Design of Intelligent Control System Based on General Regression Neural Network Algorithm

Alvin Sahroni

***Abstract*—this study discuss about the General Regression Neural Network (GRNN) implementation for control system application. Nowadays, hybrid theory tends to investigate within control system research. Previously, many kinds of neural network control schemes have been deployed. The problem was related to the optimization in training phase to satisfy the response system result based on plant's identification phase. GRNN was known as one passing learning algorithm with a highly parallel structure. The algorithmic form can be used for any regression problem in which the assumption of linearity data is not satisfied related on performance problem. This investigation using GRNN as the online dynamics learning model can be use as predictor or estimator for control signal with the performance reaches 99% to prove that GRNN is reliable as one of modern Intelligent Control System.**

***Key Words*—GRNN, online learning, Neural Network, Hybrid Theory, Speed Controller, CSTR Controller**

I. Introduction

In the past two decades, many adaptive control method have been developed, such as self-tuning PID controller, auto-tuned PID, etc. have been developed and successfully implemented in many applications [11][1]. More recently new generation of adaptive control system which uses artificial intelligence techniques are developed to deal with increasing complexity in control systems such as non-linearity, unexpected load diturbances, variable time delay, etc [3].

Artificial Neural Network (ANN) is one of intelligent control system which well known in control system application. Its capability in learning and provide generalization of signal are reliable as control method in industry or research area. ANN is an artificial intelligence methodology or certain people say as supervised machine learning that applicable to solve many complex non-linear problems than using conventional methods such as PID control system. Using this method, we can obtain the relationship between input system and output system without having to derive the physical equations of the system which knows with the term of identification system [4].

Manuscript received November 26, 2012. This work was supported by Electrical Engineering Department of UII Yogyakarta, Indonesia. Independent Research Project in Control and Microprocessor Laboratory, UII Indonesia.

Alvin Sahroni is with the Department of Electrical Engineering, UII Yogyakarta Indonesia, Kaliurang Street 14.5 (alvinsahroni@uii.ac.id)

DOI: 10.5176/2251-3043_2.4.215

There are many types of ANN architectures are used in control system. The common architecture is using Multilayer Perceptron Architecture that well known as back-propagation neural network. Back-propagation has many field for implementation, otherwise back-propagation also has several shortages which are solved with certain hybrid method to optimize its performance. Another thing that should be considerate using conventional back-propagation related to the lack of training time resulting method is not too good for machine learning implementation.

In many neural-control paradigms the back-propagation method, the gradient descent method also give another lack of performance related to slow convergence and local minimum. To improve the neural-control, General Regression Neural Network (GRNN) was proposed as predictor in neural-fuzzy controller (NFC) to provide automatic tuning of the NFC based on an online prediction model of the plant provided by GRNN [6].

This study is to provide another investigation using GRNN as online learning of neural network and for control system application. The goal was to decrease the training time to design the control system, and to provide online learning as the predictor. In the experiment, it's using DC Motor and Continuous Stirrer Tank Reactor (CSTR) as the plants. DC motor has been used in daily lives such as household, industries, and also education field. The needs related to the precision and quality between set value and process value performance. Previous investigation was done with fuzzy logic controller and back-propagation neural controller. The investigation result was giving 1.6% - 10 % of error [10].

CSTR controller was used in many chemical process applications. For certain chemical process is needed to gain stable concentration with flow rate as the actuator. The previous experiment have been develop by using neural network predictor controller using back-propagation method but this scheme gives a lot of overshoot.

II. Preliminary Study

A. GRNN Modeling

GRNN has been proposed [8] and the using of GRNN has been popular than back-propagation because the characteristic of GRNN as probabilistic solution and the known as probabilistic neural network then GRNN was the alternative feed-forward method after back-propagation in many fields.

The concept is the regression function performed on a independent variable such as X computes the most probable value of dependent variable Y within finite set of observations data. Lets $f(x)$ as the probability density function of a random vector x. Regression result of Y given by X based on below formula:

$$E[y/X] = \frac{\int_{-\infty}^{\infty} y.f(X,y)dy}{\int_{-\infty}^{\infty} f(X,y)dy} \tag{1}$$

The probability density function must be estimated from sample observations (data sets) of x and y. The general form of estimation process given by:

$$f_n(x) = \frac{1}{n\lambda}\sum_{i=1}^{n}\varphi\left(\frac{x-x_i}{\sigma}\right) \tag{2}$$

Where x_i are independent and as distributed random variables with absolutely continuous distribution function. The φ have constrains to bounds certain value to satisfy several conditions:

$$\int_{-\infty}^{+\infty} |\varphi(y)|\, dy < \infty \tag{3}$$

$$\lim_{y\to\infty} |y\varphi(y)| = 0 \tag{4}$$

And

$$\int_{-\infty}^{+\infty} \varphi(y)dy = 1 \tag{5}$$

The function $\sigma = \sigma(n)$ must be chosen with:

$$\lim_{n\to\infty} \sigma(n) = 0 \tag{6}$$

$$\lim_{n\to\infty} n\sigma^2(n) = \infty \tag{7}$$

One useful form of the weighting function φ is the kernel density function which is known as Gaussian function.

There is a constant value for distribution estimator that based on panzer's results that will converge to the underlying distribution at the sample point when it is smooth and continuous. Based upon sample values X^i and Y^i of the random variable x and y, it can provide a good distribution constant value as given by [9]:

$$\hat{f}(X,Y) = \frac{1}{(2\pi)^{(p+1)/2}\sigma^{(p+1)}}. \frac{1}{n}\sum_{i=1}^{n} e^{\left[-\frac{(X-X^i)^T(X-X^i)}{2\sigma^2}\right]} e^{\left[-\frac{(Y-Y^i)^2}{2\sigma^2}\right]} \tag{8}$$

p is the dimension of the vector variable x, n is the number of observations, σ is the width spreading of estimated kernel or smoothing factor, and Y^i is the desired or target data given by the observed input data of X^i. Then define D_i^2 by using equation:

$$D_i^2 = (X - X^i)^T (X - X^i) \tag{9}$$

Combining (8) and (9) and interchanging the order of integration and summation and given by:

$$\hat{Y}(X) = \frac{\sum_{i=1}^{N} Y^i e^{\left(-\frac{D_i^2}{2\sigma^2}\right)}}{\sum_{i=1}^{N} e^{\left(-\frac{D_i^2}{2\varsigma^2}\right)}} \tag{10}$$

The $\hat{Y}(X)$ result can be used as weighted average of all observed data of Y^i and each observed data is based on weighted exponentially of related to Euclidean distance from X.

The schemes of GRNN architecture is given on Fig.1. It contains several steps to be done to gain the result of GRNN estimation.

B. Close Loop Control System

In the control system, there are two types of control scheme. There are open-loop scheme and close-loop scheme. Generally, close-loop scheme has been entrusted to improve control accuracy by comparing process value with desired value entered by the operator. The basic scheme of close-loop control system was given in Fig.2. There are four main parts to forms close-loop scheme. There are desired input, comparator, controller, plant, process value, and sensor. Basic forms of overall system is given by:

$$\frac{Y(s)}{X(s)} = \frac{G(s).H(s)}{1+T(s).G(s).H(s)} \tag{11}$$

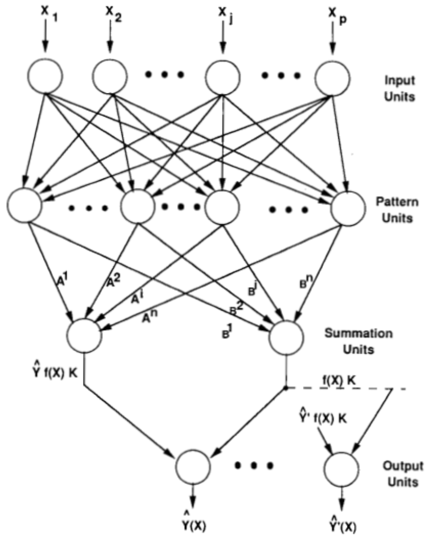


Fig.1 GRNN Architecture (Specht,1991)

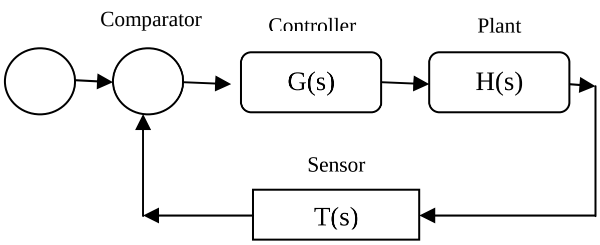


Fig.2 Close-loop Scheme

The main principal of close-loop scheme is robustness to uncertainty condition through feedback scheme. By doing this concept, it allows high performance in the presence of uncertainty and also accurately sensing the surrounding conditions to compare to desired input and doing correction through computation and actuation. The next principal is the design of dynamics through feedback. It means that the development allows the dynamics (behaviour) of the system to be modified.

Otherwise, the improvement of close-loop scheme has limitations. The main problem in close-loop is to handle the disturbance error to gain robust controller. We cannot deny that the real conditions has many uncertainty aspects that becomes the disturbance for the control system. Based on that condition, the robust controller should be adapt by doing modification and optimalization within close-loop scheme interfered with another method such as intelligent control system.

C. *Intelligent Control System*

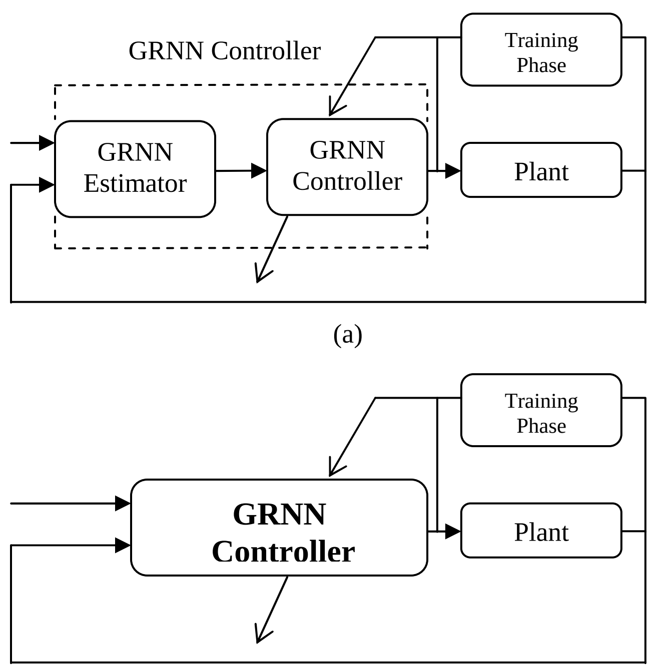
Intelligent control system has wide range of combinations method. The aim is to get adaptive system that can improve control system response. For the creation of intelligent control system means to develop certain algorithm which use the principal of artificial intelligence based on control system theory. The algorithm itself should be satisfying three following conditions [2].

1) There is natural object
2) There is an aim to create its duplicate
3) There is possible way for the realization of the supposed aim.

Using GRNN as control system has similar scheme with back-propagation control system. Intelligent control system should be trained before it can be implemented into the plant system. Intelligent control, system has not to be modeled rigidly. The important part that should be done by the control designer is to give and observe certain initialization input and evaluate it on its output.

III. Proposed Method

The propose method using two types of plants for GRNN control system implementation. The GRNN controller will applied for DC motor which is have been used widely in terms of industrial applications. This study used DC Machine Simulink model in MATLAB. With certain specifications such as 5 HP, 240V, 1750 RPM, Field Voltage 150 V. Another investigation is by doing it into CSTR plant to provide alternative type of control system application. CSTR is used in chemical processes which control the concentration within substance by flow rate actuator. GRNN itself will contain several schemes, for Speed DC Motor controller will used GRNN as estimator and GRNN as controller that united become GRNN Controller as seen on Fig. 3.



(b)

Fig.3 GRNN Control System Scheme, (a) for Speed DC Motor Controller, (b) CSTR Controller

For CSTR plant the scheme has a little difference that just using one GRNN as the controller. To implement the GRNN controller, there are several steps that should be done (Fig. 5).

Step 1. Observe open loop scheme to obtain training data set by doing open-loop scheme. The purpose is to get the plant characteristics based on the input and output data. Open-loop scheme was used to collect training data set. In developing intelligent control system based on neural network, it should generate random input as the initialization value to get system characteristics based on its output value (Fig.6). The next step is using the data for the training phase that will implement inverse model scheme (Fig. 4).

Step 2. Train both GRNN estimator and Controller with Inverse model technique (Fig. 4). After obtaining observation data, it should be trained using inverse model of ANN both as estimator and controller for neural network. Fig. 4 had shown the example of inverse model application. For speed controller will use two GRNN that act as an estimator and the other act as a predictor. The GRNN as an estimator has error state as input and desired input as the target. GRNN as an estimator will provide online training phase within the system to estimate incoming control signal to the GRNN controller. For CSTR controller, the input signal is the error state of the process plant, and the actuator is the flow rate. For each part of plants using equation (9) (10) will produce actuator signal for the plant system.

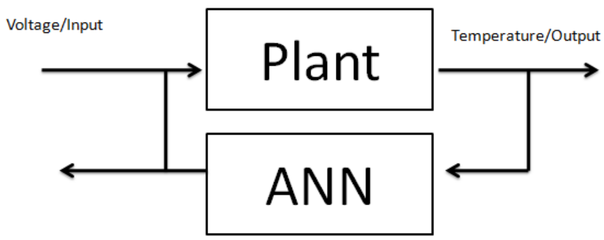


Fig.4 Example of Inverse Model (Alvin, 2012)

Step 3. Validation of GRNN is a phase to validate certain possibility of input that compared with target value to obtain MSE performance of GRNN. The performance will decide the best architecture for feed-forward scheme within close-loop scheme control system.

Step 4. Implement feed-forward algorithm using GRNN architecture that have been trained. The feed-forward controller was design in previous step. For the Speed controller, the input was the speed value, and the output is voltage within feed-forward scheme. For CSTR controller, the input is concentration value, and the output is flow rate actuation within feed-forward scheme.

Step 5. Evaluate the system response, and optimize GRNN Controller with another training data set. In the application of control system, evaluation of system response is needed as a basis of optimization in control system. Basically there are several parameters that should be evaluated to gain best performance of control system such as error signal, rise time, settling time, overshoot, etc. The optimum condition can be obtained if getting minimum value of the parameters.

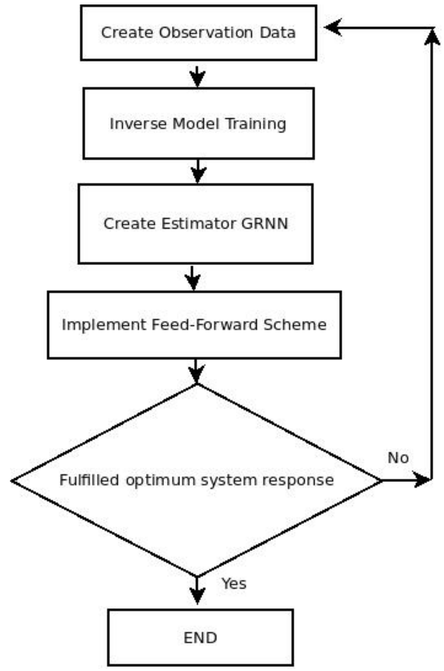
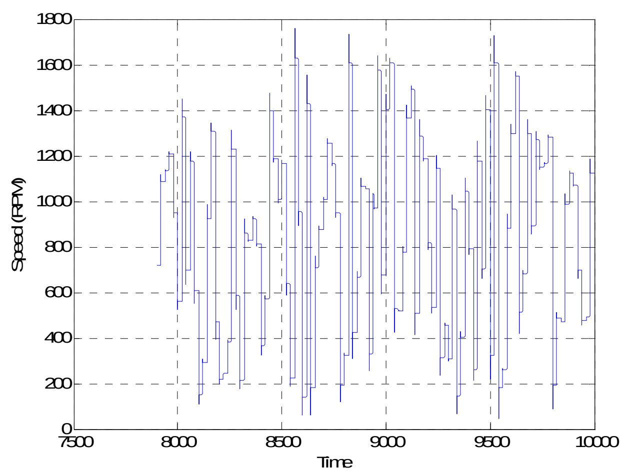
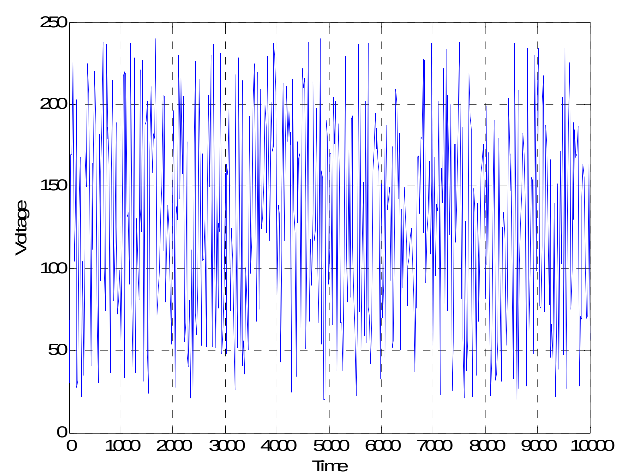


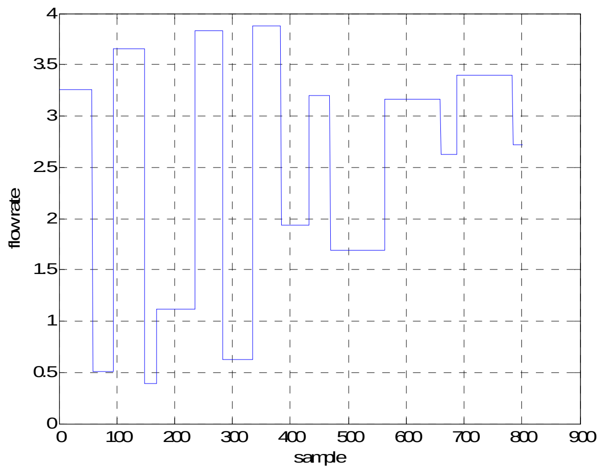
Fig.5 Flow Chart Diagram



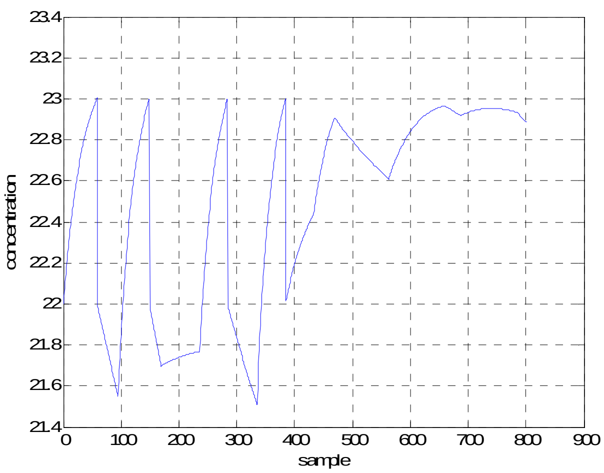
(a)



(b)



(c)



(d)

Fig.6 (a) Speed vs Time, (b) Voltage vs Time, (c) flow rate vs sample, (d) concentration vs sample

IV. Result, And Discussions

A. Training Result

After the observation phase, we should train the neural network using the inverse modeling scheme. GRNN has parameter of distribution that should be optimized in order to get a good approximation results which called “Spread Value”. The value is between 0-1. Both speed and CSTR controller still using offline training phase that should be optimized using trial and error method to find best spread distribution value (Table.1 and Table. 2) that indicated by using MSE (Mean Square Error) between target and simulation data (eq. 12). Especially in speed controller, GRNN has online training scheme that acted as estimator for the controller.

$$MSE = \frac{\sum_{i=1}^{n} error_n^2}{n_{error}} \qquad (12)$$

Table 1. Sequence to Find Best Distribution Value For Speed Controller Using GRNN

No	Distribution Value	Performance
1	0.1	3.6683e-004
2	0.3	0.0100
3	0.7	0.0542
4	0.9	0.0663
5	1	0.0703

Table 2. Sequence to Find Best Distribution Value For CSTR Controller Using GRNN

No	Distribution Value	Performance
1	0.3	6.8035e-008
2	0.7	0.0144
3	0.9	0.0307
4	1	0.0307

The best performance was obtained by using 0.1 distribution value that gives performance $3.888x10^{-4}$ for speed controller and $6.8x10^{-8}$ for CSTR controller. The best sequence will be used as the best architecture of GRNN for feed-forward scheme of control system.

B. Implementation Result

This section will show the implementation result of DC Motor speed controller and CSTR controller using GRNN Scheme. Before the implementation of close-loop scheme, the GRNN architecture for controller should be validated. The result of validation process that giving the best performances will be used for feed-forward scheme for close-loop control system (Fig.2 and Fig.4). Based on the investigation, the MSE is less than 10^{-3} compared with the target. Based on previous research, the best performance validation process of neural network should not greater than 10^{-2} [7]. Based on the reference, we can use GRNN architecture for the feed-forward scheme of close-loop control system.

Several of desired value was used to analyze the response of the plant. Firstly, desired input was generated by giving uniform random number of speed DC Motor between 450 – 1300 Rpm (Table.3, Fig.7). The second investigation using another uniform random number of desired input values (Fig.8). For the CSTR controller, GRNN applied concentration value 21 that obtain a good response without overshoot (Table.4, Fig.10). Based on the investigation we can see that the GRNN Controller that based on GRNN

Estimator and GRNN Predictor controller can give a good response system, where the value of the process can response quickly to changes related to the desired input value.

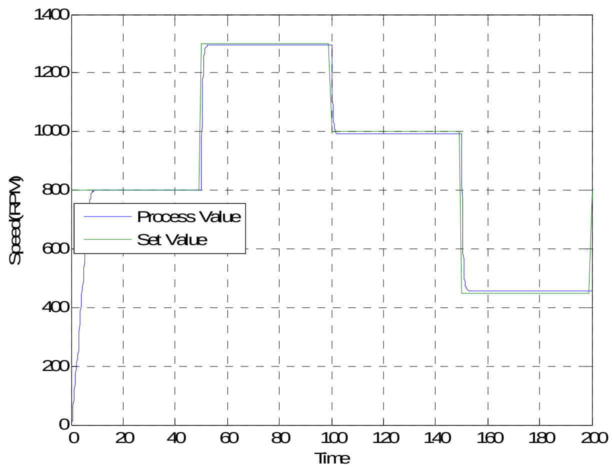


Fig.7 Implementation with 1st scenario of desired input

TABLE 3. SCENARIO AND RESULT FOR IMPLEMENTATION (SPEED CONTROLLER USING GRNN)

No	Desired Input	Error (Process-Desired Value)
1	250 Rpm	23.16 Rpm
2	500 Rpm	0.6 Rpm
3	800 Rpm	1.096 Rpm
4	1000 Rpm	0.2144 Rpm
5	1300 Rpm	0.2824 Rpm
6	1400 Rpm	0.06 Rpm
7	1700 Rpm	35.8 Rpm

TABLE 4. SCENARIO AND RESULT FOR IMPLEMENTATION (CSTR CONTROLLER USING GRNN)

No	Desired Input (Concentration)	Error (Process-Desired Value)
1	21.5	0.5
2	23	0.49

C. Evaluation

Feed-forward controller that used as the implementation for the best GRNN Architecture that will be using as the actuator signal within the plant system should be evaluated based on certain control system response characteristics. Several investigations were done and give the reliable result for implementation of GRNN as control system. GRNN could response the desired value quickly. Both of Speed DC motor and CSTR controller gives none of overshoot value based on the system response. For the final investigation, GRNN was tested by giving certain of step input and then define the response characteristics. Based on the result we conclude, the GRNN can control the speed od DC motor with the error: 1.096 Rpm, Rise Time: 2.0799 seconds, Settling Time: 7.8 seconds, and overshoot: 0% (Fig.9). Compared with standardization of control system response characteristic, it shown that GRNN performance has meet the minimum of

control system response standardization. For the common configuration for control system, the overshoot percentage should less than 5% and the error of the system is 0.137% or the performance is about 99% makes the GRNN is a reliable intelligent control system. The evaluation result gives a reliable method in intelligent control system based on GRNN Architecture as estimator and predictor. Another investigation also gives a good result, based on the previous research, CSTR controller has an overshoot value using back-propagation neural network. Using GRNN could be minimize the overshoot under 5% of it's standardization in control system response. The error value that based on value desired value compared with process value gives 0.3-0.4 of concentration value and it's still can be improved by doing another investigation related to the spread value optimization and data set. But, both of them, still need to be optimized with another heuristic optimization method such as genetic algorithm (GA), Particle Swarm Optimization (PSO), or another swarm intelligence approach.

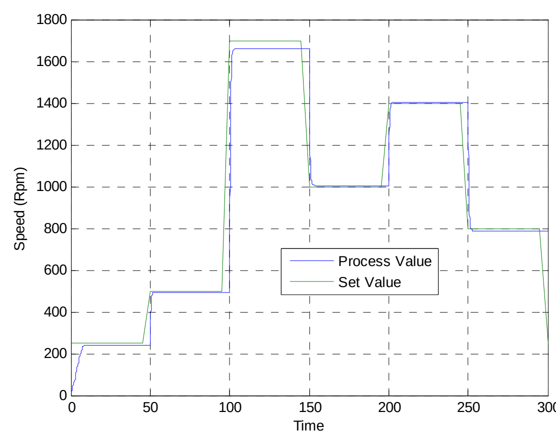


Fig.8 Implementation 2nd scenario desired input

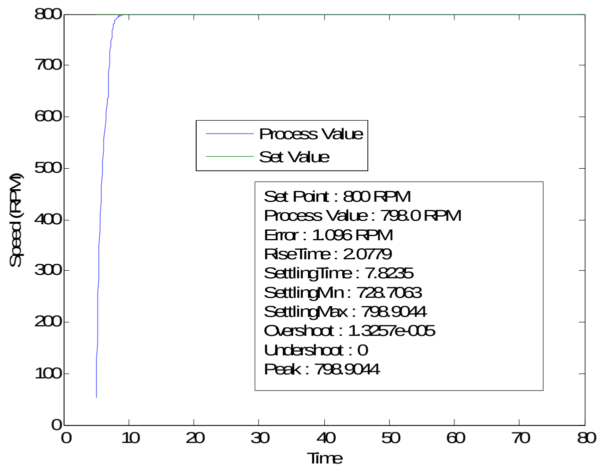


Fig. 9 System Response Characteristic at 800 Rpm

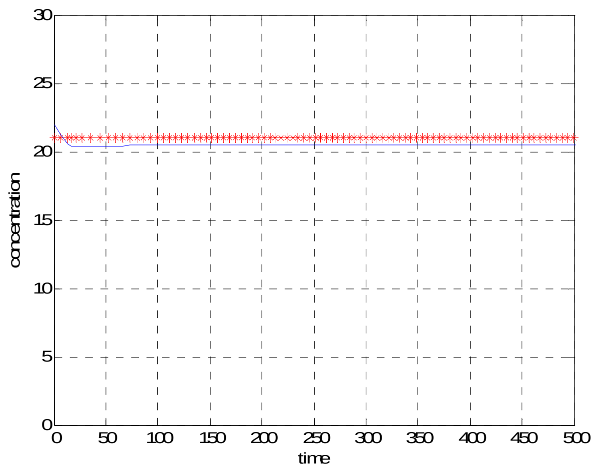


Fig.10 System Response CSTR Controller

D. *Other Recommendations*

For the future investigation, there should development for online learning both the estimator and controller. Since the GRNN has fast learner type of neural network, it can utilize certain optimization techniques within the GRNN scheme controller. Another recommendation related to the type of plant system. There should be another investigation using low step of control system such furnace or boiler to investigate the performance of GRNN Controller.

V. Conclusions

Based on the result investigation, it shows that GRNN has reliability for control system application. GRNN as a fast learner type of neural network algorithm has several limitations. The limitation related to black box model that offered by GRNN that should learned within the process running. GRNN is fast learner type that gives best result if we used online scheme of GRNN. GRNN is not reliable for offline learning, so then we should develop optimization for online learning in the future investigation.

To get a better performance, we should tune the spread value between 0-1. The smaller value will gives a best fit approximation, but the higher value will gives smooth performance of approximation.

This study shows that GRNN Control system is reliable and gives good system response characteristics in controlling speed of DC Motor and CSTR controller in chemical process. The error of the system related to desired and process value is about 0.137 %. It gives a good plan for future investigation to replace conventional intelligent control system based on neural network. It can reduce effectively for training time and it is possible to implement online intelligent control system based on GRNN architecture. The performances of GRNN controller reach greater than 95% or almost reach 99%. Finally, this study has proved the combination of GRNN Estimator as online learning neural network and GRNN predictor controller as feed-forward neural network provide reliable intelligent control system.

References

[1] Clarke, D. (1989). Properties of Generalized Predictive Control. *Automatica*, 859-875.

[2] Frankovic , B., & Oravec, V. (2005). Design of the Agent-Based Intelligent Control System. *Sign Language in the Intelligence Sensory Environment*, 39-52.

[3] J, N., & Linkens, D. (1995). *Fuzzy-Neural Control : Principles Algorithms and Applications.* UK: Prentice Hall.

[4] Maqsood, I. (2004). An Ensemble of Neural Network for Weather Forecasting. *Neural Computing and Application* , 13:112-122.

[5] Sahroni, A. (2012). An Ensemble of Neural Networks for Control System Application. *Advanced Topic in Artificial Intelligence.* Singapore: GSTF.

[6] Seng, T. L., Khalid, M., Yusof, R., & Omatu, S. (1998). Adaptive Neuro-Fuzzy Control System by RBF and GRNN Neural Networks. *Journal of Intelligence and Robotic Systems*, 267-289.

[7] Siang, J. T. (2002). *Neural Network and MATLAB Programming.* Yogyakarta, Indonesia: ANDI Offset.

[8] Specht, D. (1991). A General Regression Neural Network. *IEEE Transactions on Neural Network*, 568-576.

[9] Specht, D. (1995). *Probabilistic Neural Network and General Regression Neural Networks, Fuzzy Logic and Neural Network Handbook.* UK: Mcgraw Hill.

[10] Wiryadinata, R., & Ana Ratna Wati, D. (2005). ANN Simulation based on Back-propagation Algorithm for DC Motor Control System. *SNATI* (pp. C29-C35). Yogyakarta, Indonesia: SNATI UII.

[11] Yusof, R. (1994). *Theory and Applications of Self-Tuning PID and Generalized Predictive Controllers.* Japan: University of Tokusima.