Journal of Unmanned System Technology

Self-tuning PID Parameters by using NN-GA for Cruise Control system

Jirapun Pongfai, Wudhichai Assawinchaichote

King Mongkut's University of Technology Thonburi, Bangkok, Thailand

Abstract— This paper considers the self-tuning PID parameters by using Neural Network (NN) together with Genetic Algorithm (GA) which is called the NN-GA. The NN-GA is the combination of Neural Network and Genetic Algorithm which is optimized the learning process of NN by using GA. From the simulation results, the NN-GA gives the better transient response i.e., the percent overshoot, the steady state error, the rise time and the settling time when compared with the pure NN, pure GA and PSO.

Keywords — PID controller; self-tuning; Cruise Control; Genetic algorithm; Neural Network; Particle and Swarm Optimization; transient response analysis; artificial intelligence; Fitness function calculation.

DOI: 10.21535/just.v6i1.1004

I. Introduction

NOW, the automobile industry applies the cruise control system into the vehicle [1]. The cruise control is a mode of the car which keeps the speed of car following the driver setting. This mode is called Velocity Control Mode (VCM) [2]. It helps the driver relaxing when they drive in the highway or long-distance road by maintenance the speed of car [1, 2, 3]. The cruise control operates by the difference the desired velocity and actual velocity for keeping the small error which comes from the slope of the road changing. The cruise control fails when the external disturbance such as gravitational force and wind resistance occur [1]. From the fail point, the control system should have the proper controller for improvement the performance and reliability of controlling. The controller which is popular in the industrial applications is Proportional-Integral-Derivative (PID) controller [4, 5, 6] because of simple structure, robustness and high reliability [4].

However, the PID controller has the 3 parameters which are KP, KI, and KD. The performance of PID controller depends on the 3 parameters. The 3 parameters value should proper with system. That means it needs parameters tuning process as appropriate with the system controlling. The tuning process is very difficult to achieve the proper value [5]. Especially in the nonlinear control system, PID parameters are difficult to tuning because the disturbance can occur in every time. Cruise control system is a control system which has an external disturbance, the PID tuning process is a problem of controlling. Recently, this problem of PID parameters tuning is solved by applying artificial intelligence algorithm into process of tuning. The tuning process is auto-tuning PID parameters which sometimes is called the self-tuning. The algorithms which are popular applying are GA, NN, Particle Swarm Optimization Algorithm (PSO), Artificial Bee Colony Algorithm (ABC), Fuzzy logic and etc.

In the literature, M. K. Rout, D. Sain, S. K. Swain and S. K. Mishra [6] purpose GA to tuning PID parameters for cruise control and compare the performance with conventional PID method, state space, and Fuzzy logic. The criteria for evaluation include maximum overshoot, rise time, peak time, settling time and steady state error. In the simulation result, GA gives the maximum overshoot, rise time, peak time, settling time and steady state error better than the conventional PID method, state space, and Fuzzy logic. Hua Ji and Zhiyong Li [4] present self-tuning PID parameter by using BP neural network. It is implemented for controlling the speed of brushless DC motor. MATLAB is used to simulation the model. They show the robustness improvement and adaptability of the model by comparison the simulation result between the purposed method and traditional PID controller.

This paper is organized as follows. Section II presents the cruise control modeling and the PID controller. Section III presents the artificial intelligence algorithm. In section IV presents NN-GA implementation. In section V shows the simulation results of self-tuning PID controller by NN-GA. Finally, a conclusion is given Section VI.

II. Cruise Control Modeling

A. Mathematic model of cruise control

The cruise control is the functional of car vehicle. It involves with car speed controlling. The objective of cruise control is maintenance the speed of the car which is set by the driver. The problem of cruise control comes from the external disturbance for example slop of road, gravitation force and etc. There effect to air and rolling resistance. The problem is solved by adding cascade controller and feed forward controller to cruise control [1].

$$G_{P(s)} = \frac{2.4767}{(s+0.0476)(s+1)(s+5)} \quad \textbf{(1)}$$

Corresponding author: Wudhichai Assawinchaichote
(e-mail: wudhichai.asa@kmutt.ac.th)

This paper was submitted on May 29, 2018; revised; and accepted on June 27, 2018.

This paper presents another method of solving this problem by auto adjustment the constant of parameters which are PID parameters controller. The cruise control system is represented by transfer function [1] which is presented in **(1)**.

B. *PID controller*

Figure 1 shows the block diagram of Cruise control with PID controller where $s(s)$ is the set point, $e(s)$ is the error signal, $u(s)$ is control input and $y(s)$ is actual output. For the controller part, we use PID controller since it has the simple structure, easy to implementation [4], robustness, high reliability, and easy to engineering [5]. From the system, the PID controller parameters come from auto-tuning which this paper presents Genetic and Neural Network Algorithm (NN-GA) for auto-tuning PID parameters.

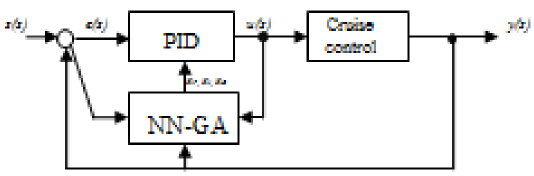

Figure 1 Block diagram of self-tuning PID parameter by using NN-GA for Cruise control system

The general form of PID controller is

$$u_{(s)} = \left(K_P + \frac{K_I}{s} + K_D s\right) e_{(s)} \tag{2}$$

where K_P is proportional gain, K_I is integral gain, and K_D is derivative gain.

III. The Artificial Intelligence Algorithm

A. *Neural Network*

NN is inspired by the human brain processing. The brain human has a complex and nonlinear of the processing system. It consists of neuron and interconnects with axons. The axons are used to transmit nerves impulse to the other neuron. The neuron connects to axons via dendrites. The contact point between dendrites and axons is synapse. Neurologists discover that the human brain learns by changing the strength of the synaptic connection between neuron [7]. The structure of the human brain is shown in Figure 2.

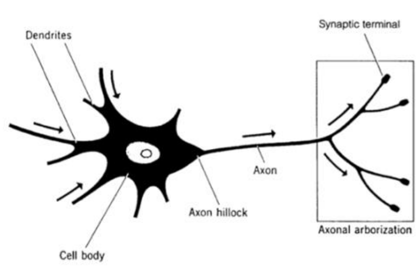

Figure 2 The structure of human brain [7]

The human brain structure is represented in conceptual of the graph. The neurons can be represented as nodes and interconnection as edges. An NN is a network structure which consists of nodes and directional links. A node is a processing unit. The link between nodes is the causal relationship between connected nodes. The output of nodes depends on the parameters pertaining to these nodes. The simple structure of the NN is shown in Figure 3.

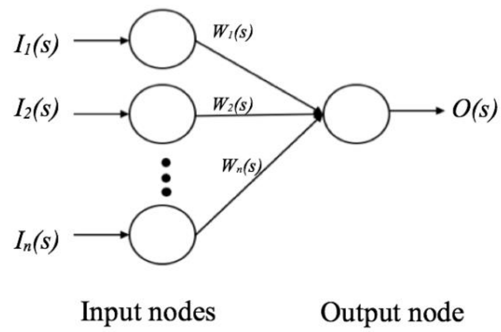

Figure 3 The simple structure of NN

From Figure 3, the simple structure of NN includes input node ($I_n(s)$), output node ($O(s)$) and edge between the input node and output node. The edge is represented by weigh which value come from the learning process.

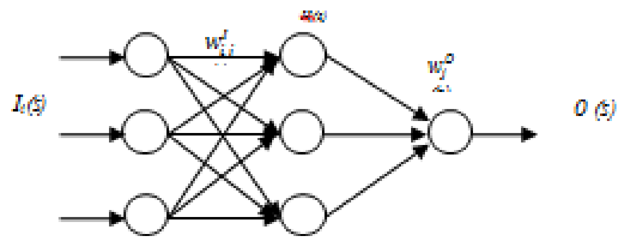

Figure 4 The simple structure of NN [7]

Figure 4 shows the multiple layer structure of NN. It is another structure of NN which is similar the perceptron. It increases the hidden nodes ($H_j(s)$) between the input nodes and the output nodes. The multiple layer structure of NN has more complexity than the perceptron. The mathematic modeling for this structure is shown in **(3)** and **(4)**.

$$H_{j(s)} = \sum_{i=1}^{n} I_i W_{ij}^{I} \tag{3}$$

$$O_{i(s)} = \sum_{i=1}^{n} H_i W_i^{O} \tag{4}$$

This paper uses the multiple layer structure of NN to apply the PID controller auto-adjustment by Genetic and Neural Network Algorithm for cruise control.

The learning processing of NN

The NN has the process of learning for output predictive the output. The learning process is called “training”. The training has two steps. Firstly, the network process is a set of the input value and output estimation. The second step is error calculation, it compares the input value and output value. It reflects the error to the network. The goal of learning is to determine a set of weight (w) that minimize the total sum of square error. It is referred to “error-correction learning”. This learning leads to the delta rule or Widrow-Hoff rule.

According to the delta rule, Let $W_{ij}(s)$ is the value of weight factor for neuron i excited by input $I_i(s)$ at the time step s. The adjustment $\Delta W_{ij}(s)$ is defined by

$$\Delta W_{ij(s)} = \eta e_{k(s)} I_{i(s)} \quad \textbf{(5)}$$

Where η is a learning rate constant which is the positive value. This parameter is the key of determine the performance of error correction learning. For update value of synaptic weight is determined by

$$W_{ij(s+1)} = W_{ij(s)} \cdot \Delta W_{ij(s)} \quad \textbf{(6)}$$

B. Genetic algorithm

The genetic algorithm is an adaptive heuristic search algorithm. It adapts the idea of engendering one population to new population from parent solution. The employing mechanisms inspired by Genetic and using natural selection together with mutation and crossover for survival evaluation of the fittest according to Darwin's theory [6].

The cycle of the GA which is represented in Figure 5. From Figure 4, it starts with the generating and initial population of the chromosome. In general, the initial population is randomly generating, then it goes to an iteration process to make the population evaluation. Each iteration consists of genetic operation and evaluation.

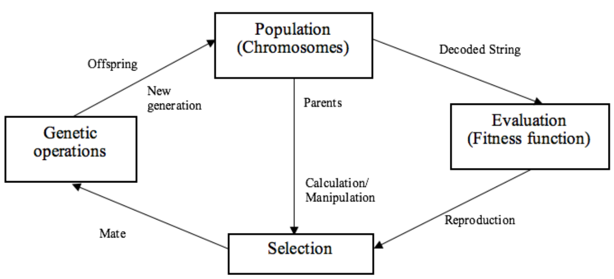

Figure 5 The simple structure of NN [7]

Population (chromosomes)

In the biological, the chromosome is built of Deoxy Ribo Nucleic Acid (DNA) [9]. It stores all the genetic information. The chromosome is divided many parts which is called "Genes". The genes have the allele which is the possibilities of the genes for one property for example genes for eye color and all possibilities alleles. The set of alleles can determine the different possible variation for the future generations of genes. For GA implementation, the chromosome is represented by the set of binary-bit string and each of it is genes.

In the biological, the chromosome is built of Deoxy Ribo Nucleic Acid (DNA) [9]. It stores all the genetic information. The chromosome is divided many parts which is called "Genes". The genes have the allele which is the possibilities of the genes for one property for example genes for eye color and all possibilities alleles. The set of alleles can determine the different possible variation for the future generations of genes. For GA implementation, the chromosome is represented by the set of binary-bit string and each of it is genes.

Genetic operation

The operation of the GA consists of selection method, crossover, and mutation. All methods of the GA are inspired by natural operation in the origin of species which is the theory of Charles Darwin.

1) Selection method [9] is the process of choosing two parents from the populations for crossing. The method of choosing is randomly the chromosome then evaluate fitness function and choose the chromosomes two chromosomes which have higher fitness. The techniques of selection method include:
   - Roulette wheel selection
   - Random selection
   - Rank selection
   - Tournament selection
   - Boltzmann selection
   - Stochastic Universal sampling

   The selection method for this research uses the roulette wheel selection because it's common to use reproduction operation. The roulette wheel is a linear search. The final value is a set of randomly proportion the sum of fitness in the population. The population is updated until toward final value. The evolution of roulette wheel selection depends on the variety of fitness in the population. The selection is used in this paper which is shown in Figure 6.
2) Crossover is a recombination operation. The first step of crossover is randomly two chromosomes in the population then random the bit and swapping all the bits after pointing between two chromosomes. The step of crossover is represented in Figure 7.
3) Mutation operation is changing the bit within the chromosomes. The process of changing is randomly a chromosome then random the position for flipping "0" to "1" or "1" to "0". The step of crossover is represented in Figure 8.

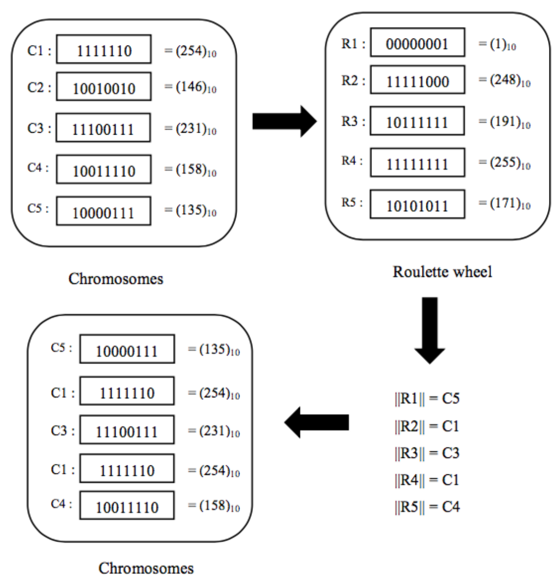

Figure 6 The method of the roulette wheel

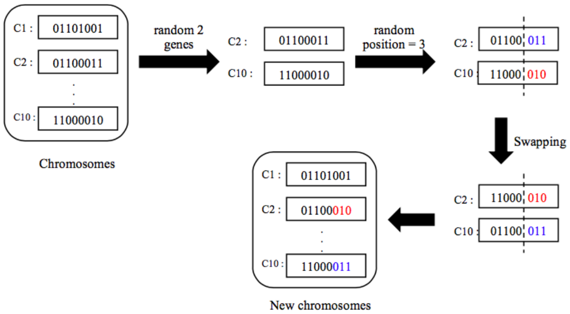

Figure 7 Crossover operation

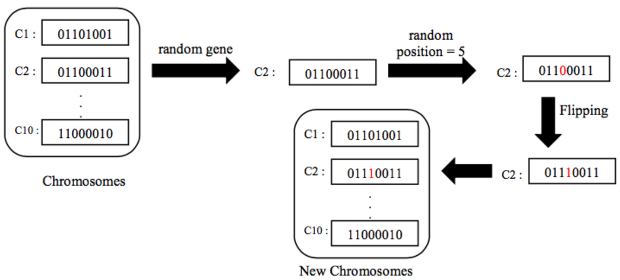

Figure 8 Mutation operation

C. NN-GA

The structure of NN-GA is similarly the structure of NN which is shown in Figure 3. The difference between NN-GA and NN is weigh adjustment. The weight of NN-GA comes from the process of GA. From Figure 3, input nodes of NN-GA are $e(s)$, $u(s)$ and $y(s)$. Hidden nodes of NN-GA have 3 nodes and output nodes of NN-GA are K_P, K_I, and K_D. The process of GA in NN-GA is executed following Figure 4. It starts when NN requests the $w_{ij}^I(s)$ and $w_j^O(s)$. From Figure 4, the process of initial population is executed when the generation is 0. Each time of request from NN, the generation is increased 1 generation. The GA processes until termination checking, it sent the $w_{ij}^I(s)$ and $w_j^O(s)$ to NN then NN operands follow **(3)** and **(4)**. The generation is reset when the setpoint is changed or the system is reset.

The GA implementation in this paper uses 8 bits string for represent the size of the chromosome. The population size of each $w_{ij}^I(s)$ and $w_j^O(s)$ of NN is 40. The probability for crossover is 0.8 and probability for mutation is 0.01. The generation value for the terminate process is 100 generations and the condition for checking is shown in **(7)**.

$$J = \frac{1}{2}\sum_{i=1}^{N} e_{(s)}^2 \qquad \textbf{(7)}$$

where

J : the fitness function value

$e(s)$: error signal of the system

IV. NN-GA Implementation

The NN-GA algorithm is a combination of NN algorithm and GA by using the GA to optimize learning process of NN. That means the weight of NN is auto adjusted by GA. It is implemented by using MATLAB programming (m-script).

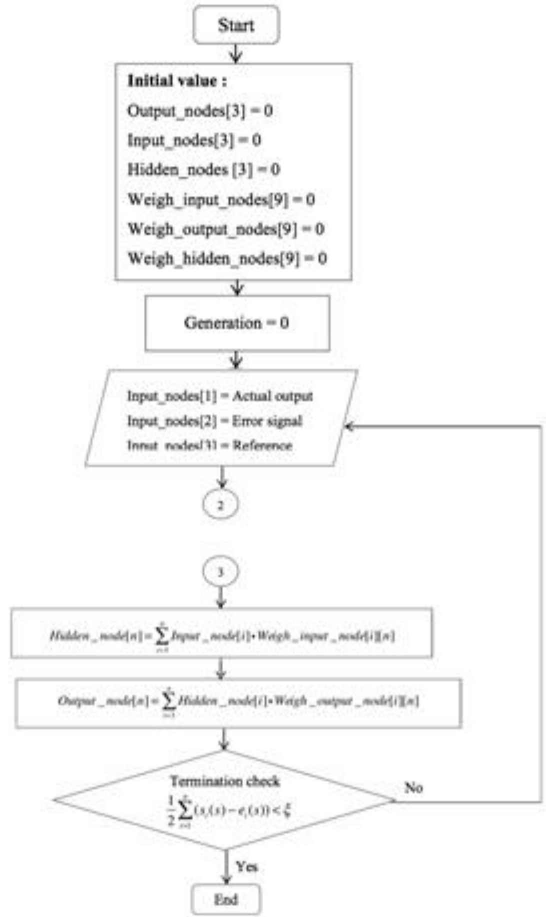

Figure 9 The flow chart of NN-GA (NN part)

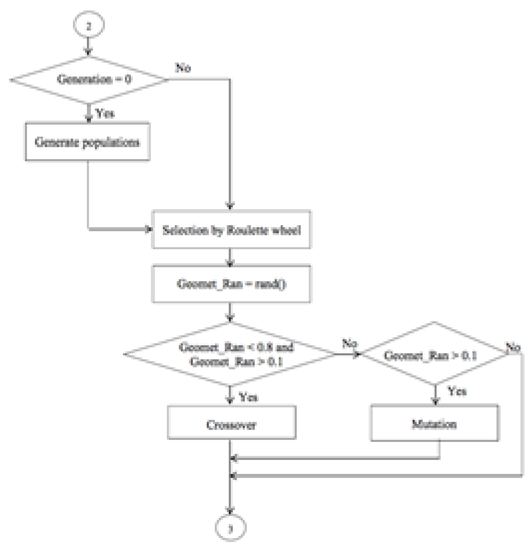

Figure 10 The flow chart of NN-GA (GA part)

The flow of this algorithm which shows in Figure 9 and Figure 10 is separated into 2 parts including part of the NN as shown in Figure 9, part of GA as shown in Figure 10. The part of the NN is for preparing the variables for the operation in both NN and GA. In operation of NN, it is similar the normal NN but the process of random weight is changed to call the GA function. Another part is the GA. This part is similar the normal GA but it removes the process of termination checking because this process is done in the process of NN.

V. Simulation Result

In this simulation, the speed control for cruise control is considered. The set point of the simulation is 1. The algorithms for comparative performance with NN-GA are pure GA, pure NN, and PSO. The criteria of evaluation performance which follows transient response analysis which includes the acceptable maximum overshoot ±0.4%, the steady state error ±1%, the rise time less than 20 ms and the settling time less than 40 ms. The simulation result is shown in Figure 11-Figure 14, Table 1 and Table 2.

A. The transient response analysis (Exclude noise)

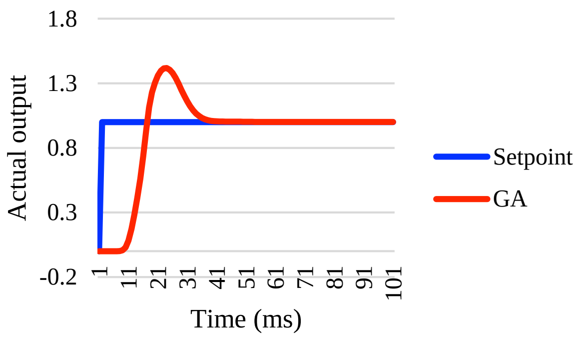

Figure 11 The step response of Brush DC motor by using GA

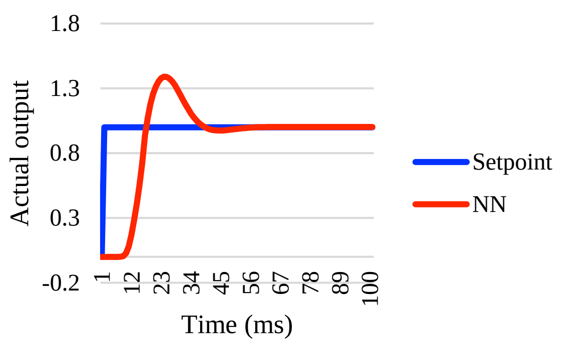

Figure 12 The step response of Brush DC motor by using NN

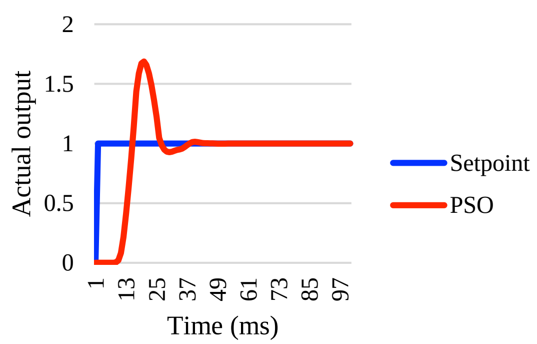

Figure 13 The step response of Brush DC motor by using PSO

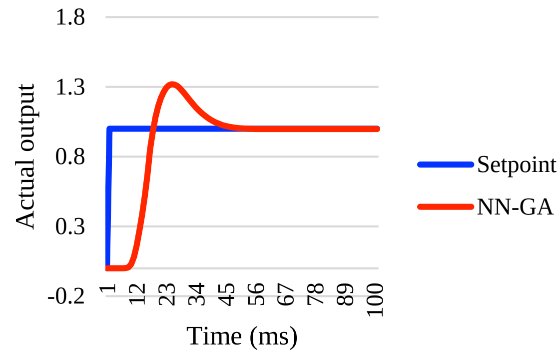

Figure 14 The step response of Brush DC motor by using NN-GA

Table 1 PID parameters with set point 1000 rpm for brush DC motor

Tuning algorithm	K_P	K_I	K_D
GA	5.9	2.7	3.0
NN	8.0	5.1	5.0
PSO	23.20	37.55	10.0
NN-GA	6.1	2.6	3.8

Table 2 The result of comparative performance in transient response analysis

Tuning algorithm	GA	NN	PSO	NN-GA
Maximum overshoot (%)	0.417	0.388	0.687	0.319
Steady state error (%)	0	0	0	0
Rise time (ms)	16	16	15	17
Settling time (ms)	32	34	37	36

In case of the cruise control noise exclusion, we found that GA is given 0.417% for maximum overshoot, 0% for steady state error, 16 ms for rise time and 32 ms for settling time. The transient response of GA is shown in Figure 11. NN is given 0.388% for maximum overshoot, 0% for steady state error, 16 ms for rise time and 34 ms for settling time. The transient response of NN is shown in Figure 12. PSO is given 0.687% for maximum overshoot, 0% for steady state error, 15 ms for rise time and 37 ms for settling time. The transient response of PSO is shown in Figure 13. NN-GA is given 0.319% for maximum overshoot, 0% for steady state error, 17 ms for rise time and 36 ms for settling time. The transient response of NN-GA is shown in Figure 14.

The Table 1 shows the PID parameters for cruise control in case of noise exclusion which comes from the difference algorithm. From the simulation, we found that each algorithm gives the difference PID parameter value. GA gives 5.9 for K_P, 2.7 for K_I and 3.0 for K_D. NN gives 8.0 for K_P, 5.1 for K_I and 5.0 for K_D. PSO gives 23.20 for K_P, 37.55 for K_I and 10.0 for K_D. NN-GA GA gives 6.1 for K_P, 2.6 for K_I and 3.8 for K_D.

B. The fitness function evaluation (Exclude noise)

In the self-tuning process, it has another criterion of self-tuning. This is fitness function calculation in each of time of tuning. From the transient response analysis, we found that each algorithm gives the similar PID parameters. But the time of tuning is the difference. The fitness function criteria are evaluated following the (7) which fitness function value for the end of tuning is 0.1% of setpoint. The fitness function of GA is shown in Figure 15, Figure 16 for NN, and Figure 17 for PSO.

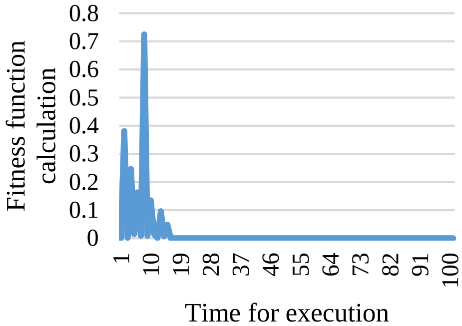

Figure 15 The fitness function evaluation by using GA

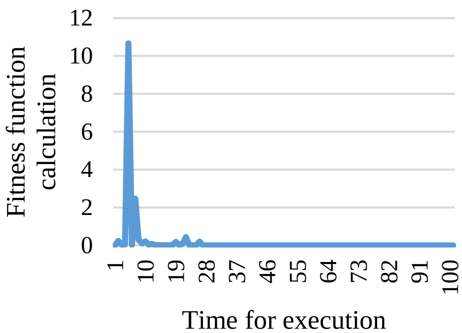

Figure 16 The fitness function evaluation by using NN

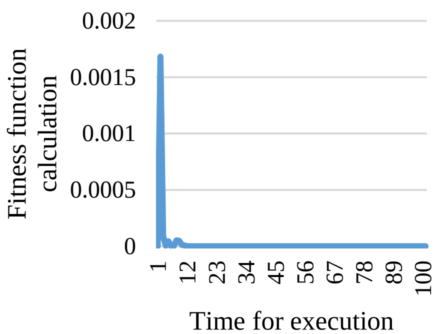

Figure 17 The fitness function evaluation by using PSO

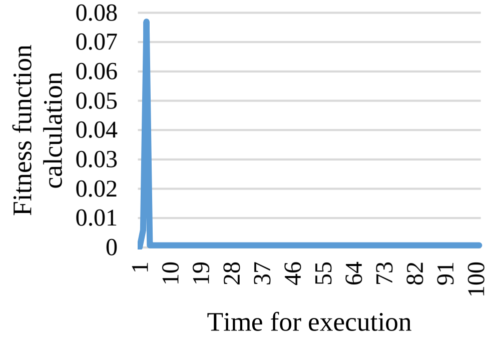

Figure 18 The fitness function evaluation by using NN-GA

From the fitness function calculation in each algorithm, GA uses 16 times for self-tuning PID parameter which is shown in Figure 15. NN uses 25 times for self-tuning PID parameter which is shown in Figure 16. PSO uses 11 times for self-tuning PID parameter which is shown in Figure 17. NN-GA uses 3 times for self-tuning PID parameter which is shown in Figure 18.

C. The fitness function evaluation (Exclude noise)

The Figure 19 shows the noise which increases into the cruise control system in each algorithm. The noise is the band-limited white noise which is the block in the Simulink block. The noise setting is 1 for noise power, 20 ms for sample time and 100 for speed setting.

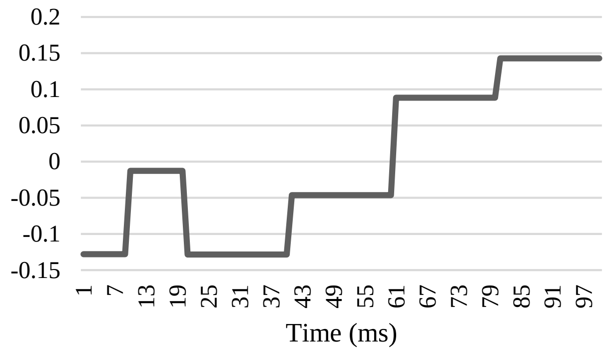

Figure 19 The noise for GA robustness evaluation

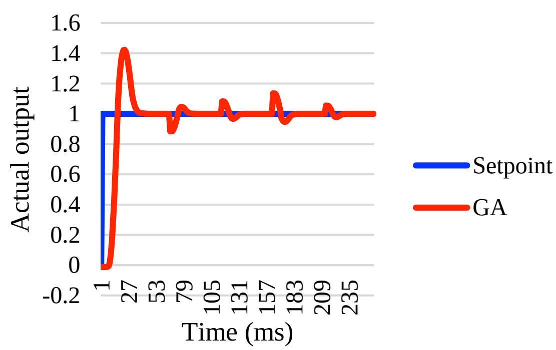

Figure 20 The step response of Brush DC motor by using GA

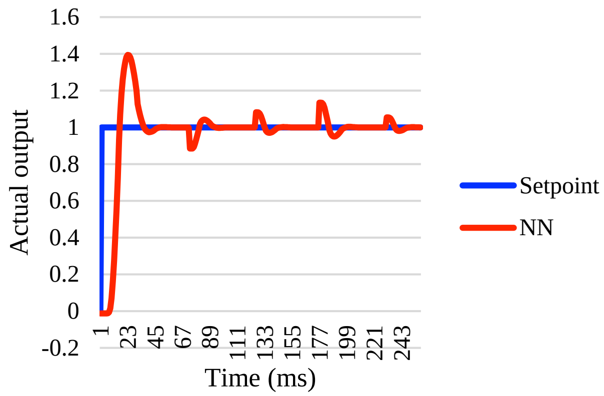

Figure 21 The step response of Brush DC motor by using NN

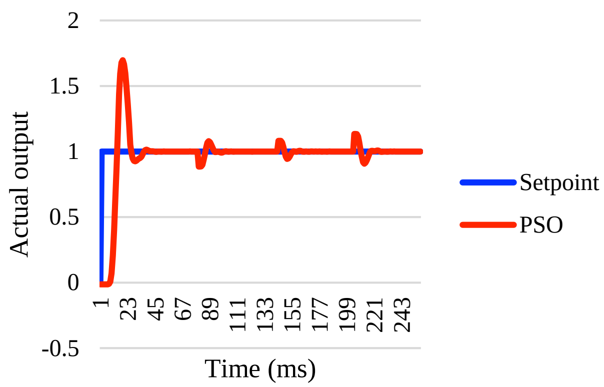

Figure 22 The step response of Brush DC motor by using PSO

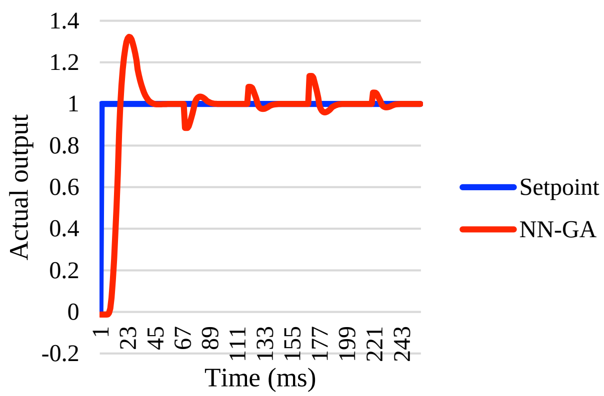

Figure 23 The step response of Brush DC motor by using NN-GA

Figure 20-Figure 23 show the simulation result of the system which is added noise to the system. From the simulation result, we can see that all algorithms can quickly resume toward the steady state. That means the control result in all algorithm is robustness performance.

VI. Conclusion

From the simulation result, GA and PSO give the maximum overshoot exceed the criteria. All algorithms provide rise time, settling time and steady state error following the criteria. From fitness function calculation, GA, NN, and PSO take time and fitness function for self-tuning PID parameters more than NN-GA. That means the self-tuning PID parameters based on Cruise Control motor by using NN-GA provides the better performance than the pure GA, the pure NN, and PSO. In case of robustness improvement, all algorithms are robustness. In conclusion, the NN-GA has high effectiveness, provides the good performance and robustness for tuning PID parameters based on Cruise control.

Acknowledgment

The authors would like to thanks, Faculty of Engineering at King Mongkut’s University of Technology Thonburi for supporting the funding of this researching.

References

[1] Khairuddin Osman, Mohd. Fuaad Rahmat and Mohd Ashraf Ahmad, "Modelling and Controller Design for a Cruise Control System", 2009 5th International Colloquium on Signal Processing & Its Application, March 2009.

[2] Mahyar Vajedi and Nasser L. Azad, "Ecological Adaptive Cruise Controller for Plug-In Hybrid Electric Vehicles Using Nonlinear Model Predictive Control", IEEE TRANSACTION ON INTELLIGENT TRANSPORTATION SYSTEMS, vol. 17, Issue. 1, January 2016.

[3] D. Sain, S.K. Swain and S.K. Mishra M.K. Rout, "PID Controller Design for Cruise Control System using Genetic algorithm”, International Conference on Electrical, Electronics and Optimization Techniques, November 2016.

[4] Hua Ji and Zhiyong Li, "Design of Neural Network PID Controller Based on Brushless DC motor," IEEE, October 2009.

[5] Xufei Dai, Zhili Long and Jianguo Zhang, “PSO based on chaotic Map and Its Application to PID Controller Self-tuning”, 2015 16th International Conference on Electronic Packaging Technology, August, 2015.

[6] PANG-NING TAN, MICHAEL STEINBACH and VIPIN KUMAR (2006). INTRODUCTION TO DATA MINING. United state: Pearson Education, Inc.

[7] Michael A. Arbib (2003). The Handbook of Brain Theory and Neural Network, 2 nd ed. United state: Impressions Book and Journal Services, Inc.

[8] Hua-yun CAO and Fu-ming PENG, "Optimization of engine speed neural network PID controller based on genetic algorithm”, IEEE, November 2011.

[9] MEHMED KANTARDZIC (2001). DATA MINING Concepts, Models, Methods and Algorithm. new jersey: IEEE Press.