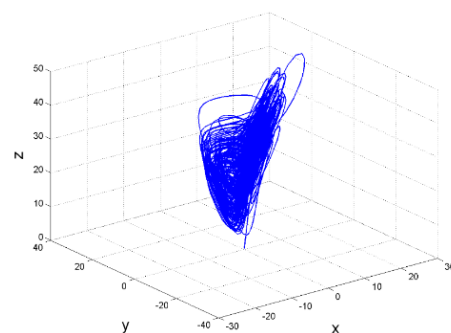


Fig. 1. 3-D projection of the novel 4-D hyperchaotic system on (x, y, z) space.

A. Equilibrium Points

The equilibrium points of the novel hyperchaotic system (1) are obtained by solving (4) [11]:

$$\dot{X} = f(X) = 0 \quad (4)$$

The parameter values are taken as in the hyperchaotic case (2), and are found as:

$$E_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} E_1 = \begin{bmatrix} 7.6394 \\ 14.4163 \\ 41.5161 \\ -264.2990 \end{bmatrix} E_2 = \begin{bmatrix} -8.6394 \\ -16.3034 \\ 41.5161 \\ 298.8957 \end{bmatrix} \quad (5)$$

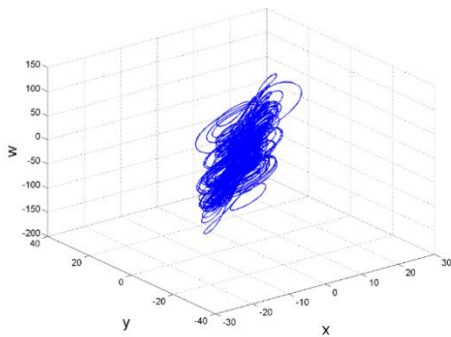


Fig. 2. 3-D projection of the novel 4-D hyperchaotic system on (x, y, w) space.

B. The Lyapunov Exponents

The Lyapunov exponents of the system (1) are numerically determined as:

$$\begin{aligned} L_1 &= 1.0623, & L_2 &= 0.2952 \\ L_3 &= 0, & L_4 &= -20.1063 \end{aligned} \quad (6)$$

Since there are two positive Lyapunov exponents in (3), it is clear that the 4-D system (1) is hyperchaotic.

C. The Kaplan-Yorke Dimension

The Kaplan-Yorke dimension of the 4-D hyperchaotic system (1) is derived as in (7):

$$D_{KY} = 3 + \frac{L_1 + L_2 + L_3 + L_4}{|L_4|} = 3.0675 \quad (7)$$

which is fractional as is in [1].

III. GENETICALLY OPTIMIZED PID CONTROL OF THE SYSTEM.

Control of nonlinear systems is difficult in the absence of a systematic procedure as is available for linear systems. Many techniques are limited in their application to a special class of systems. Here again, more commonly available methods are heuristic in nature and the genetic algorithm can reduce the arbitrariness in the design of a controller to a great extent. However, even if a model of the nonlinear system is available, no systematic and generally applicable control theory is available for the design of controllers for nonlinear systems. The best-known controllers used in industrial control processes are proportional-integral derivative (PID) controllers because of their simple structure and robust performance in a wide range of operating conditions.

In the past decades, control theory has gone through major developments. Advanced and intelligent control algorithms have been developed in [18-20]. However, the PID-type controller remains the most popular in industry. Reasons for

this are the simplicity of this control law and its few tuning parameters. Hundreds of tools, methods and theories are available for this purpose. However, finding the appropriate parameters for the PID controller is still a difficult task, so in practice control engineers still often use trial-and-error for the tuning process. PID control consists of three types of control, Proportional, integral and derivative control. PID controller algorithm can be implemented in many forms but they are mostly used in feedback loops in [11-16]. A simple diagram illustrating the schematic of the PID controller can be seen in Figure 3 [11, 15], where:

$$\begin{aligned} u_i &= K_p e_i + K_I \int_0^t e_i(\tau) d\tau - K_D \frac{de_i}{dt}, \\ i &= x, y, z, w \end{aligned} \quad (8)$$

where u_i is the output PID controller, e_i is the error signal, and K_p, K_I, K_D are the proportional, integral and, the derivative gains respectively.

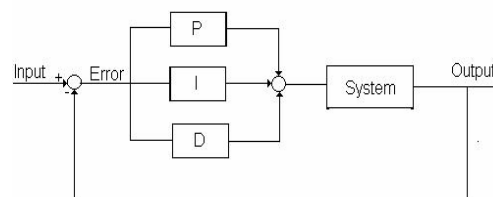


Fig. 3. Schematic of the PID control

The proportional controller output uses a “proportion” of the system error to control the system. However, this introduces an offset (static) error into the system. The integral controller output is proportional to the amount of time there is an error present in the system. The integral action removes the offset introduced by the proportional control but introduces a phase lag into the system. The derivative controller output is proportional to the rate of change of the error. Derivative control is used to reduce or eliminate overshoot and introduces a phase lead action that removes the phase lag introduced by the integral action. The effective PID controller’s parameters K_p, K_I, K_D are optimized using genetic algorithms. Genetic Algorithm (GA) is a stochastic global search method that mimics the process of natural evolution [18-20]. It starts with no knowledge of the correct solution and depends entirely on responses from its environment and evolution operators (i.e. reproduction, crossover and mutation) to arrive at the best solution. By starting at several independent points and searching in parallel, the algorithm avoids local minima and converging to sub optimal solutions.

IV. GENETIC ALGORITHM STAGES

There are three main stages of a (GA), known as reproduction, crossover and mutation (Figure 4).

- **Reproduction:** Just like in natural evolution, an example of a common selection technique is the “roulette wheel” selection method.
- **Crossover:** Once the selection process is complete, the crossover algorithm is initiated. The simplest crossover

technique is the single point crossover.

- Selection: Using selection and crossover on their own will generate an amount of different strings. If there is not enough diversity in the initial strings, the GA may converge on sub-optimum strings due to a bad choice of initial population. This problem may be overcome by the introduction of a mutation operator into the GA.
- Mutation: Mutation is the occasional random alteration of a value of a string position. The probability of mutation is normally low around 0.1% - 0.01%. Once a string is selected for mutation, a randomly chosen element of the string is changed or 'mutated'.

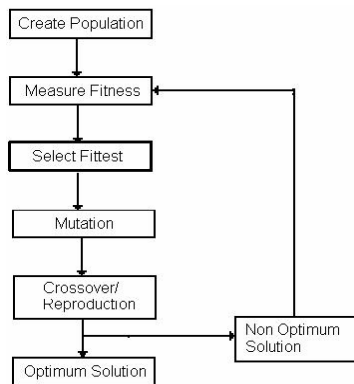


Fig. 4. The GA process

The steps in creating and implementing a GA are: [19]

- Generate an initial, random population of a fixed size of individuals.
- Evaluate their fitness.
- Select the fittest members of the population.
- Reproduce using a probabilistic method (e.g., roulette wheel).
- Implement crossover operation on the reproduced chromosomes.
- Execute mutation operation with low probability.
- Repeat step 2 until a predefined convergence criterion is met.
- The convergence criterion of a GA is a user-specified condition i.e. the maximum number of generations or when the string fitness value exceeds a certain threshold.

The hyper chaotic system with the PID controllers is described as in (9):

$$\dot{X} = f(X) = \begin{cases} \dot{x} = a(y - x) + w + u_x \\ \dot{y} = cy - xz + u_y \\ \dot{z} = xy - bz + y + u_z \\ \dot{w} = xz + rw + u_w \end{cases} \quad (9)$$

where:

$$u_i = K_p e_i + K_I \int_0^t e_i(\tau) d\tau - K_D \frac{de_i}{dt},$$

$$i = x, y, z, w \quad (10)$$

Since this work is using GA to optimize the gains of the PID controller, there are going to be three strings assigned to each member of the population, these members will be comprised of a K_p , K_I , K_D in (10). Each string will be evaluated throughout the course of the GA, real (floating point) numbers will be used to encode the population. The three terms are entered into the GA application in Simulink via the declaration of a three-row, between -0.2 and 0.2 which is called the initial range, the controlled system is given a step input and the error is assessed using an appropriate error performance criterion i.e.

$$e = E_0 - X \quad (11)$$

where

$$e = \begin{bmatrix} e_x \\ e_y \\ e_z \\ e_w \end{bmatrix}, E_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, X = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \quad (12)$$

The chromosome is assigned an overall fitness value according to the magnitude of the error, the smaller the error the larger is the fitness value.

A. GA Options:

- Variable bounds matrix=[-0.2,0.2],
- Population Size=80 individuals,
- Number of generations=100,
- Selection function=stochastic uniform,
- Crossover fraction=0.8,
- Mutation function=default function,
- Stopping criteria=error performance criterion or the maximum number of generations,

In order to evaluate the PID values chosen by GA, the objective function is written based on error performance criterion:

$$F_{obj} = \sum abs e_i \quad (13)$$

The objective function (13) is designed as the sum absolute error which gets the absolute value of the error to remove negative error components. This kind of error performance criteria is good for simulation studies. The best population may be plotted to give an insight into how GA converged to its final values. Table I shows the PID gain values of the best solution for the 4-D novel hyperchaotic system. For numerical simulations of the system, we take the parameter values of (2) and the initial conditions of (3).

Figure 5 represents the converged states to the equilibrium point, Figure 6 shows 3-D projection (x, y, z) space of chaos suppression, and Figure 7 shows the action control of the output of the PID optimized by the GA.

TABLE I. OPTIMIZED PID GAIN VALUES

PID Controller	K_p	K_I	K_D
u_x	0.8736	0.9530	-1.0075
u_y	-1.1834	0.8585	2.9629
u_z	1.4453	0.1727	-1.7600
u_w	2.5530	0.0005	2.8930

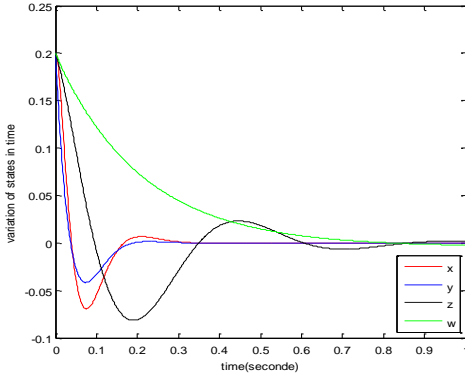


Fig. 5. Time-history of converged states (control in action at 0.8s)

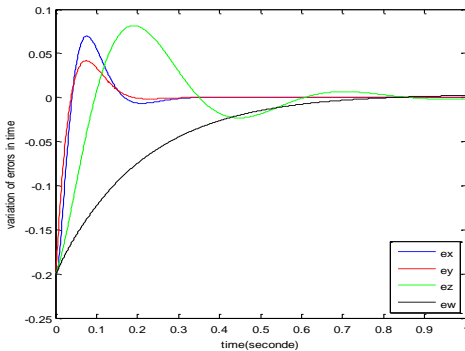


Fig. 6. Time-history of chaos suppression errors.

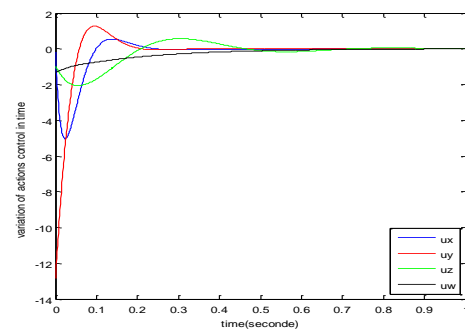


Fig. 7. Action control of the PID optimized by the GA

B. Impact Factor of the GA Options on the Optimized PID Controllers.

Genetic algorithm is typically characterized by three options. To get the optimum PID controllers, two of them remain fixed while the third option varies. Let ex , ex_1 , ex_2 the errors for the state x for the three bounds respectively.

1) The Initial Range Variable

The crossover and the mutation probability are kept fixed. Variable bounds matrix= $[-0.2;0.2]$, $[-0.5;0.5]$, $[-0.8;0.8]$, crossover fraction=0.8, mutation function=default function. Table II show the best PID optimized for the bounds matrix $[-0.8;0.8]$.

TABLE II. PID GAIN VALUES, BOUNDS MATRIX= $[-0.8;0.8]$

PID Controller	K_p	K_I	K_D
u_x	4.015	10.044	-3.437
u_y	-1.94	7.084	5.959
u_z	2.306	1.131	-8.61
u_w	10.119	0.002	11.846

Figure 8 represents the time history of chaos suppression errors for the state x , the best one is for the bound $[-0.8;0.8]$, the time of convergence is typically small.

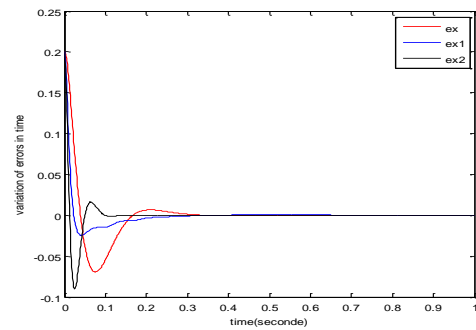


Fig. 8. Time-history of chaos suppression errors.

2) Fixed Crossover Fraction Variable, Bounds Matrix and Mutation Probability.

The best bound matrix $[-0.8;0.8]$ and the same probability mutation default function are used while the crossover rate varies (0.6;0.4;0.8). Table III represents the best PID optimized for crossover rate=0.6. The time of convergence is typically small (control in action at 0.08s) (Figure 9).

TABLE III. PID GAIN VALUES CROSSOVER RATE=0.6

PID Controller	K_p	K_I	K_D
u_x	-3.643	4.438	-11.021
u_y	2.007	17.038	7.866
u_z	16.364	2.53	-11.973
u_w	12.339	0.001	16.909

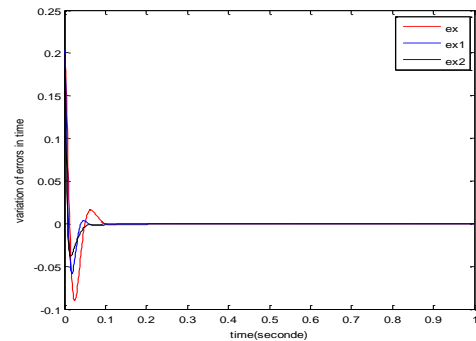


Fig. 9. Time-history of chaos suppression errors.

3) Fixed Mutation Probability Variable, Bounds Matrix and Crossover.

The best of the bound matrix [-0.8;0.8] and crossover rate=0.6 are used, while the probability mutation default function varies according to the Gaussian and adaptive feasible function. Table IV gives the best PID values.

TABLE IV. PID VALUES, GAUSSIAN FUNCTION

PID Controller	K_p	K_I	K_D
u_x	5.8108	8.9852	-11.8140
u_y	2.7394	9.8449	14.7133
u_z	9.4557	-0.0000	-11.7124
u_w	15.4197	11.3442	21.3892

Mutation probability=Gaussian function, the time of convergence is typically small (control in action at 0.10s). Figure 10 shows the chaos suppression.

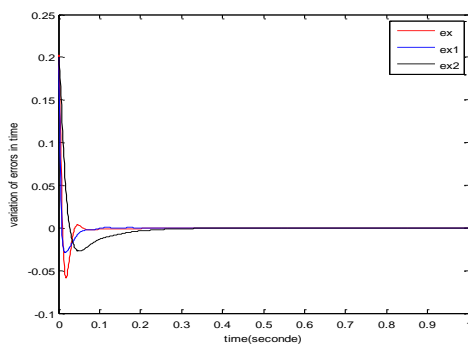


Fig. 10. Time-history of chaos suppression errors.

V. CONCLUSION

In this paper, a novel hyperchaotic system was described and its qualitative properties were discussed. Lyapunov exponents were calculated to prove that the system is hyperchaotic. Chaos suppression was done by the use of PID controller's gains which were optimized by the use of GA. After varying GA options, i.e. the initial range of the controller's gains, the crossover rate and the mutation function, in order to view their impact factor on the optimized PID controllers, the optimum controller was the one with the smallest time for chaos suppression. It was shown that the best controller had bound matrix [-0.8;0.8], crossover rate=0.6 and mutation probability=default function with control in action at 0.08s.

REFERENCES

- [1] S. Vaidyanathan, C. Volos, *Advances and Applications in Chaotic Systems*, Springer, Berlin, 2016
- [2] A. T. Azar, S. Vaidyanathan, *Advances in Chaos Theory and Intelligent Control*, Springer, Berlin, 2016
- [3] E. N. Lorenz, "Deterministic non-periodic flows", *Journal of the Atmospheric Sciences*, Vol. 20, No. 2, pp. 130-141, 1963
- [4] O. E. Rossler, "An equation for continuous chaos", *Physics Letters A*, Vol. 57, pp. 397-398, 1976
- [5] V. Dobrushkin, *Applied Differential Equations*, CRC Press, 2015
- [6] Q. Jia, "Hyperchaos generated from the Lorenz chaotic system and its control", *Physics Letters A*, Vol. 366, pp. 217-222, 2007

- [7] O. S.Ojoniyi, A. N. Njah, "A 5D hyperchaotic Sprott B system with coexisting hidden attractors", *Chaos, Solitons & Fractals*, Vol. 87, pp. 172-181, 2016
- [8] Q. Yang, M. Bai, "A new 5D hyperchaotic system based on modified generalized Lorenz system", *Nonlinear Dynamics*, Vol. 88, No. 1, pp. 189-221, 2017
- [9] Y. Li, W. K. S. Tang, G. Chen, "Generating Hyperchaos via state feedback control", *International Journal of Bifurcation and Chaos*, Vol. 15, No. 10, pp. 3367-3375, 2005
- [10] W. Fa-Qiang, L. Chong-Xin, "Hyperchaos evolved from the Liu chaotic system", *Chinese Physics*, Vol. 15, No. 5, pp. 963-968, 2006
- [11] K. Rajagopal, L. Guessas, S. Vaidyanathan, A. Karthikeyan, A. Srinivasan, "Dynamical Analysis and FPGA Implementation of a Novel Hyperchaotic System and Its Synchronization Using Adaptive Sliding Mode Control and Genetically Optimized PID Control", *Mathematical Problems in Engineering*, Vol. 2017, Article ID 7307452, 2017
- [12] A. Karthikeyan, K. Rajagopal, "Chaos Control in Fractional Order Smart Grid with Adaptive Sliding Mode Control and Genetically Optimized PID Control and Its FPGA Implementation", *Complexity*, Vol. 2017, Article ID 3815146, 2017
- [13] A. Jayachitra, R. Vinodha, "Genetic Algorithm Based PID Controller Tuning Approach for Continuous Stirred Tank Reactor", *Advances in Artificial Intelligence*, Vol. 2014, Article ID 791230, 2014
- [14] D. Davendra, I. Zelinka, R. Senkerik, "Chaos driven evolutionary algorithms for the task of PID control", *Computers and Mathematics with Applications*, Vol. 60, No. 4, pp. 1088-1104, 2010
- [15] L. Guessas, K. Benmahammed, "Adaptive Backstepping and PID optimized by Genetic Algorithm In Control of Chaotic systems", *International Journal of Innovative Computing, Information and Control*, Vol. 7, No. 6, pp. 5299-5312, 2011
- [16] W. D. Chang, "PID control for chaotic synchronization using particle swarm optimization Chaos", *Solitons & Fractals*, Vol. 39, No. 2, pp. 910-917, 2009
- [17] H. K. Khalil, *Nonlinear Systems*, Pearson Education, New York, 2002
- [18] R. L. Haupt, S. E. Haupt, *Practical Genetic Algorithms*, Wiley, New York, 2004
- [19] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, 1989
- [20] Q. Wang, P. Spronck, R. Tracht, "An Overview Of Genetic Algorithms Applied To Control Engineering Problems", *Second International Conference on Machine Learning and Cybernetics*, Xi'an, China, November 5, 2003