

Combustion Control of Diesel Engine using Feedback Error Learning with Kernel Online Learning Approach

Elfady Satya Widayaka¹ and Hiromitsu Ohmori²

¹Graduates School of Science and Technology, Keio University

²Department of System Design Engineering, Keio University
3-14-1, Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan

Email : elfady.satya@gmail.com, ohm@sd.keio.ac.jp

Abstract. This paper shows how to design Multivariable Model Reference Adaptive Control System (MRACS) for “Tokyo University discrete-time engine model” proposed by Yasuda et al (2014). This controller configuration has the structure of “Feedback error learning (FEL)” and adaptive law is based on kernel method. Simulation results indicate that “kernelized” adaptive controllers can improve the tracking performance, the speed of convergence and the robustness to disturbances.

1. Introduction

In diesel engine, the usage of common rail system, Exhaust Gas Recirculation (EGR), and supercharger technology in combustion system allows diesel engine to achieve balance in thermal efficiency and environmental performance (soot emission and engine noise). Combustion system in diesel engine is a complex system which takes consideration of mechanical, thermal, fluid, chemical process and reaction. Calculation regarding those processes is performed by onboard computer known as Engine Control Unit (ECU). With the global spread of diesel engine recently, control system that built upon mathematical model of combustion process is actively researched.

In this paper, the proposed control configuration is 2-Degree of Freedom control system called Feedback Error Learning (FEL) [7] [8]. The focus is designing Feedforward (FF) controller based on inverse model the plant. There are 5 type of algorithm proposed in controller design.

(Algorithm I) Adaptive FF controller designed from linearized plant model.

(Algorithm II) Adaptive parameter in algorithm I is sensitive to disturbance. To increase robustness, dead zone band is introduced to controller on top of adaptive controller.

(Algorithm III) Since the plant is nonlinear, it is not possible to obtain perfect inverse model. Therefore, kernel method is used to model non-linear elements in FF controller.

(Algorithm IV) Similar to algorithm I, dead zone band also applied to algorithm III to increase robustness.

(Algorithm V) Algorithm III and IV uses kernel trick to model system nonlinearity, in algorithm V neural network is used to model the nonlinearity and inverse plant.

Algorithms above will be applied to discrete-time onboard combustion model of diesel engine by Tokyo University (Yasuda et. al. [1] [2] [3]). Algorithm III and IV is expected to perform well against disturbance, owing to Kernel method nonlinear modeling capability and robustness to disturbances. The effect of including dead zone method can be observed in algorithm II and IV.



The objective of this paper is to propose an advanced control system. Especially, control system that capable of updating its parameter when experimented or applied in real life.

Generally, linear FEL controller shows good reference tracking performance when used against linear system. However, when used against nonlinear system the performance deteriorated [4]. This reduced performance is caused by imperfect linearization of nonlinear model. By using Adaptive control to the FEL controller, the tracking performance on nonlinear system is improved. However, adaptive FEL controller is sensitive to disturbance, making convergence time a bit of an issue. To address this issue, FEL is expanded using Kernel method [11]. To improve both convergence time and tracking capability, 5 algorithms stated before is designed, simulated and compared in this paper.

2. Tokyo University Discrete-Time Engine Model

Onboard fuel combustion model proposed by Yasuda et.al [1] [2] (referred as Tokyo University Model from now) is discrete-time combustion model diesel engine. In Tokyo University Model, one engine cycle is divided into several discrete points (i.e. intake, compression, etc.) in order to obtain minimum information needed to do predictive control of combustion. The combustion model used from here on is single injection model. The discretization point of one engine cycle is shown in figure 1, while table shows inputs, outputs, and state variables of combustion model.

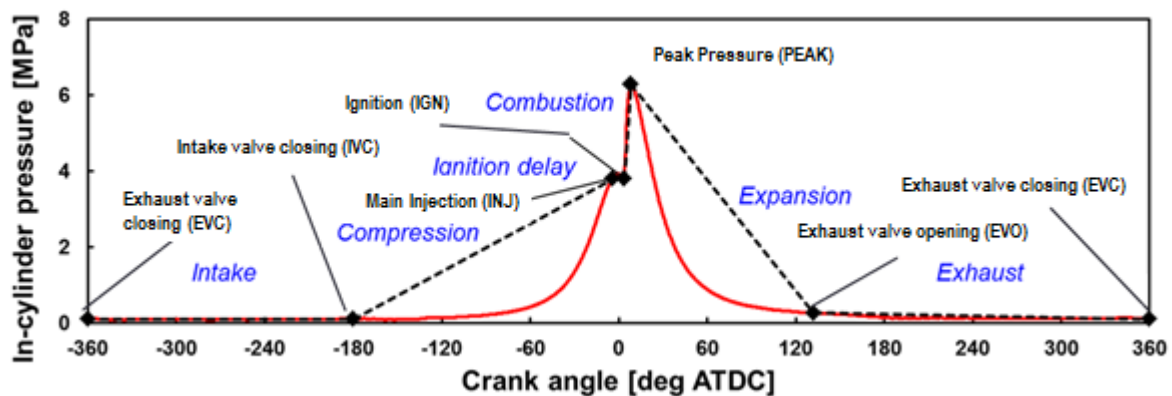


Figure 1. Discretization of Engine Cycle.

Table 1. State variables, inputs, and outputs of the plant.

State Variables		Input Variable		Output Variable	
$T_{RG,k}$	Residual Gas Temperature	$Q_{fuel,k}$	Injection Quantity	W	Load
		$\theta_{INJ,k}$	Injection Timing	$P_{PEAK,k}$	Peak Pressure
$n_{O_2,RG,k}$	Residual Gas Oxygen Mole	$P_{boost,k}$	Booster Pressure	$\theta_{PEAK,k}$	Peak Pressure Timing
		φ_k	External EGR Rate		

If exhaust valve closing time is taken as the starting point of k-th cycle, including combustion process and other polytropic changes within engine cycle, there are 6 event points where linear approximation is done. The 6 points are EVC (Exhaust Valve Closing), IVC (Intake Valve Closing), INJ (Fuel Injection), IGN (Fuel Ignition), PEAK (Peak Pressure), and EVO (Exhaust Valve Opening). After calculating the system output such as peak pressure, the state variable for k+1 cycle can be obtained. The state equation and output equation then can be represented as below (Yasuda et. al. [1] [2]). For simplicity, external EGR rate input in this paper is fixed to 30% value.

State Equation:

$$\begin{cases} T_{RG,k+1} = f_1(T_{RG,k}, n_{O_2RG,k}, Q_{fuel,k}, \theta_{INJ,k}, P_{boost,k}, \varphi_k) \\ n_{O_2RG,k+1} = f_2(T_{RG,k}, n_{O_2RG,k}, Q_{fuel,k}, \theta_{INJ,k}, P_{boost,k}, \varphi_k) \end{cases} \quad (1)$$

Output Equation:

$$\begin{cases} W_k = f_3(T_{RG,k}, n_{O_2RG,k}, Q_{fuel,k}, \theta_{INJ,k}, P_{boost,k}, \varphi_k) \\ P_{PEAK,k} = f_4(T_{RG,k}, n_{O_2RG,k}, Q_{fuel,k}, \theta_{INJ,k}, P_{boost,k}, \varphi_k) \\ \theta_{PEAK,k} = f_5(T_{RG,k}, n_{O_2RG,k}, Q_{fuel,k}, \theta_{INJ,k}, P_{boost,k}, \varphi_k) \end{cases} \quad (2)$$

Function $f_1, \dots, f_5$ can be derived by using volume, pressure, and temperature calculated in each linearization point in engine cycle. By linearizing around a reference point, the following system is obtained.

$$\begin{bmatrix} T_{RG,k+1} \\ n_{O_2RG,k+1} \end{bmatrix} = A \begin{bmatrix} T_{RG,k} \\ n_{O_2RG,k} \end{bmatrix} + B \begin{bmatrix} Q_{fuel,k} \\ \theta_{INJ,k} \\ P_{boost,k} \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} W_k \\ P_{PEAK,k} \\ \theta_{PEAK,k} \end{bmatrix} = C \begin{bmatrix} T_{RG,k} \\ n_{O_2RG,k} \end{bmatrix} + D \begin{bmatrix} Q_{fuel,k} \\ \theta_{INJ,k} \\ P_{boost,k} \end{bmatrix} \quad (4)$$

Equation above can also be expressed as $x_{k+1} = Ax_k + Bu_k, y_k = Cx_k + Du_k$ with A, B, C , and D are constant matrixes with appropriate dimensions. The discrete-time system is not obtained by sampling real-time system with certain interval, but an event-based discrete time system with 3 input (fuel injection quantity, fuel injection timing, boost pressure), 3 output (load, peak pressure, peak pressure timing), and 2 state variables (residual gas temperature, mole of residual gas). Assuming symmetrical matrix D exist with $\det D \neq 0$, then the equation (3) and (4) have the same nonlinear structure as equation (1) and (2).

Feedthrough term might cause some causality problem with feedback. To avoid this, the input side is expanded to include one-step delay as below:

$$u_k = \frac{I_3 z^{-1}}{1 - a z^{-1}} v_k = \frac{I_3}{z - a} v_k, u_{k+1} = a I_3 u_k + v_k \quad (5)$$

The parameter is set $a := 0.16$. By substituting the expansion term above, the system can be rewritten as equation below.

$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \begin{bmatrix} A & B \\ O & aI \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \begin{bmatrix} O \\ I \end{bmatrix} v_k, y_k = \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \quad (6)$$

On equation (6), the input term previously u_k is now replaced by v_k .

3. Feedback Error Learning

Feedback Error Learning (FEL) was originally proposed as learning mechanism imitating motoric control on living organism (Kawato et. al. [7] [8]). In living organism, inverse model of motor system is computed adaptively in cerebellum, which leads to a fast and smooth feedforward control of motor system. FEL is a 2-Degree of Freedom control system with the good tracking capability of feedforward control and disturbance suppression of feedback (FB) control. Configuration of FEL controller is shown in figure 2.

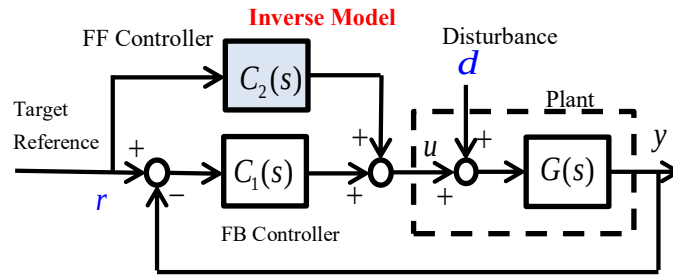


Figure 2. FEL Configuration.

If W_r is reference tracking model from reference input r to output y and W_d is disturbance model from disturbance d to output y , FF controller and FB controller is C_2 and C_1 respectively, theoretically, both controller can be designed as followed.

$$C_1 = W_d^{-1} - G^{-1}, \quad C_2 = G^{-1} \quad (7)$$

When W_d is close to zero, it is necessary to suppress the disturbance. However, it will result in high gain on feedback controller. On the other hand, to make sure W_r term tracks the reference input, the FF controller gain in lower frequency band need to be increased. Engine control algorithm with structure as shown in figure 2 then proposed.

4. FEL Engine Control

Generally, with only FEL configuration, steady state error will always present due to linearization. Thus, the controller needs to be expanded into its adaptive form. The adaptive form is designed using the concept presents in [9] and [10]. Using equation (6) as base equation, the relationship of input-output of the system can be rewritten into equation below.

$$\begin{aligned} y_{k+1} &= -A_0 y_k - A_1 y_{k-1} - A_2 y_{k-2} + B_0 v_k + B_1 v_{k-1} + B_2 v_{k-2} \\ &= A(z^{-1})y_k + B(z^{-1})v_k \end{aligned} \quad (8)$$

The polynomial matrixes A and B are as stated below.

$$A(z^{-1}) := -A_0 - A_1 z^{-1} - A_2 z^{-2}, \quad B(z^{-1}) := B_0 + B_1 z^{-1} + B_2 z^{-2}$$

Since B_0 is non-singular matrix ($\det B_0 \neq 0$), equation (8) can be rewritten to equation below.

$$\begin{aligned} y_{k+1} &= A(z^{-1})y_k + B_0 v_k + [B(z^{-1}) - B_0]v_k \\ &= B_0[v_k + B_0^{-1}A(z^{-1})y_k + B_0^{-1}\{B(z^{-1}) - B_0\}v_k] \end{aligned} \quad (9)$$

With y_M as expected output value, FB controller $C_1(z) = K$, and output error is formulated as $e_k := y_k - y_{M,k}$, the FB controller output $v_{FB,k}$ can be derived as shown in equation (10). Parameter of FF controller will be adjusted to make $v_{FB,k}$ close to zero, will be adjusted.

$$\begin{aligned} v_{FB,k} &= K e_k = -K[y_{k+1} - y_{M,k+1}] \\ &= -KB_0[v_k + B_0^{-1}A(z^{-1})y_k + B_0^{-1}\{B(z^{-1}) - B_0\}v_k - B_0^{-1}y_{M,k+1}] \\ &= -KB_0 v_{FB,k} - KB_0[-v_{FB,k} + v_k + B_0^{-1}A(z^{-1})y_k + B_0^{-1}\{B(z^{-1}) - B_0\}v_k \\ &\quad - B_0^{-1}y_{M,k+1}] \\ &= -KB_0 v_{FB,k} - KB_0[v_k - v_{FB,k} + \Theta \varphi_k] \end{aligned} \quad (10)$$

Θ is constant matrix and the signal vector is defined as below.

$$\varphi_k := [-y_k^T, -y_{k-1}^T, -y_{k-2}^T, -v_{k-1}^T, -v_{k-2}^T, y_{M,k+1}^T]^T$$

Equation (10) is the error equation used to update adaptive parameter of FEL. If Γ is a matrix that satisfy the equation

$$\Gamma KB_0 + (\Gamma KB_0)^T - (\Gamma KB_0)^T \Gamma KB_0 > 0 \quad (11)$$

and $\Gamma > 0$ then using the following adaptive control law and parameter update law, the output error will asymptotically close to zero.

(Algorithm I)

$$\text{Control Law: } v_k = v_{FB,k} + v_{FF,k}, \quad v_{FB,k} = K e_k, \quad v_{FF,k} = \hat{\Theta}_k \bar{\varphi}_k \quad (12)$$

Parameter Update Law:

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} - \Gamma u_{FB,k} [1 + \bar{\varphi}_k^T \bar{\varphi}_k]^{-1} \bar{\varphi}_k^T \quad (13)$$

$$\text{Where } \bar{\varphi}_k = [-y_{M,k}^T, -y_{M,k-1}^T, -y_{M,k-2}^T, -v_{FF,k-1}^T, -v_{FF,k-2}^T, y_{M,k+1}^T]^T$$

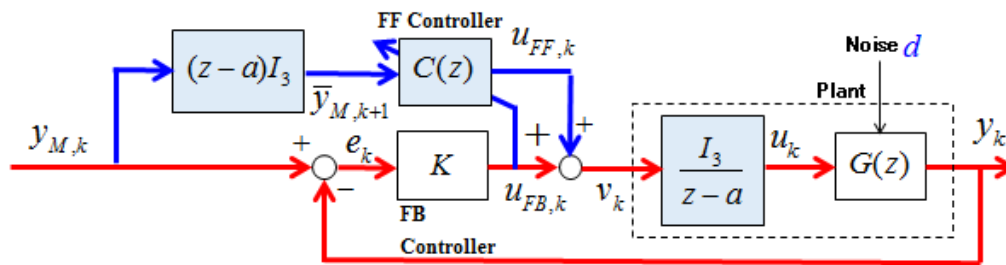


Figure 3. FEL Engine Control

For Algorithm II, dead zone method will be included in the controller.

(Algorithm II)

Parameter Update Law:

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} - \Gamma \bar{u}_{FB,k} [1 + \bar{\varphi}_k^T \bar{\varphi}_k]^{-1} \bar{\varphi}_k^T \quad (14)$$

The adaptive law which utilizes dead zone is formulated as shown below.

$$\bar{u}_{FB,k} = [D(u_{FB1,k}) \quad D(u_{FB2,k}) \quad D(u_{FB3,k})]^T$$

$$D(u_{FBj,k}) = \begin{cases} u_{FBj,k} - d_{oj} & , u_{FBj,k} > d_{oj} \\ 0 & , |u_{FBj,k}| \leq d_{oj} \\ u_{FBj,k} + d_{oj} & , u_{FBj,k} < -d_{oj} \end{cases}, \quad j = 1, 2, 3$$

Dead zone d_{oj} width is chosen by observing prior simulation. Ideally, in linear system represented by equation (3) and (4) with $C(z)$ as FF controller or inverse model of system, then statement below is true.

$$G(z) \cdot \frac{I_3}{z-a} \cdot C(z) \cdot zI_3 = I_3$$

Then to obtain the initial condition of FF controller $C(z)$, can be calculated by formula below.

$$C(z) = (z-a)I_3 \cdot G^{-1}(z) \cdot \frac{I_3}{z} = \frac{z-a}{z} G^{-1}(z)$$

Inverse model of the system $G^{-1}(z)$, can be calculated as shown below.

$$G^{-1}(z) = \left[\begin{array}{c|c} A - BD^{-1}C & BD^{-1} \\ \hline -D^{-1}C & D^{-1} \end{array} \right]$$

5. Kernel-based FEL Engine Control

In predictive control configuration such as adaptive control and machine learning, overtraining and overfitting might occur. In the previous section, it is stated that error caused by linear approximation based on equation (8) can be reduced by parameter adjustment. In this section, Kernel-based Adaptive FEL will be derived without applying direct linearization on equation (1) and (2). The method of derivation is based on Kernel Multivariable Analysis in [11].

Nonlinear state space representation of equation (1) and (2) can be rewritten into input-output relationship as below.

$$y_{k+1} = F(y_k, y_{k-1}, y_{k-2}, v_k, v_{k-1}, v_{k-2}) \quad (15)$$

Taking equation (3) into account then, equation (15) could be rewritten into equation below.

$$y_{k+1} = G(\zeta_k)[u_k + F(\zeta_k)] + \Delta_{1,k} \quad (16)$$

Where $\zeta_k = [y_k^T, y_{k-1}^T, y_{k-2}^T, v_{k-1}^T, v_{k-2}^T]^T$, $G(\zeta_k): R^{15} \rightarrow R^{3 \times 3}$ is regular matrix function corresponding to the matrix B_0 in Adaptive FEL in previous section, $\Delta_{1,k}$ is modeling error between equation (15) and (16), and $F(\zeta_k): R^{15} \rightarrow R^3$ is a smoothing vector function. Output of FB controller then can be represented as equation below.

$$\begin{aligned} v_{FB,k} &= K[y_{k+1} - y_{M,k+1}] = -K[G(\zeta_k)[v_k + F(\zeta_k)] + \Delta_{1,k} - y_{M,k+1}] \\ &= -KG(\zeta_k)[v_k + F(\zeta_k) - G^{-1}(\zeta_k)y_{M,k+1}] - K\Delta_{1,k} \\ &= -KG(\zeta_k)v_{FB,k} - KG(\zeta_k)[v_k - v_{FB,k} + F(\zeta_k) - H(\zeta_k)y_{M,k+1}] - K\Delta_{1,k} \end{aligned} \quad (17)$$

With $H(\zeta_k) = G^{-1}(\zeta_k)$, and nonlinear function $F(\zeta_k)$, $H(\zeta_k)$ is kernel representation of equation (17).

$$F(\zeta_k) = \Theta_F \Phi_F(\zeta_k), \quad H(\zeta_k) = \Theta_H \Phi_H(\zeta_k) \quad (18)$$

Where:

$$\Phi_F(\zeta_k) = [\kappa(\zeta_{k-L+1}, \zeta_k), \dots, \kappa(\zeta_k, \zeta_k)]^T \in R^{L \times 1}, \Theta_F \in R^{3 \times L}$$

$$\Phi_H(\zeta_k) = \text{blockdiag}[\Phi_{F,k}, \Phi_{F,k}, \Phi_{F,k}]^T \in R^{L \times 3}, \Theta_H \in R^{3 \times L}$$

$$\kappa(x, y) = \exp(-\rho \|x - y\|^2) : \text{Gaussian Kernel}$$

Parameter of Gaussian kernel ρ is set to $\rho = 0.01$ and dataset L is set to $L = 3$. By substituting equation (18) to equation (17) then the following equation can be obtained.

$$\begin{aligned} v_{FB,k} &= -KG(\zeta_k)v_{FB,k} - KG(\zeta_k)[v_k - v_{FB,k} + \Theta_F \Phi_F(\zeta_k) - \Theta_H \Phi_H(\zeta_k)y_{M,k+1}] - K\Delta_{1,k} \\ &= -KG(\zeta_k)v_{FB,k} - KG(\zeta_k)[v_k - v_{FB,k} + \Theta \phi_k] - K\Delta_{1,k} \end{aligned} \quad (19)$$

$\Theta = [\Theta_F \ \Theta_H]$ is the unknown parameter to be estimated and updated. Kernel FEL system will have control law and parameter update law as shown below.

(Algorithm III)

Adaptive Control Law:

$$\begin{aligned} v_k &= v_{FB,k} + v_{FF,k}, v_{FB,k} = Ke_k \\ v_{FF,k} &= -\hat{\Theta}_{F,k} \Phi_F(\zeta_k) + \hat{\Theta}_{H,k} \Phi_H(\zeta_k)y_{M,k+1} \\ v_{FF,k} &= [\hat{\Theta}_{F,k} \ \hat{\Theta}_{H,k}] \begin{bmatrix} -\Phi_F(\bar{\zeta}_k) \\ \Phi_H(\bar{\zeta}_k)y_{M,k+1} \end{bmatrix} = \hat{\Theta}_k \bar{\phi}_k \end{aligned} \quad (20)$$

Where:

$$\bar{\zeta}_k = [y_{M,k}^T, y_{M,k-1}^T, y_{M,k-2}^T, v_{FF,k-1}^T, v_{FF,k-2}^T]^T$$

$\hat{\Theta}_k = [\hat{\Theta}_{F,k} \quad \hat{\Theta}_{H,k}]$ is estimated value of $\Theta = [\Theta_F \quad \Theta_H]$, the result from the following update law.

Parameter Update Law:

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} - \Gamma u_{FB,k} [1 + \bar{\phi}_k^T \bar{\phi}_k]^{-1} \bar{\phi}_k^T \quad (21)$$

Adaptive gain Γ need to satisfy both $\Gamma > 0$, and equation below.

$$\Gamma KG(\zeta_k) + (\Gamma KG(\zeta_k))^T - (\Gamma KG(\zeta_k))^T \Gamma KG(\zeta_k) > 0 \quad (22)$$

Equation (22) is corresponding to adaptive FEL equation (11).

(Algorithm IV)

In this algorithm, dead zone method is included in the parameter update law. Equation (21) then can be replaced with equation below.

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} - \Gamma \bar{u}_{FB,k} [1 + \bar{\phi}_k^T \bar{\phi}_k]^{-1} \bar{\phi}_k^T \quad (23)$$

Dead zone width is the same as the one used in adaptive FEL in previous section.

6. Neural Network FF Controller

To control nonlinear plant used in this paper, FEL controller is using Kernel trick to model the nonlinearity. However, choosing the appropriate kernel function is quite a problem. In this section, as an alternative neural network is used to approximate the nonlinear function of the plant. The neural network structure used here is consisted of 3 layers; input, hidden, and output layer, as shown in figure 4.

(Algorithm V)

$$\text{Control law: } v_k = v_{FB,k} + v_{FF,k}, v_{FB,k} = K e_k \quad (24)$$

$$\text{Output layer (} \ell = 1, 2, 3 \text{) : } v_{FF,k} = f_{out}(\sum_{j=1}^{15} w_{\ell j} o_j + \theta_{\ell}) \quad (25)$$

$$\text{Hidden layer (} j = 1, \dots, 15 \text{) : } o_j = f_{hidden}(\sum_{i=1}^3 w_{ji} y_{Mi,k+1} + \theta_j) \quad (26)$$

The activation function used here are $f_{out} = x$: linear function and $f_{hidden}(x) = \frac{1}{1+\exp(-x)}$: sigmoid function.

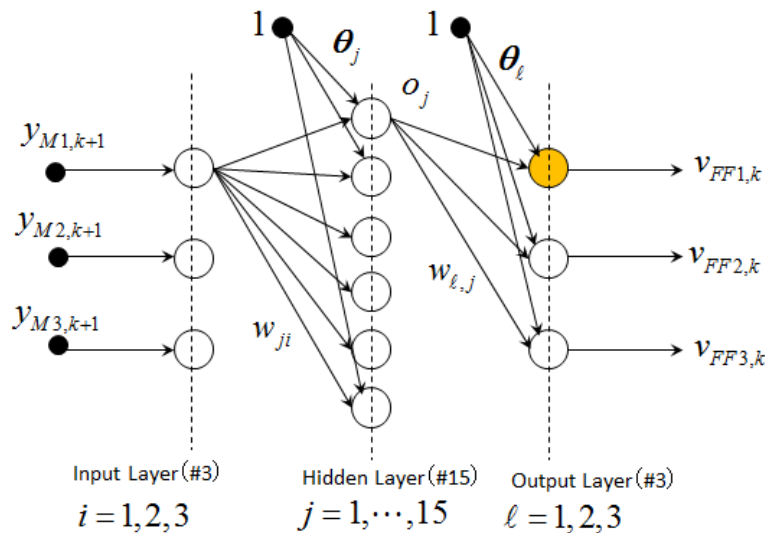


Figure 4. Schematic of a 3 layer Neural Network

Bias and weight are updated by gradient descent backpropagation algorithm.

Parameter Update Law:

Output layer:

$$w_{\ell j,k} = w_{\ell j,k-1} - \eta \frac{\partial E}{\partial w_{\ell j}}, \eta = 0.01 \quad (27)$$

Hidden layer:

$$w_{ji,k} = w_{ji,k-1} - \eta \frac{\partial E}{\partial w_{ji}}, \eta = 0.01 \quad (28)$$

Where $E = \|v_{FB,k}\|^2$. The initial weight for neural network is determined by offline training / indirect learning mechanism.

7. Simulation

The plant used in simulation is represented by nonlinear model in equation (1) and (2). The simulation duration is 200 engine cycles, and the target reference is changed as below. $[0.5 \ 0.2 \ 0]^T$

$$\Delta y_{M,k} = \begin{cases} [0.5 \ 0.2 \ 0]^T & k = 40 \\ [-0.5 \ -0.2 \ 0]^T & k = 80 \\ [0.3 \ 0.15 \ 0]^T & k = 120 \\ [-0.5 \ -0.2 \ 0]^T & k = 160 \end{cases},$$

FB Controller used for algorithm I~IV is set $K = 1.2I_3$ as for algorithm V is set $K = \text{diag} [0.5 \ 0.35 \ 0.1]$. Table 2 shows the summary of simulation done in this section.

Table 2. Summary of FEL Control System

	Adaptive FEL Controller		Kernel FEL Controller		Neural Network
	Algorithm I	Algorithm II	Algorithm III	Algorithm IV	Algorithm V
No Disturbances	Figure 5	Figure 8	Figure 10	Figure 12	Figure 16
Disturbances	Figure 6	Figure 9	Figure 11	Figure 13	Figure 17

The adaptive gain Γ for algorithm I~IV is set to $\Gamma = aB_0^{-1} = 0.16B_0^{-1}$. The initial parameter in algorithm I and II is set to $\hat{\Theta}_0 = O_{3 \times 18}$. As for algorithm III and IV, the initial parameter is set as following.

$$\hat{\Theta}_{F,0} = O_{3 \times L}$$

$$\hat{\Theta}_{H,0} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Disturbances stated in table 2 is white noise averaged at 0, with variance value of 0.1 in the right hand part of equation (2) (at system output). Dead zone band is set at $d_{0j} = 0.01$.

In figure 5 and 6 are shown system output when no disturbance and with disturbance respectively when controlled by Algorithm 1. Figure 8 and 9 shows the system output from algorithm II, with no and with disturbance respectively. By comparing results in figure 6 and 9, it can be seen that dead zone method increase robustness of the system.

Figure 10 and 11, shows the result of algorithm III without and with disturbance respectively. Figure 13 and 14, shows the result of algorithm IV, also without and with disturbance respectively.

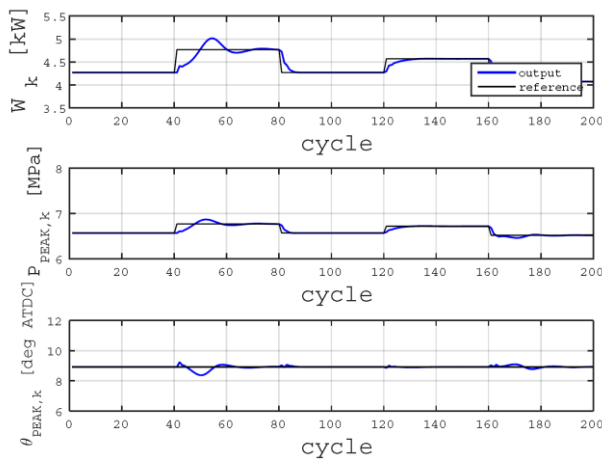


Figure 5. Plot of Plant Output : Algorithm I with no disturbances

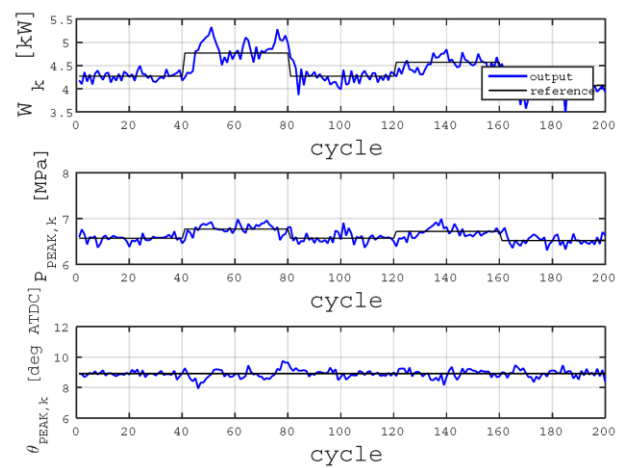


Figure 6. Plot of Plant Output : Algorithm I with disturbances

On figure 6 it can be seen that despite disturbed by noise, the controller manage to control plant output. Figure 7 shows the changes in adaptive parameter of FF controller.

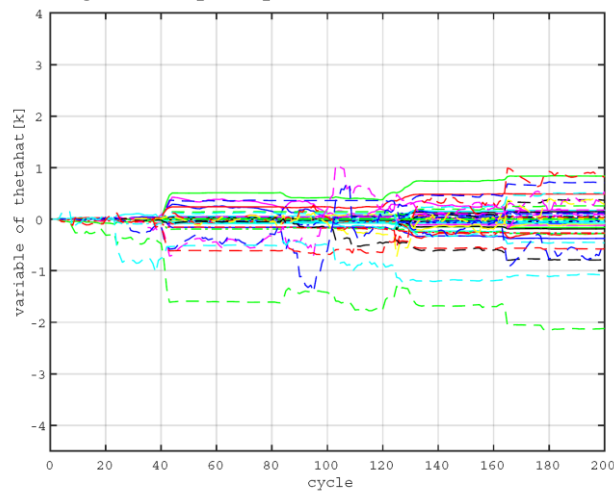


Figure 7. Plot of Parameter Changes : Algorithm I with disturbances

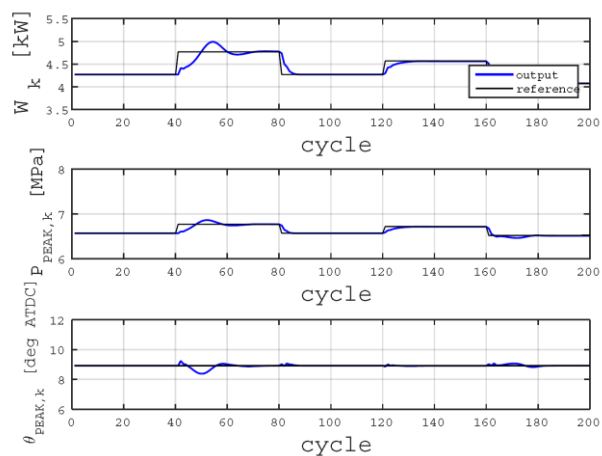


Figure 8. Plot of Plant Output : Algorithm II with no disturbances

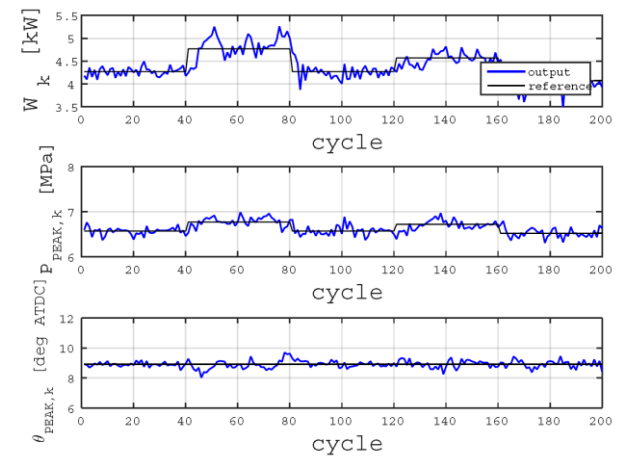


Figure 9. Plot of Plant Output : Algorithm II with disturbances

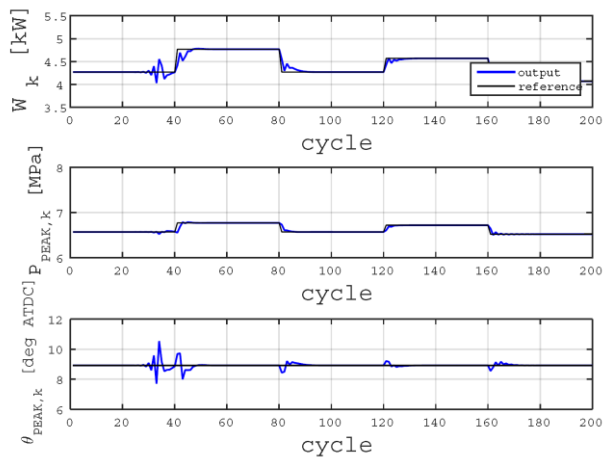


Figure 10. Plot of Plant Output : Algorithm III with no disturbances

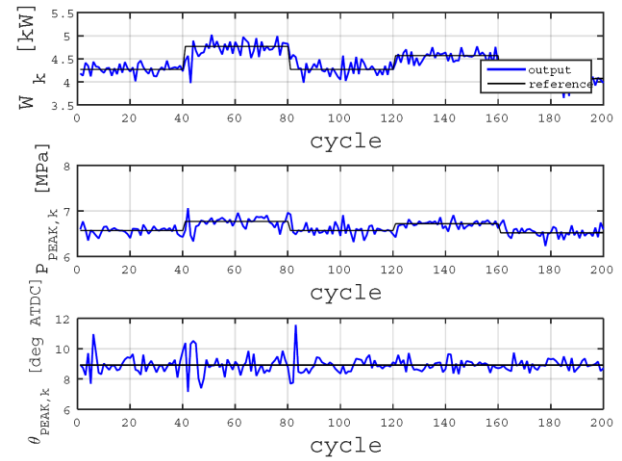


Figure 11. Plot of Plant Output : Algorithm III with disturbances

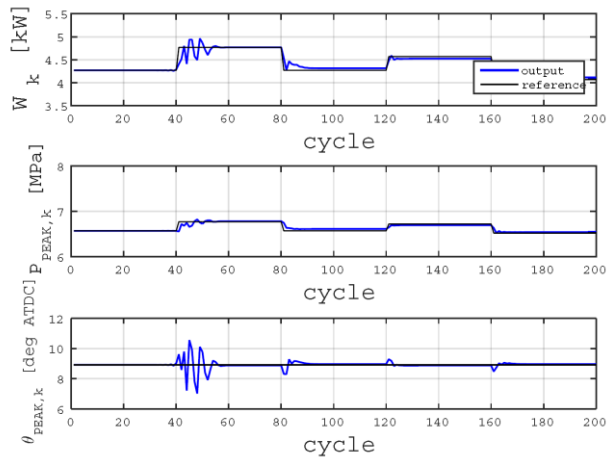


Figure 12. Plot of Plant Output : Algorithm IV with no disturbances

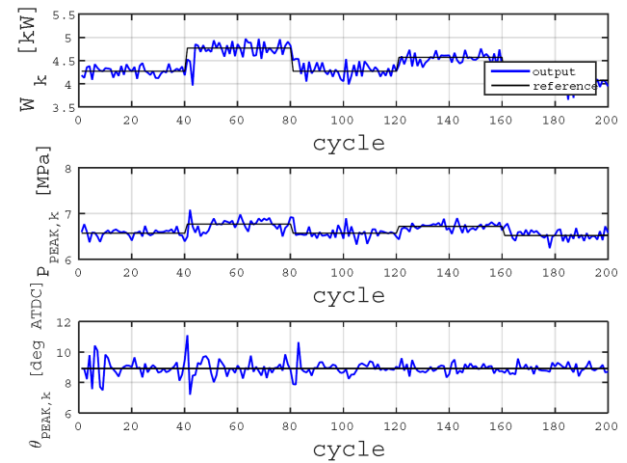


Figure 13. Plot of Plant Output : Algorithm IV with disturbances

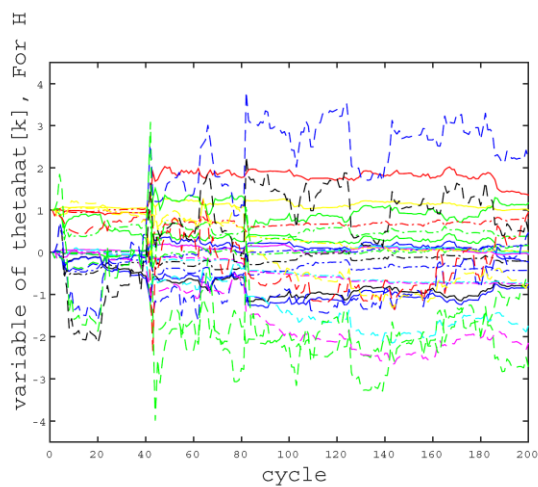


Figure 14. Plot of Parameter Changes : Algorithm III with disturbances

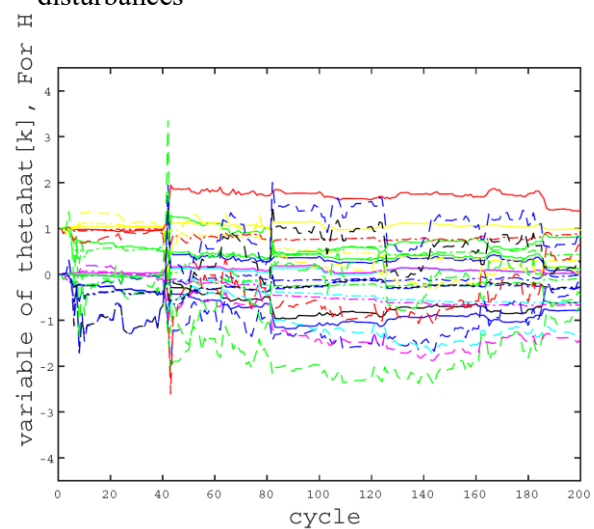


Figure 15. Plot of Parameter Changes : Algorithm IV with disturbances

By comparing figure 5 and figure 8, dead band have no effect when disturbance is not present in the system. When disturbances present, dead zone suppress the parameter changes.

Difference between standard FEL controller and Kernel based FEL controller can be seen by comparing figure 5 and figure 10. Kernel based controller will have slight violent oscillation around the first step signal, but followed by faster transient response.

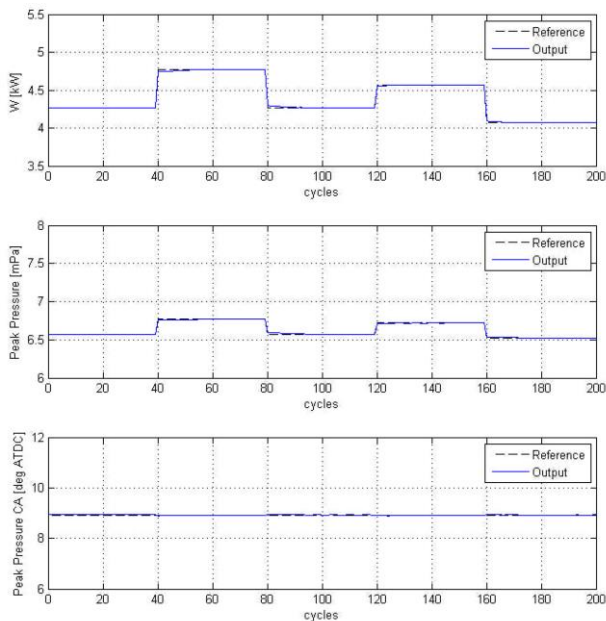


Figure 16. Plot of Plant Output : Algorithm V with no disturbances

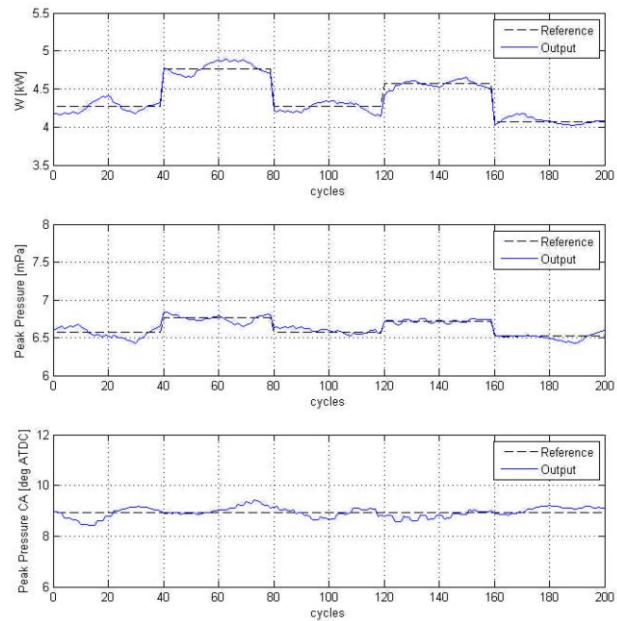


Figure 17. Plot of Plant Output : Algorithm V with disturbances

Comparing the parameter changes in figure 14 and figure 15, the effect of dead zone suppressing parameter changes in controller can be seen.

The result of algorithm V which is using neural network to model inverse plant of the model can be seen in figure 16 and 17. The neural network controlled plant had good tracking capabilities even at the first step of the signal when compared to algorithm I~IV. This is assuming the neural network is already well trained by offline training prior to simulation.

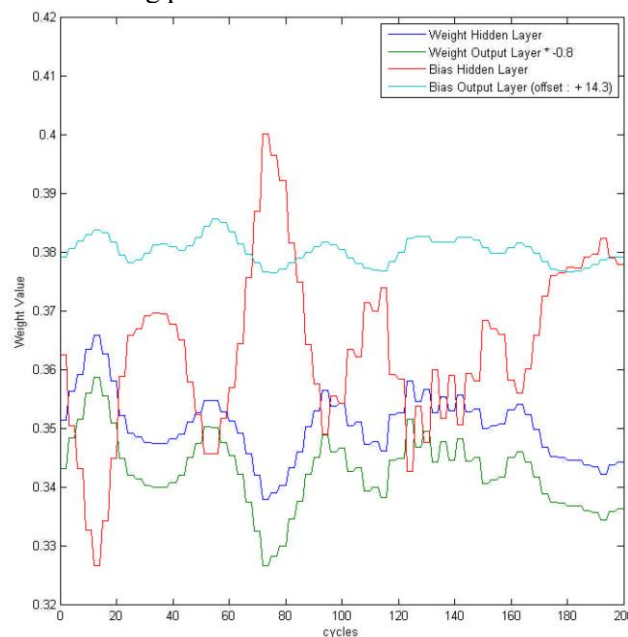


Figure 18. Plot of Plant Output : Algorithm V with disturbances

Neural network in algorithm V have 108 weight and bias parameters in total. Among all 108, 4 are chosen and shown in figure 18. When disturbance are applied, the weight is changed albeit slightly, showing that learning abilities through disturbance.

FB controller in this paper is set as fixed gain, but to increase robustness against stochastic disturbance, expansion toward dynamic FB controller is needed.

8. Summary

FEL Controller design is applied to onboard diesel combustion Model by Yasuda et.al. ^{[1][2]} In this paper there are 5 types of algorithm used as FF controller design. The existence of FB controller is used to adjust FF controller parameter by multiplying it with output error. In FEL configuration, role division are necessary, FF controller are mainly used to correct modelling error of the plant, while FB controller provides robustness against disturbances.

9. References

- [1] Kazuma Yasuda, Yudai Yamasaki, Shigehiko Kaneko, Yusuke Nakamura, Norimasa Iida, Ryo Hasegawa, "Diesel Combustion Model for On-board Application" International Journal of Engine Research, doi:10.1177/1468087415611331, October 26, (2015)
- [2] Fumiya Shimizu, Kazuma Yasuda, Yudai Yamasaki, Shigehiko Kaneko, Yusuke Nakamura, Norimasa Iida, Ryo Hasegawa, "Discrete Dynamics Model for Diesel Combustion, Second report: Expansion of the Model to Multi-Injection(Two Stage Injection)," The 25th Internal Combustion Engine Symposium, (The Japan Society of Mechanical Engineers, Tsukuba, 2014) CD-ROM (2014) (in Japanese)
- [3] Ryosuke Ikemura, Yudai Yamasaki, Shigehiko Kaneko, "Modeling of Diesel Engine for Model Based Control," The 58th Japan Joint Automatic Control Conference, Kobe University, November, 14-15 (2015) (in Japanese)
- [4] Yuji Ikutake, Xinyu Zhang, Kouhei Tanaka, "Combustion Control of Diesel Engine Based on Feedback Error Learning," The 58th Japan Joint Automatic Control Conference, Kobe University, November, 14-15 (2015) (in Japanese)
- [5] Ikuroh Mizumoto and Jyunpei Tsunematsu, "Combustion Control System Design of Diesel Engine via ASPR based on Output Feedback," The 58th Japan Joint Automatic Control Conference, Kobe University, November, 14-15 (2015) (in Japanese)
- [6] Mitsuo Hirata, Souta Ishizuki and Masayasu Suzuki, "H infinity Control of Combustion in Diesel Engines using a Discrete Dynamics Model," The 58th Japan Joint Automatic Control Conference, Kobe University, November, 14-15 (2015) (in Japanese)
- [7] D. M. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the Cerebellum," Trends in Cognitive Sciences, Vol.2 pp. 338-347 (1998)
- [8] Hiroaki Gomi and M. Kawato: Learning Control for Closed Loop Systems based on Feedback-Error Learning, Transactions of the Institute of Systems, Control and Information Engineers, Vol. 4, No. 1, pp. 37-47 (1991) (in Japanese)
- [9] G. C. Goodwin et al., Discrete-Time Multivariable Adaptive Control, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. AC25, NO. 3, JUNE (1980)
- [10] G. C. Goodwin and R. S. Long: Generalization of Results on Multivariable Adaptive Control, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. AC25, NO. 6, DECEMBER (1999)
- [11] Shotaro Akaho: Kernel Multivariable Analysis, Iwanamishoten, (2008) (in Japanese)

Acknowledgments

This research is done as part of "Control and System Modelling on Fuel Technology Innovation" (Research Leader: Shigehiko Kaneko of Tokyo University) of SIP (Strategic Innovation Program) under Japan Science and Technology Agency (JST).