工學碩士 學位論文

A Study on the Identification and Speed Control of a Diesel Engine Using Levenberg-Marquardt Backpropagation Algorithm Neural Networks

Levenberg-Marquardt Backpropagation Algorithm Neural Network을 이용한 디젤엔진 동정과 속도제어에 관한 연구

指導敎授　劉　永　昊

2002년 2월

韓國海洋大學校 大學院

制御計測工學科

金　璟　燁

Contents

A Study on the Identification and Speed Control of Diesel Engine Using Levenberg-Marquardt Backpropagation Algorithm Neural Network

Kim, Kyung Yup
Department of Control and Instrumentation Engineering

Abstract

Diesel engine is known as nonlinear system because of its dead time due to injection delay and ignition delay. So, it is very difficult and complex to model this nonlinear system because it varies widely according to number of cylinder and RPM.
In this paper, in order to design the speed control system of a diesel engine, neural network architecture is introduced and the optimal structure of neuro emulator is determined based on the modelling of a diesel engine, trained with various backpropagation algorithms and the performance of each trained networks is compared . Also, neuro controller, the inversely trained neural network of neuro emulator, is designed for the speed control system of a diesel engine. The selective neuro controller is proposed for the sake of improvement of the neuro controller performance and by combining a PI controller with the proposed controller, the efficiency of this combination speed control system of a diesel engine is ascertained.

Chapter 1. Introduction

1.1 Background

Dead time, one of the main reasons which make diesel engine to be a nonlinear system is mainly caused by injection delay and ignition delay. So it varies according to the speed of the engine and number of cylinders. There are a lot of studies on model of diesel engine for speed control purpose. For example, G.E Harland and K.F.Gill[8] modelled diesel engine as second order system which is shown in Fig. 1.1

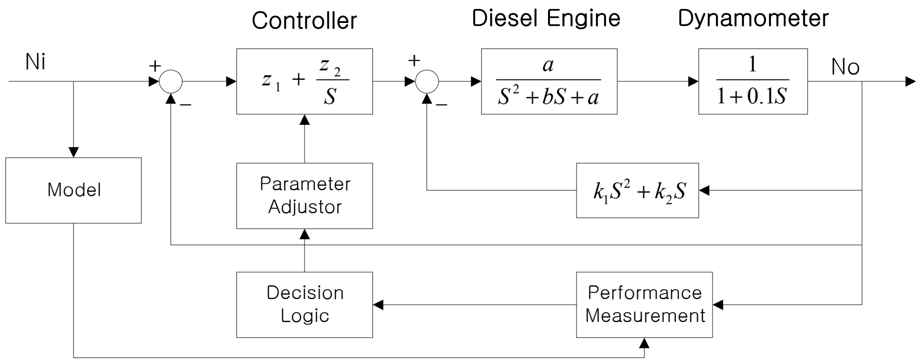

Fig. 1.1 Block diagram of a diesel engine control system proposed by Harland

As the parameter of diesel engine system becomes changed, the ratio of k_1/k_2 is obtained experimentally and the method to tune the PI controller in order to be close to model response was implemented with analog computer.

P.E.Wellstead, C.Thiruarooran, D.E. Winterbone[9] made modelling 6cyl. 4 stroke, medium speed (operating speed 1000rpm ~ 1800 rpm), turbo-charged 6 YEX Ruston Diesel Engine(MARK II) divided into three part, that is, HSHL(High Speed High Load, 1800 rpm, 810Nm), MSML (1400 rpm, 620Nm) and LSLL(1000 rpm, 210Nm) due to its nonlinear characteristics.

In the study on controlling the fuel injection timing in order to minimize the rate of fuel consumption in 3 cyl. 350 ps marine diesel engine, Y.Murayama, T.Terano, S.Masui, N.Akiyama[10] measured the ratio of fuel consumption five times every injection angle in order to obtain the precise ratio of fuel consumption and sought for optimal value through the regression analysis based on collected data in order to prevent the engine system from hunting nearby the optimal value. After then, they composed the fuzzy optimization control system to determine whether a series of operating was proper or not based on the knowledge of engineer and verified the performance through the simulation and application to real plant.

Norcontrol[11],[12] company modelled the dead time of the engine, combustion and revolution system as first order system each and developed the DGS8800 digital governor system with PI controller.

S. T. Lyngso[13],[14] company developed the EGS 900 system using MRAC(Model Reference Adaptive Control) algorithm.

But it is not easy to configure control system of diesel engine with satisfactory performance over all operating range based on linear control theory.

In Japan approximately 84% of the control industry still uses the conventional PID controller for its simplicity to implement.[1] From this fact, it seems that the specifications required for real applications of control theories are that the control algorithm should be simple enough to be implemented and to be understood. Recently, with sophisticate increase of performance of microprocessor, it is not difficult to implement complex control algorithm like fuzzy control and neural network ,so called intelligent control method which works well on the nonlinear control system. This intelligence appears to fuzzy or neural network in the shape of learning ability, flexibility, robust and nonlinearity.

1.2 Study Objective

As above mentioned, various modellings of diesel engine system have been achieved. Due to the nonlinearity of many parameters of diesel engine system, the control parameters need to be adjusted in order to apply to real system in the whole area through many experiments. It is difficult to find the appropriate parameters controlling the diesel engine system satisfactory over all situations. So, because of these difficulties of controlling diesel engine, intelligent control theory, that is, using control algorithm of fuzzy and neural networks has tendency to be applied to real plant. Especially, neural networks were found to be suitable for solving nonlinear and complex control problems that conventional and traditional control methods have no practical solution yet[1].

In this paper, even if the control plant is nonlinear and the system parameters of the plant are not clear, in order to control the speed of diesel engine used for driving generator robustly, it is proposed to design the speed control system of diesel engine using neural network. Because the training data collected from the engine aren't represented over all kinds of situation, the PI controller to compensate the generated error by only neuro controller is proposed and simulated to ascertain the performance of the speed control system using the MATLAB program.

This paper comprises 5 chapters. Chapter 2 gives a brief overview of neural networks to provide background knowledge for the purpose of modelling systems and designing controllers. In Chapter 3, an emulator is designed to model the feedforward dynamics of a diesel engine by various backpropagation algorithms. The neuro emulator is trained using training data collected from a real diesel engine system under various backpropagation algorithms. In Chapter 4, a neuro controller is designed, and trained with the selected backpropagation algorithm. Diesel engine speed control system is composed with trained neuro emulator and controller. After then, the results of this simulation is analyzed. Chapter 5 summarizes the conclusions of the study.

Chapter 2. Review of Neural Networks

This section provides background information on neural networks. Basic concepts and learning algorithms of neural networks are briefly described here.

2.1 Neuron Model

An ordinary artificial neuron (non-fuzzy neuron) is an information processing element which basically attempts to model the behaviour of the biological neuron. The model of an ordinary neuron is shown in Fig. 2.1. the input-output relationship of the neuron is given below:

$$o = f\left(\sum_{i=0}^{n} w_i x_i + \theta\right) \tag{2.1}$$

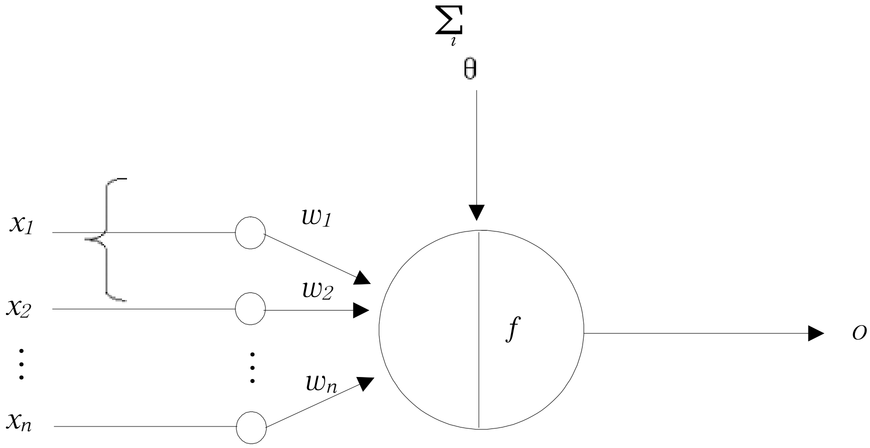

Fig. 2.1 A neuron model

where, x_i is the ith input, w_i is the synaptic weight associated with x_i, θ is a threshold level, o is the output, n is the number of inputs and f is an activation function which can be a hard limiter, a sigmoid function, a hyperbolic tangent function or a linear function, etc as follows.

(i) Hard limiter

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \qquad (2.2)$$

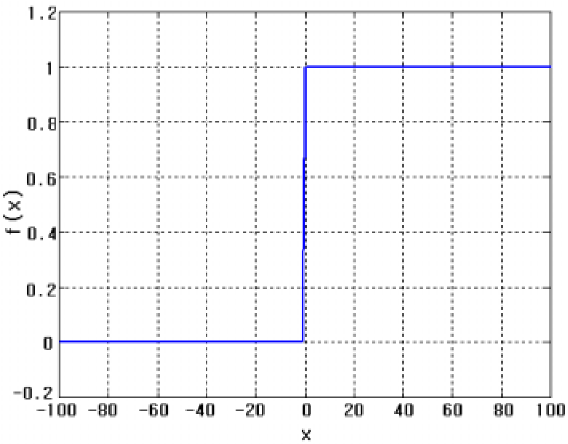

Fig. 2.2 Hard limiter

ii) Sigmoid function

$$f(x)=\frac{1}{1+e^{-x}} \tag{2.3}$$

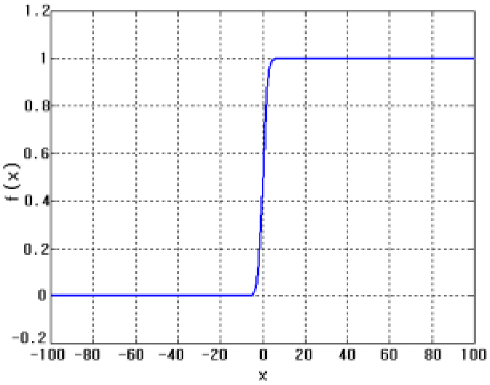

Fig. 2.3 Sigmoid function

iii) Hyperbolic tangent function

$f(x)=tanh(x)$ (2.4)

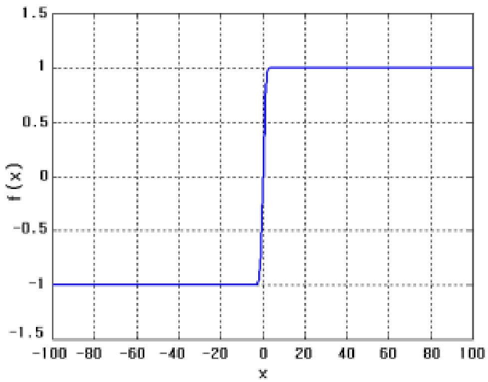

Fig. 2.4 Hyperbolic tangent function

iv) Linear function

$f(x)=x$ (2.5)

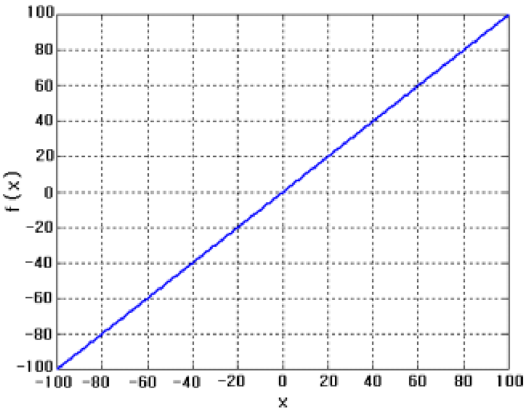

Fig. 2.5 Linear function

The neuron sums weighted inputs $w_i x_i$ $(1 \leq i \leq n)$ and passes the summed result through activation function f to produce the final output o.

2.2 Neural Networks

Due to their simplicity, a number of processing elements can be connected together in a well structured form called a neural network(NN). In a NN, processing elements are linked to one another via adjustable or fixed weights representing the strengths of the connections. These connection weights reflect the effect of the output of one neuron on another.

NNs can take a variety of forms, depending upon the way their processing elements are connected. For example, NNs can be classified as single-layer or multilayer networks, depending upon the organization of layers of processing elements, such as the input layer, the hidden layer and the output layer. If a network has only an input layer and an output layer, it is called a single-layer network. On the other hand, if a network contains one or more hidden layers in addition to the input and output layers, it is called a multilayer network.

NNs can be classified as feedforward or recurrent networks according to the flow direction of input signals. In a NN, if the input flows only in one direction from the input to the output, the NN is called a feedforward network. If, however, the network has at least on feedback loop, it is called a recurrent network.

The network structure and node interconnection are very important in determining the performance of the network. There is currently no systematic methodology for defining them and they are often decided by applying heuristics or trial-and-eror techniques.

2.3 Learning of Neural Networks

A NN can learn by employing a learning (or training) algorithm to capture knowledge. Interconnection weights between processing elements dictate the intelligence of the NN. The learning algorithm adjust the values of these weights to learn specific tasks. There are three types of network learning: supervised, unsupervised and reinforcement.

Supervised learning needs a set of training data. In this particular model of learning, the learning rule adjusts network weights according to the difference between the desired output and the network output, usually in such a way that the difference is minimized. Unsupervised learning is required in cases where the network is provided with only input signal.

In this learning mode, weights are adjusted by a learning rule based solely upon the input signals and the current network outputs. In reinforcement learning, the network receives a reward/penalty signal. The weights of the network are modified to develop an input and output behaviour which maximizes the probability of receiving a reward and minimizes that of receiving a penalty.

2.3.1 Simple Backpropagation

The dominant learning algorithm for NNs used in real applications is backpropagation[1]. Backpropagation is a supervised learning technique based on the gradient-descent method, which minimizes a quadratic error criterion measured at the output layer by means of modifying network weights. Backpropagation algorithm is explained as follows.

$$o_k = f(net_k) \tag{2.6}$$

$$\left.\begin{aligned} net_k &= \sum_j w_{kj}\, o_j + \theta_k \\ o_j &= f(net_j) \\ net_j &= \sum_i w_{ji}\, o_i + \theta_j \end{aligned}\right\} \tag{2.7}$$

$$\left.\begin{aligned} E_p &= \frac{1}{2}\sum_k (\tau_k - o_k)^2 \\ E &= \sum_p E_p \end{aligned}\right\} \tag{2.8}$$

where, i, j, k : input, hidden and output layer respectively.

τ_k : engine output

$o_{j,k}$: output of neuron at j,k layer

$net_{j,k}$: values of network at j,k layer

E : sum squared error

p : pattern number

$\theta_{j,k}$: bias values of neuron at j, k layer

$f(x)$: activation function

Learning can be carried out by following equations to reduce error for the output layer.

$$\left.\begin{aligned} w_{kj}(k+1) &= w_{kj}(k) + \Delta w_{kj}(k) \\ \Delta w_{kj} &= \eta \cdot \delta_k \cdot o_j \\ \delta_k &= (\tau_k - o_k) \cdot f'(net_k) \end{aligned}\right\} \tag{2.9}$$

where, η : learning rate, $(\eta > 0)$

$\delta_{j,k}$: error signal of neuron at j,k layer

And, for hidden layer the following equations can be carried out.

$$\left.\begin{aligned} & w_{ji}(k+1) = w_{ji}(k) + \Delta w_{ji}(k) \\ & \Delta w_{ji} = \eta \cdot \delta_j \cdot o_i \\ & \delta_j = f'(net_j) \cdot \sum_k \delta_k \cdot w_{kj} \end{aligned}\right\} \tag{2.10}$$

2.3.2 Backpropagation with Momentum(BPM)

There are often cases to drop into local minima when a system is learned with simple backpropagation and the error can't be reduced. For this, a momentum term is included.

$$\left.\begin{aligned} & \Delta w_{kj}(k+1)\ \eta\, \delta_k\, o_j + \alpha \Delta w_{kj}(k) \\ & \Delta w_{ji}(k+1) = \eta\, \delta_j\, o_i + \alpha \Delta w_{ji}(k) \end{aligned}\right\} \tag{2.11}$$

where, α is called momentum and generally used to be set to 0.9 [1].

2.3.3 Adaptive Backpropagation(BPA)

In order to decrease training time, adding the adaptive learning rate is very helpful. If $\frac{E_{new}}{E_{old}} > 1.04$ [2], the new weights, biases, output, and error are discard, and continued with new learning rate selected multiplying by 0.7[2]. In other cases, if E_{new} (2.12) $< E_{old}$, the increased rate continued with multiplied by 1.05[2].

2.3.4 Fast Backpropagation(BPX)

This is the hybrid type of two backpropagation algorithms, that is, backpropagation with momentum(BPM) and adaptive backpropagation(BPA). The former tends to minimize the possibility that the sum squared error stays in local minima with high error, and the latter tends to train faster than just backpropagation algorithm. So, this algorithm is called as fast backpropagation algorithm.

2.3.5 Levenberg-Marquardt Backpropagation(BPLM)

Gradient descent is a very simple search technique where parameters, such as weights and biases, are moved in the opposite direction to the error gradient. Each step down results in smaller error until an error minima is reached. The use of momentum changes this only sightly by making changes proportional to a running average of the gradient.

The Levenberg-Marquardt algorithm changes parameters by following rules and is more powerful than gradient descent.

$$d_k = \left(J^T(x_k)J(x_k) + \lambda_k I\right)^{-1} J^T(x_k)F(x_k)$$

$x_{k+1} = x_k - d_k$

where, λ_k : adaptive value (2.13)

$J(x_k)$: jacobian of $F(x_k)$

$F(x_k)$: error vector for all patterns

I : identity matrix

x_k : weight vector

If $F(x_k) < F(x_{k+1})$, adaptive value λ_k is decreased by multiplying predefined value 0.1[3]. If $F(x_k) > F(x_{k+1})$, λ_k is increased by multiplying predefined value 10[3].

2.4 Initialization of Neural Networks

In general, the weights of NNs are initialized randomly. For enhancing the learning performance in this paper, the weights are initialized using the Nguyen-Widrow initialization method, which considers the numbers of input nodes and output nodes. This method focuses optimal setting of the weights between input and hidden layer. Therefore, the weights between the hidden and output layers are initialized with the value between -0.5 and +0.5[5]. Those between the input and hidden layers are also initialized in this range $-0.5 < w_{ji} < +0.5$[5]. After then, they are changed and used by following rule before training[5].

$$\left. \begin{aligned} w_{ji}^{new} &= \frac{\beta \quad w_{ji}^{old}}{\| \quad w_{ji}^{old} \|} \\ \beta &= 0.7 \sqrt[n]{p} \end{aligned} \right\}$$

where, n : the number of input layer neurons. (2.14)

p : the number of the hidden layer neurons.

The biases weights in the hidden layer θ_j, is in the range of $-\beta < \theta_j < \beta$.

This method readjusts the weights between input layer and hidden layer before training and results in shortened training time in some application including XOR operation[5].

As already mentioned, NNs involve parameters, such as the learning rate and the momentum, selected for optimizing the learning performance. The values of these parameters are often determined by employing heuristics or trial-and-error techniques.

Chapter 3. Design of a Neuro Emulator for Diesel Engines

To obtain NNs to emulate a diesel engine system, the relationship between input and output of NNs needs to be designed considering the characteristic of the diesel engine system. For this, a model of the diesel engine system represented by a block diagram is considered in this chapter. Then, NNs are trained using above mentioned algorithms and are compared in terms of the training efficiency. The trained network is employed to the speed control system as the neuro emulator of diesel engine in the next chapter.

3.1 Modelling of a Diesel Engine System

The schematic diagram of a diesel engine speed control system is like Fig. 3.1.

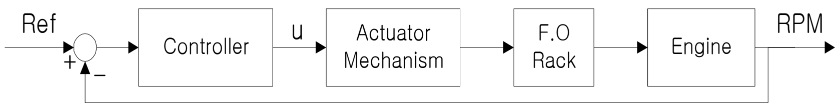

Fig. 3.1 Schematic diagram of the speed control system of diesel engine

Even if it is slightly different according to what kinds of actuator is used, actuator is generally regarded as first order system because the actuator movement is quite small in normal status. Injection delay and ignition delay make engine dead time as mentioned in Chapter 1. Combustion and revolution system can be modelled as a first-order system respectively, and shown in Fig. 3.2[15].

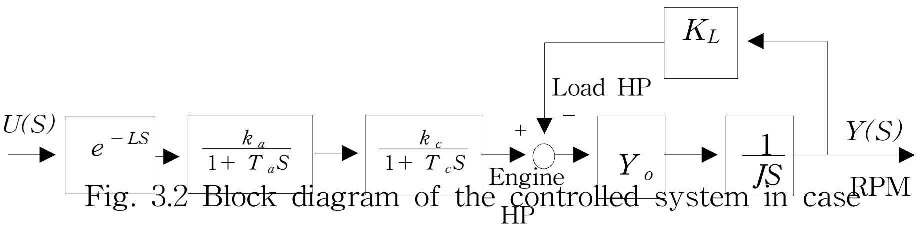

Fig. 3.2 Block diagram of the controlled system in case of regarding the engine dead time

where, L is the total dead time summing injection delay and combustion delay, T_c and k_c is time constant and steady state gain respectively in combustion system of engine, K_L is the load characteristics to convert rpm to Horse Power at operating rpm, J is the moment of inertia including propeller and additional water effects, Kg · m · sec^2. Y_o is the constant to convert Horse Power developed by engine to rpm. This engine system used for a generator holds on 1800[rpm] for four poles diesel driven generator. Even though the combustion system is modelled as first order system in Fig. 3.2, the pressure generated by combustion appears similarly to the shape of rectangular wave with delay time in 1800[rpm]. So, in high speed 1800[rpm], the combustion system considered with the dead time obtained from ignition delay and injection delay can be modelled as first order system. In this study, the diesel engine system is assumed with third order system considering the dead time.

3.2 Structure of a Neuro Emulator

The internal structure of a neuro emulator is configured like Fig. 3.3 based on a diesel engine modelled as third order system[1],[20].

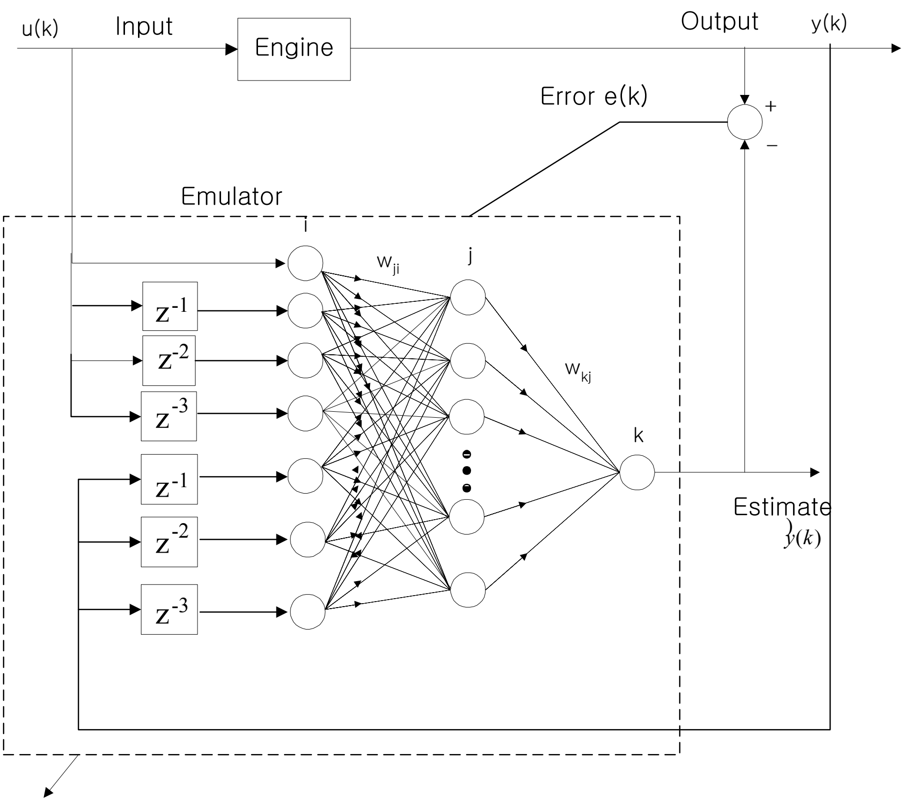

Fig. 3.3 Internal structure of a neuro emulator

The neuro emulator architecture considered in this paper is composed of three layers; an input layer, a hidden layer and an output layer. Based on the third-order system like Fig. 3.2, engine control signal u(k), engine output revolution per minute y(k) and their delayed signals u(k-1) ,u(k-2), u(k-3), y(k-1), y(k-2), y(k-3) are chosen as input of neuro emulator.

In order to determine the proper number of hidden node, the same structure as Fig. 3.3 with respect to the various number of hidden node was trained respect to the various number of hidden node. As expected, if the numbers of hidden node is small, the error of networks tends not to be converged at large error. But if the number of hidden node is too large, the convergency tends to be improved, but the number of epoch for training to be large. Based on try-and-error experiment[5], when the node number is selected from 9 to 12, this network meets convergency speed. Finally, eleven nodes for hidden layer are chosen.

The output layer consists of only one node corresponding to the estimate value of the engine output. The tangent sigmoid function is used for the activation function of the input neurons and the hidden neurons and the linear function for output layer[2].

3.3 Data Collection

To obtain training data set, the following system as shown in Fig. 3.4 is considered and then input and output data patterns are made. Data acquisition is carried out from a diesel engine speed control system composed of four parts; a digital governor, an actuator, a MPU, and a PC. Whenever MPU approaches close to the fly wheel teeth, it generates pulse relating to rpm and this rpm data makes feedback to the digital governor. According to the difference between reference rpm and feedback rpm, appropriate control input is generated. This control input signal makes an actuator operated and the fuel quantity injected into the engine is adjusted according to the control input. And, reference, rpm and control input are transmitted to the PC and stored in the shape of text files using RS232 serial communication.[4]

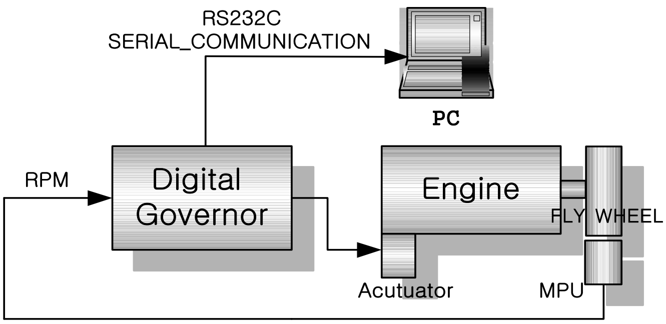

Fig. 3.4 Engine system

Table. 3.1 shows the specification of experimental devices in Fig. 3.4.

Table. 3.1 Engine system specification

Engine	
220[V], 3-phase, 50[Kw] Generator Driving 4 Cycle and 4 Cylinders 1800[rpm] ISUTSU Diesel Engine	
Digital Governor and Actuator	
Digital Govenor Actuator(Solenoid)	: ASA FzPI200 : GAC 175

Fig. 3.5 presents the control input with respect to above output data entering the actuator and the fuel quantity is adjusted according to this control input signal. Fig. 3.6 shows the trend of the diesel engine rpm used for training data. These data are obtained from above system and includes some disturbances and load condition so as to train the neural network in various conditions.

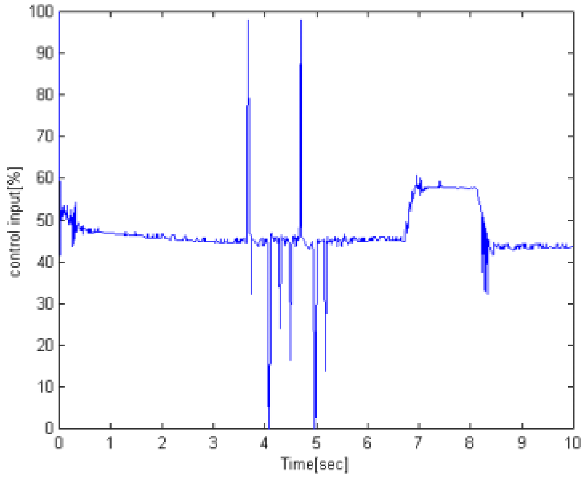

Fig. 3.5 Control input signal of the diesel engine used for training

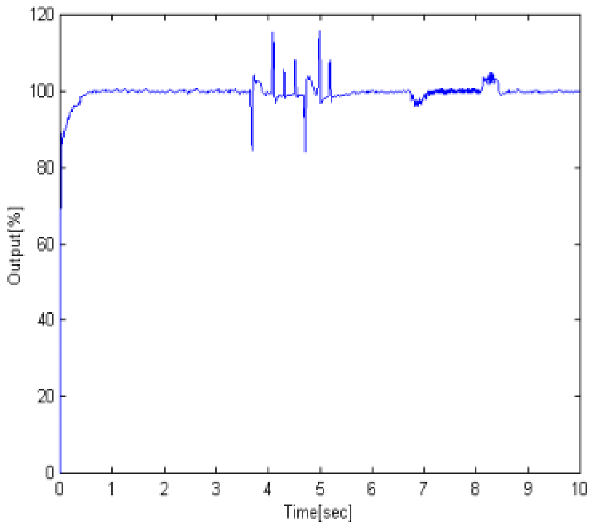

Fig. 3.6 Output signal of the diesel engine used for training

3.4 Training Results and Analysis with respect to Various Backpropagation Algorithms

When the system is identified using NNs as depicted in Fig. 3.3, the off-line training using input and output patterns obtained through the engine system results in Fig. 3.7 and Fig. 3.8 according to three kinds of backpropagation algorithms.

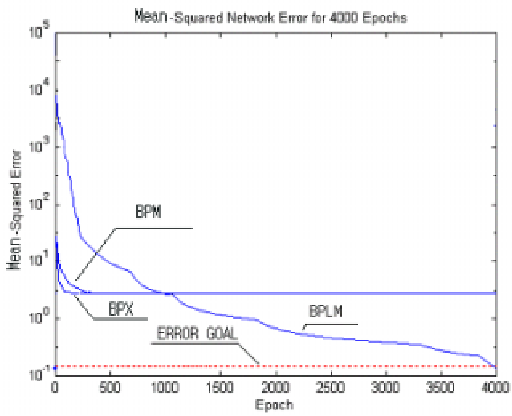

Fig. 3.7 Mean squared errors

Mean squared errors with respect to training epochs are shown in Fig. 3.7. As seen through the above figure, BPM and BPX have less efficiency than BPLM in a view of convergency speed to the error goal. In addition, time responses of NNs trained by the three backpropagation algorithms are compared with each other as following Fig. 3.8, 3.9, and 3.10. As above mentioned, the network trained by BPLM follows the real engine output with small MSE compared with BPM and BPX.

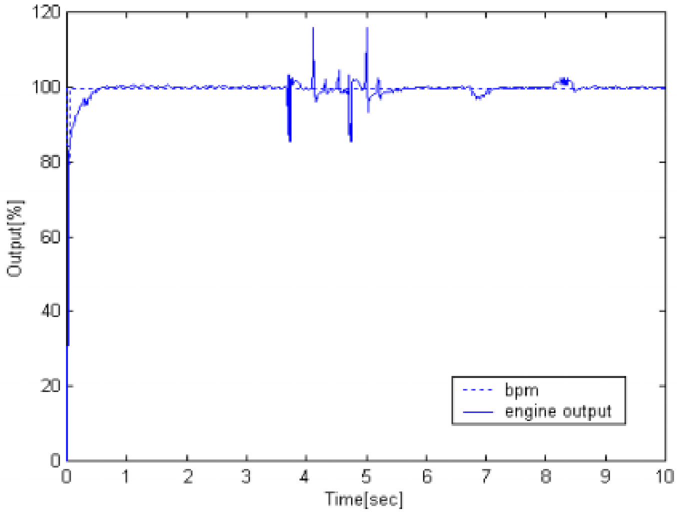

Fig. 3.8 The dynamic response characteristics of neural networks trained by bpm algorithm

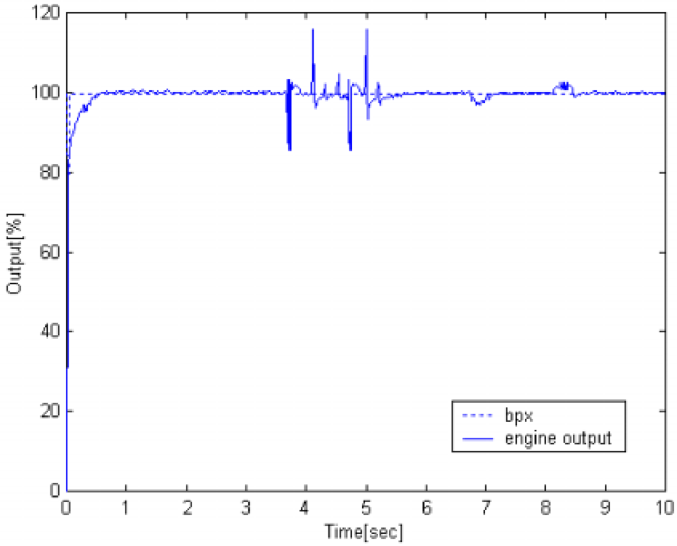

Fig. 3.9 The dynamic response characteristics of neural networks trained by bpx algorithm

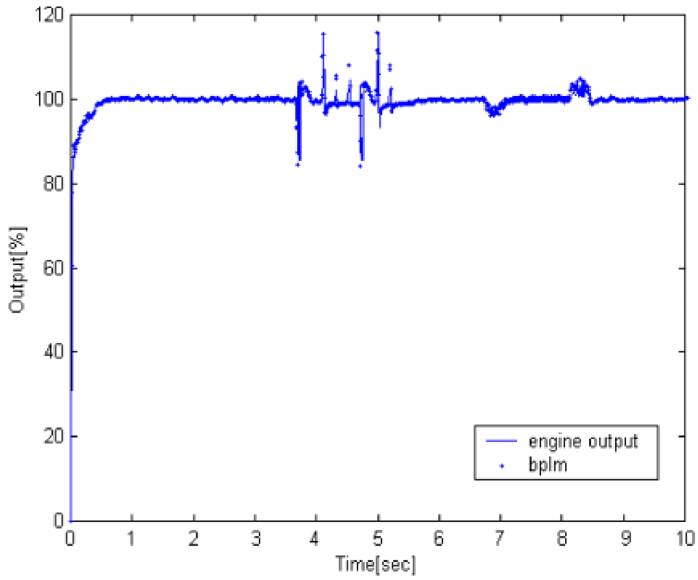

Fig. 3.10 The dynamic response characteristics of neural networks trained by bplm algorithm

Table. 3.2 Simulation results using training data

	BPLM	BPM	BPX
Learning Rate*		0.0001	
Momentum*		0.95	
λ_k*	0.001		
Epoch	4000	4000	
Error Goal	0.1		
Convergence	O	X	
MSE	0.09	4.2	4.2

* : default

In case of applying BPM and BPX, the mean squared error didn't reach the error goal in spite of 4000 epochs, but BPLM algorithm needed 4000 epochs to converge on the error goal. Table 3.2 shows the final parameters after training completion.

The efficiency of the trained network was evaluated with validate data. It is shown in Fig. 3.11 that NNs outputs are similar to the engine real output y(k) in case of using validate data. The MSE calculated with neuro emulator and engine output is about 3.

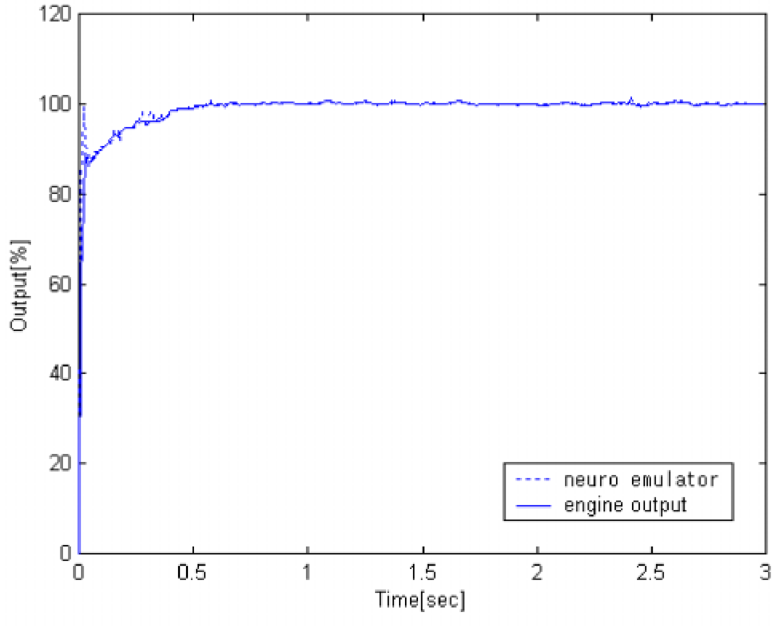

Fig. 3.11 The dynamic response characteristics of the diesel engine and the neuro emulator using validate data

Chapter 4. Design of a Neuro Controller for Diesel Engines

In order to compose the neuro control scheme, a neuro controller is designed using training data. Due to some issued problems in this chapter, an alternative control system is proposed to compensate the generated error.

4.1 Neuro Controller Design

For purpose of composing a series control system, a neural network is employed to identify inverse dynamics models through learning. Inverse dynamics identification is regarded as finding the inverse mapping of the plant as illustrated in the following architecture.

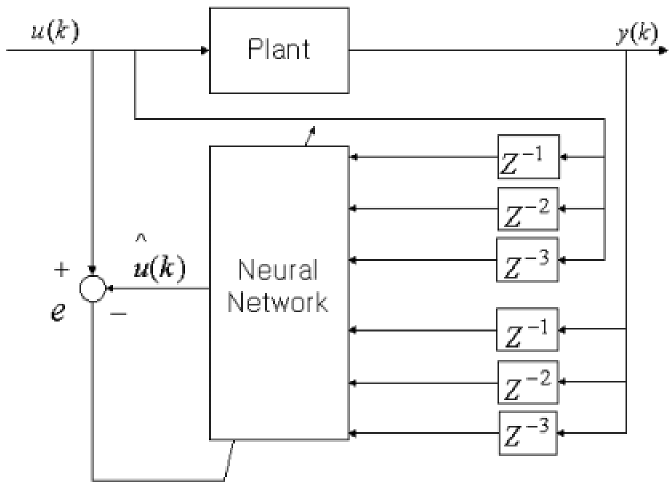

Fig. 4.1 Identification of plant inverse dynamics

This architecture is similar to the above mentioned plant identification scheme, but the input signals of networks are different from the case of plant modeling.
Since this inverse identification is obtained for control purpose, it should generate the control signal with respect to output signal. The following Fig. 4.2 shows the comparison with the output of the neuro controller trained using BPLM algorithm and the training data control input u.

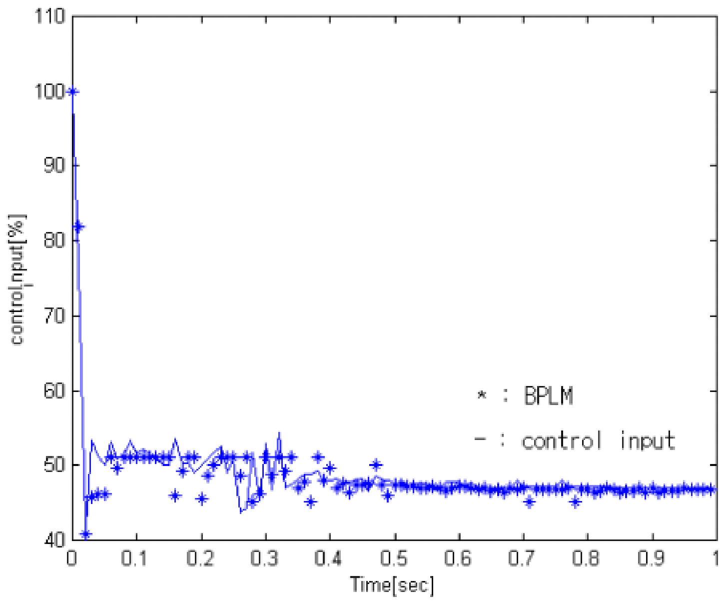

Fig. 4.2 The dynamic response characteristic of control input of the neuro controller trained with training data and real plant

The MSE with respect to every pattern is about 4.33. The inversely trained networks will generate appropriate outcomes and be able to control the speed of the diesel engine instead of the controllers which has been used conventionally. The following graph compares real control input with the control input generated by inversely trained neural networks, so called neuro controller in case of using validate data.

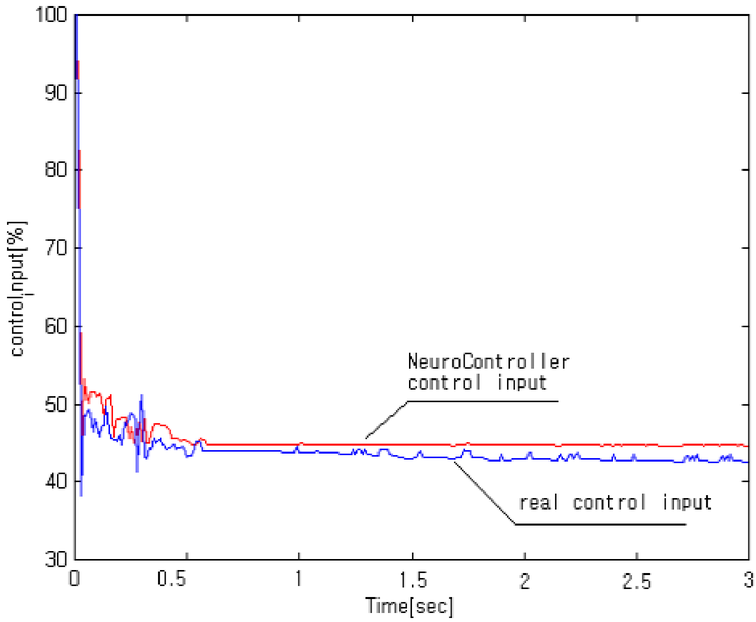

Fig. 4.3 The dynamic response characteristics of neuro controller by using validate data

4.2 Design of a Neuro Control System

The neuro control scheme is shown in Fig. 4.4. In training the neural network, the reference is included in training data because the training data is collected based on reference value. So, like Fig. 4.1, only six inputs, $u(k-1)$, $u(k-2)$, $u(k-3)$, $y(k-1)$, $y(k-2)$, $y(k-3)$, are chosen as input of the neuro controller except for reference $r(k)$. Fig. 4.5 shows the response of the diesel engine speed control system from start to steady state.

In addition, impulse disturbance is added to system, and the dynamic characteristics of the diesel engine speed control system is investigated. But, as seen in Fig. 4.2, current neuro controller was trained with training data obtained from the start to steady state without disturbance. As shown by the Fig. 4.6, when the system is stimulated by disturbance, neuro controller does not work well and generate system hunting. However, if the neuro controller is trained using training data including disturbance and load condition, it tends not to be trained very well in whole area. In other words, the training stops with relatively large MSE. In order to solve this difficulties, this training data is divided into two parts; the part from start to steady state without disturbance and the disturbance part. Using these two kinds of training data, two neuro controllers, that is, neuro controller 1 and neuro controller 2, are trained with respect to each part. After then, one of the two neuro controllers is selected according to the operating environment as Fig. 4.7.

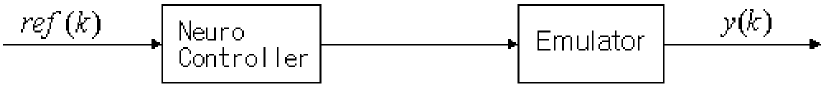

Fig. 4.4 Series neuro control scheme

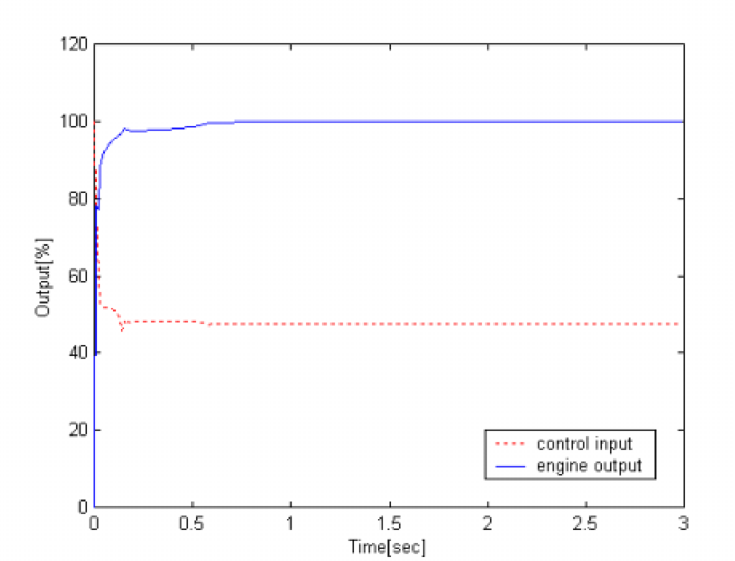

Fig. 4.5 The dynamic response characteristics of the engine by using series control system

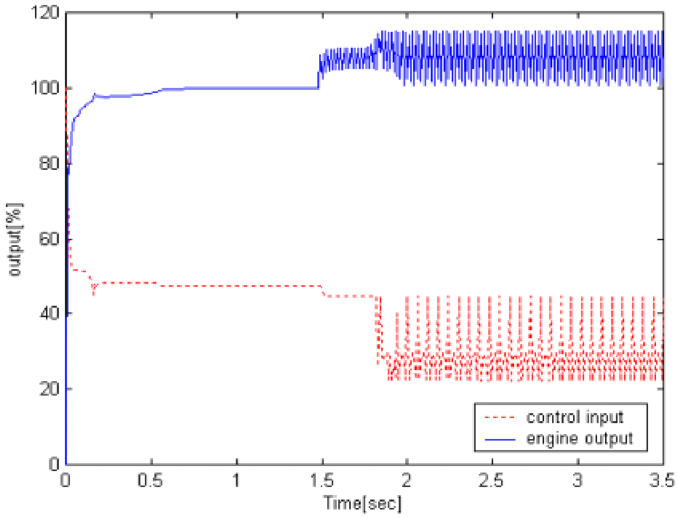

Fig. 4.6 The dynamic response characteristic of neuro controller in stimulating the system with impulse noise without training the neuro control with noise data

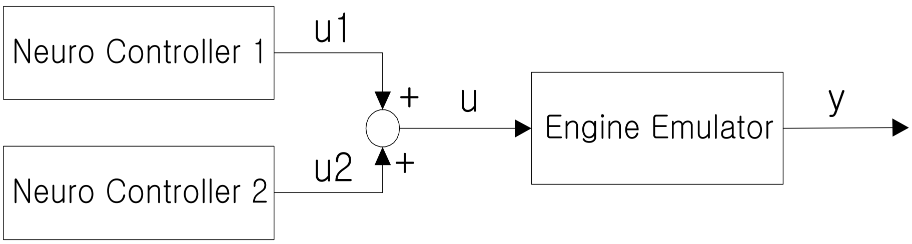

Fig. 4.7 Switching of the neuro conroller

The proposed neuro controller, Fig. 4.7, performs appropriate operation for the noise like Fig. 4.8. However, after the system is stimulated by disturbance, it takes much time to return to the reference. So, this delay time will be compensated by a PI controller.

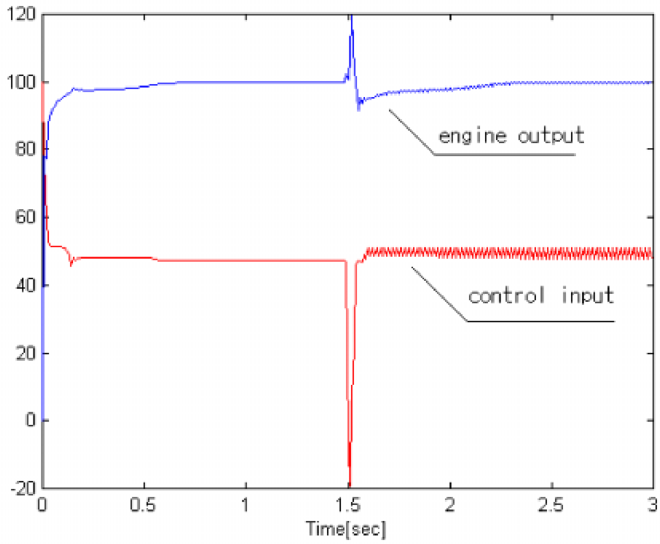

Fig. 4.8 The dynamic response characteristics of the engine in case of adding some impulse noise

4.3 Design of Combination Control System with PI and Neuro Controller

As depicted in the above figures, the control scheme has some potential problems, that is, hunting, offset, etc because the trained network doesn't globally represent the system. As far as there are these kinds of problems, it is not desirable to apply this neuro controller to the diesel engine control system. So, to make better control performance of the neuro control systemfor diesel engine, a compensated control scheme is introduced as proposed by Kawato et al.[1]. Fig 4.9 shows the control scheme with a PI type compensator.

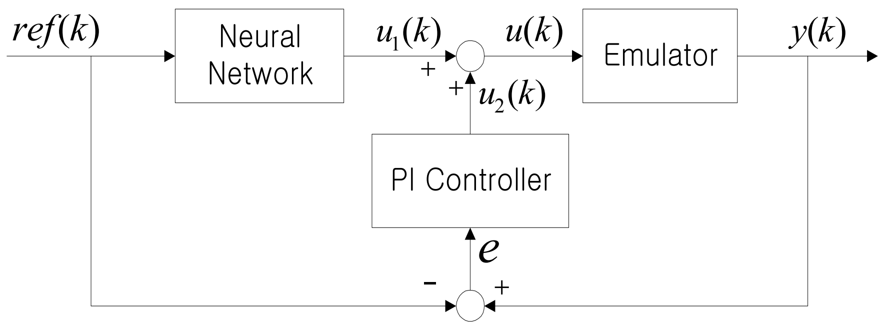

Fig. 4.9 Combination type neuro control scheme with a PI type compensator

In the control scheme, NNs play a role as a main controller and the PI controller as a auxiliary controller. The PI controller is used to adjust control input $u(k)$ for the plant to follow a desired reference $ref(k)$ as precisely as possible. As shown in Fig 4.10, the response approaches the reference level faster than the control scheme using only a neuro controller.

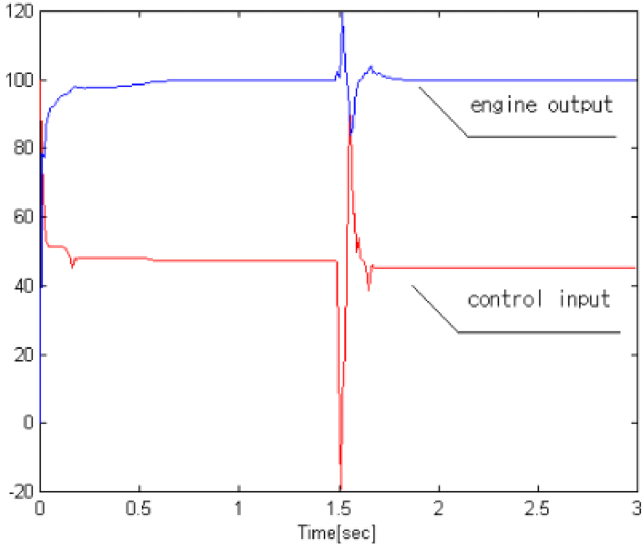

Fig. 4.10 The dynamic response characteristic of engine using proposed combination control system

Chapter 5. Conclusion

In this thesis, neural emulator and neuro controller for speed control system of diesel engine are proposed. To find out optimal configuration of neuro emulator for diesel engine, various kinds of backpropagation algorithm is compared. Among above mentioned three kinds of backpropagation algorithms, backpropagation algorithm using Levenberg-Marquardt optimization was proven to be most optimal for diesel engine identification.

In order to improve the control performance, selective training method for neuro controller is proposed and neuro controller trained by this method was proven to be more efficient on speed control of diesel engine in the case of existing disturbance.

For fast response in the case of existing disturbance, combination control system which is combined with neuro controller and conventional PI controller is proposed. Simulated results show that combination control system with neuro controller and conventional PI controller controls efficiently speed of generator driven by diesel engine in the case of existing disturbance.

In the future the study about developing dedicated controller implemented this combined neuro control system and application to real diesel engine should be done.

Reference

[1] Sigeru Omatu, Marzuki Khalid, Rubiyah Yusof, Neuro-control and Its application, Springer, 1995.

[2] Howard Demuth, Mark Beale, TOOLBOX-Neural Network, The MathWorks, 1993

[3] Andrew Grace, TOOLBOX-Optimization, The MathWorks,1993

[4] Y. I. Kim, "A Study on the Speed Control of Medium Speed Diesel Engine using the Fuzzy-PI Controller", Korea Maritime University, 1999

[5] C.S.Oh, Neuro-Computer, Naeha, 2000

[6] Mo-Yuen Chow, Methodologies of Using Neural Network and Fuzzy Logic Technologies for Motor Incipient Fault Detection, World Scientific, 1997

[7] H. G. Tak et al. "A Study on the Optimization of Backpropagation Learning Algorithm", Kyunghee University, 1992

[8] G. E. Harlang and K. F. Gill, "Design of a Model Reference Adaptive Control for an Internal Combustion Engine", Measurement and Control Vol.6, PP. 173-176, 1973

[9] P. E. Wellstead, C. Thiruarooran, and D. E. Winterbone, "Identification of Turbo-charged Diesel Engine", 1978, Proceeding 7th IFAG world congress Helsinki, Filand

[10] Y.Murayama, T.Terano, S.Masui, N.Akiyama, "Optimizing Control of a Diesel Engine"

[11] Norcontrol, "Diesel Governor System DGS8800" Doc. No. 0759A, 1983
[12] Norcontrol, "Diesel Governor DGS8800 System Information" 1985, 3
[13] S. T. Lyngso, "Electronic Governor System EGS900" CI900.166.623 version 2.01, 1987
[14] S. T. Lyngso, "Electronic Governor System EGS900" STL 07.A.32 version 2.02
[15] Y. H. Yu, "A Study On The Speed Control Of A Marine Diesel Engine ", Korea Maritime University, 1990
[16] S. B. Lee, Fuzzy-Neuro Control System, Kyohasa, 1999
[17] Serdar Iplikci, "IEEE NNC Student Summer Research Support Program Report", IJCNN'2001
[18] L. C. Jain, R. P. Johnson, Y. Takefuji, L. A. Zadeh, Knowledge-Based Intelligent Techniques in Industry, CRC Press, 1999, PP.193-223
[19] Valluru B. Rao, Hayagriva V. Rao, C++ Neural Networks and Fuzzy Logic, MIS Press, 1993, PP.103-147
[20] Martin Brown, Chris Harris, Neurofuzzy Adaptive Modelling and Control, Prentice Hall, 1994, PP.30-74

감사의 글

지난 2년간의 대학원 생활을 마감하면서 돌이켜보니 많은 기억들이 떠오릅니다. 주어진 과제들을 맡으면서 문제에 부딪힐 때마다 밤을 지새우고 고민해야했던 시간들과 겪어야했던 시행착오의 경험은 쉽게 얻지 못할 값진 재산이 되었습니다. 하지만, 이러한 소중한 기억들의 주인공이 저 혼자만은 아니기에 함께 했던 많은 분들께 지면으로나마 감사의 마음을 전하고자 합니다.

부족한 제자를 받아주시고 지금의 이 자리까지 헌신적으로 이끌어 주시고 학문의 참 스승으로서 그리고 인생의 참 선배로서 격려와 채찍을 아끼지 않으신 유영호 지도교수님께 진심으로 감사드립니다. 또한, 본 논문을 세밀하게 검토해주시고 값진 충고로 관심을 기울여 주신 진강규 교수님과 김종화 교수님께 감사드리며, 항상 관심어린 눈으로 지켜봐 주신 조석제 교수님과 그외 자동화 정보 공학부의 모든 교수님께 감사드립니다.

언제나 부담 없이 다가설 수 있도록 친절하게 많은 도움을 주시는 천행춘 이사님께 감사의 말을 전하고 싶습니다. 그리고 대학원 2년 동안 누구보다 많은 시간을 함께 하면서 꼼꼼하게 조언해주시고 지쳐있을 때 격려의 말로 제 곁에서 힘이 되어주신 영일형께 감사의 마음을 전하고 싶습니다. 그리고 1년간 선배로서 친구로서 도움과 격려를 준 인호에게 감사의 말을 전합니다. 그리고, 2년이라는 시간 동안 실험실에서 함께 기쁨과 슬픔을 나누었던 복산과의 추억도 소중하게 간직하고 싶습니다. 그리고, 대학원 생활동안 나의 벗으로

함께 하며 위로가 되었던 성일과 성호에게 감사의 마음을 전합니다. 그리고 언제나 도움주기를 거절하지 않은 강주와 하동경 선배에게도 감사의 말을 전합니다. 그리고 늘 친절한 미소로 위로해주신 군호선배와 꼼꼼하게 학사업무를 챙겨주신 김경언 조교님께도 감사의 말씀을 전합니다. 그리고 논문이 나오기까지 여러모로 도움을 준 순영이와 이제 막 대학원 생활을 시작하게 될 현경에게도 대학원 생활을 더욱 열심히 하라는 당부의 말을 전합니다. 그리고 많은 분량의 원고를 불평없이 세심하게 봐준 경은 자매에게 감사의 마음을 전해드립니다.

지금까지 항상 가까이에서 오빠에 대한 신뢰를 잃지 않고 지켜봐준 저의 두 동생, 경미, 미령과 함께 이 기쁨을 나누고 싶습니다.

그리고 아들을 위해 전폭적인 신뢰와 헌신적인 사랑으로 돌봐 주시며 언제나 제게 힘이 되어주시길 원하시는 저의 소중한 부모님께 진심으로 감사드리며 사랑한다는 말씀을 전하고 싶습니다.

마지막으로 부족한 한 인간을 선택하시고 자기 목숨을 버리기까지 하신 그 무한한 사랑으로 지금도 변함없이 살아 계셔서 나의 삶 속에서 일하시는 모든 지식의 근본이신 나의 주 나의 하나님을 찬양하며 이 모든 영광을 그분께 돌려드립니다.