Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

On-line Feedforward Map Generation for Engine Ignition Timing Control

Shunpei Tamaki * **Yoshihiro Sakayanagi** **
Kazuma Sekiguchi *** **Tatsuya Ibuki** * **Kohei Tahara** ****
Mitsuji Sampei *

* *Tokyo Institute of Technology, Tokyo, Japan (e-mail: sampei@ctrl.titech.ac.jp)*
** *Toyota Motor Corporation, Shizuoka, Japan (e-mail: yoshihiro_sakayanagi@mail.toyota.co.jp)*
*** *Tokyo City University, Tokyo, Japan (e-mail: ksekiguc@tcu.ac.jp)*
**** *Panasonic Corporation, Osaka, Japan (e-mail: tahara.kohei@jp.panasonic.com)*

Abstract: This paper newly proposes an on-line learning method for control maps by using Gaussian filters. In the present method, man-hours for calibration of control maps can be decreased, and complicated structures of control maps are learned without prior knowledge. Moreover, by the effect of Gaussian filtering, smoothed maps can be created even under noisy conditions or few measured points. We also introduce improvements of the algorithm to cope with engine deterioration due to aging. In this work, the proposed method is applied to minimum advance for best torque control on actual vehicles with just one driving data, and the accuracy of the learned map is verified through simulation and experiments.

Keywords: Engine control; Spark advance control; Online calibration; Feedforward control; Learning control

1. INTRODUCTION

High performances of engine systems (e.g. Lower emission or better gas mileage) have been of great need for recent environmental concerns (Manizie et al. [2007], Wang [2008], Saerens et al. [2009], Hsieh and Wang [2011]). To satisfy various demands, we are often required to control many actuators optimally by taking account of many engine conditions such as intake/exhaust valve timing (Moriya et al. [1996], Ma et al. [2007]). In many actual engine systems, feedforward control plays a central role in which we have to determine control inputs for each engine condition, such as engine speed, torque, in advance. Throughout this paper, we call the condition-input table *control map*. Although the control map is efficient for control, it is necessary to be careful about the fact that the number of the control map axes greatly increases when we handle sophisticated engine control, and as a result, man-hours for determining the control map increases exponentially under the development.

This paper focuses especially on ignition timing as one of the parameters to be optimized. The optimization of ignition timing provides Minimum Advance for Best Torque (MBT) (Zhu et al. [2007]) values, which is effective for lower emission or better gas mileage. Since MBT values change depending on engine speed and charging efficiency, we have the control map representing MBT for each engine condition (MBT-map). In current productions, they are experimentally determined by conducting spark sweeps at every operating condition. We consider this case as an example of feedforward engine control.

Numerous research works have been devoted to learning methods of engine control values. The authors in Leonhardt et al. [1999], Park et al. [2001] claim that Radial Basis Function Neural Networks are effective to learn feedback errors of MBT. However, they are used secondarily only for like offset corrections. Feedback control methods seem to be good as in Olsson et al. [2001], Zhu et al. [2007] since it does not rely on any control maps. But it requires accurate models of engines, and engine models are generally so complicated that many man-hours are necessary for modeling. Moreover, we have to measure engine states by using various sensors in feedback control (Eriksson et al. [1997], Upadhyay and Rizzoni [1998]). It is also necessary to be sensor fail-safe systems for the sake of high reliability of engine systems. In view of these facts, we apply control maps to feedforward control in this work.

In this paper, we newly present an on-line learning method for control maps with Gaussian filtering for random data set. Our method does not need to design learning data set. To the best of our knowledge, there exists no method for learning control maps under this situation. The proposed method enables us to reduce man-hours since it can automatically learn necessary and sufficient regions of control maps with just one driving mode. The Gaussian filter used in this work is motivated by image processing (Tomasi and Manduchi [1998]). This filter can create smoothed maps even under noisy conditions or few measured points. Moreover, a complicated structure can be learned without prior knowledge. We evaluate the validity of our method by simulation and experimental results compared with our

previous method. The main contributions of this paper are as follows: (i) We propose a novel algorithm to learn MBT values automatically with only one driving data. (ii) We moreover provide improvements of the algorithm taking account of engine aging and characteristics of MBT-maps. (iii) We perform the experiment with actual car systems in order to confirm the effectiveness of the proposed control algorithm.

The remainder of this paper is organized as follows. We first introduce our target engine system and experimental system in Section 2. In Section 3, we state the problem setting and propose on-line control map generation algorithms. Section 4 gives additional elements for the present methods to handle machine aging and accuracy deterioration. The verification through simulation and experiments are given in Section 5. We finally draw conclusions in Section 6.

2. EXPERIMENTAL SYSTEM AND ENGINE MODEL

Throughout this paper, in addition to the design of a new learning algorithm for automatically determining control maps, we focus on the application to a real system. Therefore, we first introduce our experimental system for engine control in this section. We especially consider MBT control of engines, and the present algorithm can be also applied to other actuators.

2.1 MBT control

We consider ignition timing of engines. Ignition timing is the timing when ignition occurs in a combustion chamber near the end of the compression stroke. It is represented by crankshaft angles [°CA] before top the dead center [°CA BTDC]. On the other hand, crankshaft angles after the top of dead center is represented by [°CA ATDC]. Correct setting of MBT has a significant impact on fuel consumption. However, MBT changes depending on the engine speed (ne [r/min]) and charging efficiency (kl [%]). Accordingly, we have to determine MBT with respect to each ne and kl for good gas mileage.

In the conventional control method, the MBT value is determined by prepared control maps. A control map is grid point data representing the optimal MBT value for each driving condition. It consists of ne axis and kl axis, and each grid point has an MBT value. Therefore, we consider the ne, kl-MBT control map (called MBT-map for simplicity). Fig. 1 shows the flow of the control with MBT using the MBT-map. Our goal is to identify the MBT-map by on-line learning from the initial value 0°CA BTDC. The crank angles of the engine when the combustion rate after ignition (Qr) becomes 50% is called 50% combustion point ($CA50$ [°CA ATDC]). It is empirically known that the ignition timing for $CA50$ at 8°CA becomes MBT (Leonhardt et al. [1999]).

2.2 Detection of combustion rate by cylinder pressure sensors

Fig. 2(a) shows a detection method of the combustion rate. According to the first law of thermodynamics and the ideal gas law, the amount of a heat quantity Q [J] at a crank angle θ [°CA ATDC] is derived as follows.

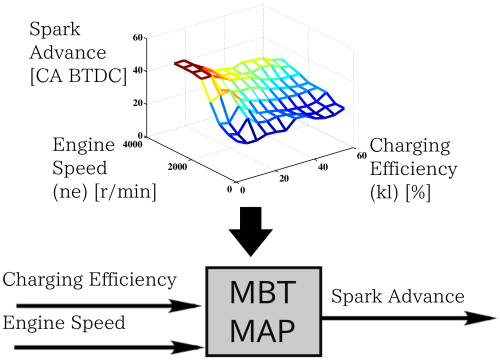
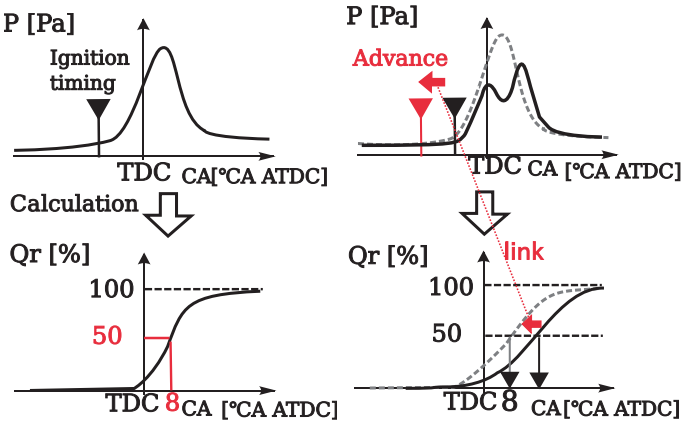


Fig. 1. Spark advance controller



(a) Calculation from cylinder pressure

(b) Spark advance control

Fig. 2. Combustion rate

$$dQ = dU + dW = \frac{1}{\kappa - 1} d(PV) + PdV, \tag{1}$$

$$Q(\theta) = \frac{1}{\kappa - 1} [PV]_{ivc}^{\theta} + \int_{ivc}^{\theta} PdV. \tag{2}$$

Here, P [Pa] is cylinder pressure, U [J] is internal energy, W [J] is work, κ is a heat capacity ratio, and ivc [°CA ATDC] is the intake valve timing. Let evo [°CA ATDC] be exhaust valve open timing. Then, from the heat quantity of one explosion $Q(evo)$, the combustion ratio Qr [%] is obtained as follows,

$$Qr(\theta) = Q(\theta)/Q(\text{evo}) \times 100. \tag{3}$$

We explore MBT by the simple method as shown in Fig. 2(b). Namely, if the heat quantity is more (less) than 50 % when the crank angle is 8 degree, then we retard (advance) the combustion timing.

2.3 Experimental system

We equip a cylinder pressure sensor on an experimental vehicle. Also, the learning method proposed in Sec. 3 is implemented in an engine control unit. Then, learning of the MBT-map is conducted while actual driving. Fig. 3 illustrates the experimental system and Tab. 1 shows experimental vehicle conditions. We use the front engine rear drive (FR) Sedan with V6 gasoline engine. LA#4 is the EPA Urban Dynamometer Driving Schedule (UDDS) representing city driving conditions. Time series data on

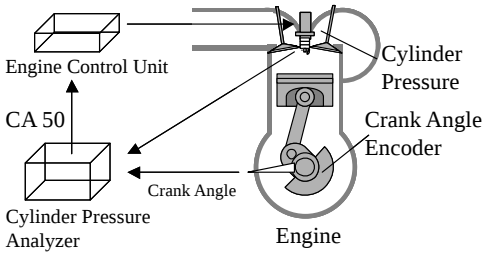


Fig. 3. Spark advance control system

Table 1. Experimental vehicle

Experimental vehicle	FR Sedan
Engine	V6 gasoline
Driving mode	LA#4

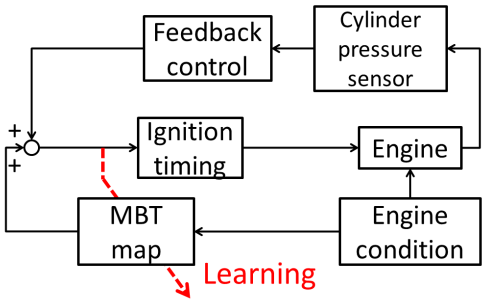


Fig. 4. Control flow of learning method

the actual driving is shown in Fig. 8. In this work, we demonstrate the effectiveness of our algorithm through this experimental system.

3. ON-LINE LEARNING ALGORITHM OF CONTROL MAP

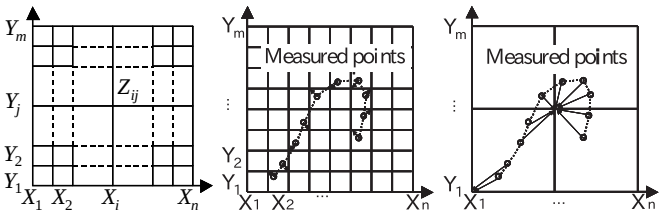
3.1 Formulation of control maps

We first represent engine speed, charging efficiency and MBT by x, y and z respectively. In the online map learning method, the MBT value is acquired by feedback control as described in Sec. 2. The learning method including MBT control is shown in Fig. 4. The relationship between (x, y) and z is represented by a two dimensional pattern, and we divide x, y plane into the grid pattern. Each grid has a value of z. We define axes of the map are $X_i, Y_j, i = 1, \ldots, n,\ j = 1, \ldots, m$ corresponding to the value Z_{ij} as shown in Fig. 5(a). Then, the on-line MBT-map learning problem is defined as calculating Z_{ij} from time-series data $[\zeta_1\ \zeta_2\ \cdots\ \zeta_n] \in \mathbb{R}^{3\times n}$, $\zeta_k = [x_k\ y_k\ z_k]^{\mathrm{T}} \in \mathbb{R}^3$.

3.2 Previous learning method

To the best of our knowledge, there exist a few methods to learn MBT-maps on-line. Our previous work applies a discrete time low-pass filter to the nearest grid point (Miyashita et al. [2006]). Namely, this algorithm chooses the nearest point Z_{ij} from measured data, and then updates Z_{ij} with the following smoothing process.

$$Z_{ij}(k) = Z_{ij}(k-1) + \frac{z_k - Z_{ij}(k-1)}{M}. \tag{4}$$



(a) Grid point (b) Fine grid (c) Rough grid

Fig. 5. The conventional learning method

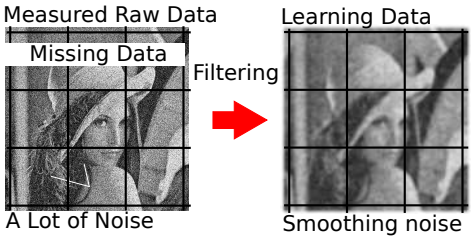


Fig. 6. Image of learning method

Here, $M > 0$ is a smoothing coefficient. In this method, there exists a trade-off between missing data and exact learning. That is, if the grid spacing is fine enough, then the number of data points for each grid decreases as shown in Fig. 5(b). Consequently, the MBT-map is not properly learned in the missing data area. We thus have to enlarge grid spacing to avoid this issue although we cannot learn complex characteristics, as indicated in the Fig. 5(c).

3.3 Basic idea of new on-line learning method

The preceding subsection points out the problem of the previous method. To solve this issue, we consider learning MBT-maps by using filtered measured data. Our basic idea is to regard measured data as images (Fig. 6). Then, filtering is expected to reduce noise influences and yield a complement effect of unmeasured points. As a result, the MBT-map can represent a complex structure with high accuracy.

In this study, we propose the learning method utilizing a Gaussian filter as such a image processing scheme. The filtered value Z_{ij} by a Gaussian filter is calculated by a weighted average of measured data for each grid. Then, measured data ζ_k is given, and the grid point value Z_{ij} is calculated as follows.

$$Z_{ij}(n) = \frac{\sum_{k=1}^{n} z_k w(x_k, y_k, X_i, Y_j)}{\sum_{k=1}^{n} w(x_k, y_k, X_i, Y_j)}. \tag{5}$$

Here, $w(x_k, y_k, X_i, Y_j)$ is the weight for each grid point, and the weight w is determined by a Gaussian function depending on the distance between the operating point and the grid point as follows.

$$w = \frac{1}{\sqrt{2\pi}\sigma_x} exp\left(-\frac{(x_k - X_i)^2}{2\sigma_x^2}\right) \cdot \frac{1}{\sqrt{2\pi}\sigma_y} exp\left(-\frac{(y_k - Y_i)^2}{2\sigma_y^2}\right). \tag{6}$$

Here, $\sigma_x, \sigma_y > 0$ are the standard deviations for (x, y) coordinates, respectively. Then, the learned MBT-map

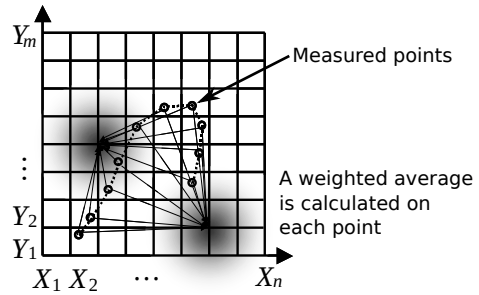


Fig. 7. Proposed learning method

becomes smooth according to setting higher value of σ_x and σ_y. Note here that the designer can set σ_x, σ_y arbitrarily to match the learning target. In order to learn the MBT-map for each measured data on-line, we make (5) possible to be calculated sequentially. We now define the numerator (denominator) of (5) as $N_{ij}(n)$ ($D_{ij}(n)$). Then, (5) is split into the following equations.

$$\begin{aligned} N_{ij}(n) &= \sum_{k=1}^{n} z_k w(x_k, y_k, X_i, Y_j) \\ &= N_{ij}(n-1) + z_n w(x_n, y_n, X_i, Y_j), \qquad (7) \\ D_{ij}(n) &= \sum_{k=1}^{n} w(x_k, y_k, X_i, Y_j) \\ &= D_{ij}(n-1) + w(x_n, y_n, X_i, Y_j). \qquad (8) \end{aligned}$$

Therefore, we can update them sequentially for each sample on-line by holding two (N_{ij} and D_{ij}) maps. The denominator D_{ij} map gives an indicator of learning frequencies for each grid point because D_{ij} represents the summation of the weights.

Claim 1. In the present algorithm, the update rule is simple and it is not necessary to memorize any past data at each step k. This learning method is thus applicable on-line. Additionally, the on-line scheme enables us to build a control map by using only one driving data. In summary, the present scheme can decrease man-hours.

3.4 Property of Gaussian filtering

In contrast to the previous method (4) updating the nearest one grid point from measured points, the present method (5) updates all grid points as shown in Fig. 7. Therefore, in spite of having small grid spacing, our method can learn complex characteristics of the MBT-map since the Gaussian filter has smoothing effects. Accordingly, the effect of noise or the bias of measured data is attenuated. Note that the minimum value of Z_{ij} in (5) is higher than that of input data z_k as shown below.

$$\begin{aligned} Z_{ij} &= \frac{\sum_k Z_k w(x_i, y_j, X_k, Y_k)}{\sum_k w(x_i, y_j, X_k, Y_k)} \\ &\geq \frac{\sum_k \min(Z_k) w(x_i, y_j, X_k, Y_k)}{\sum_k w(x_i, y_j, X_k, Y_k)} = \min(Z_k). \end{aligned}$$

Similarly, the maximum value of Z_{ij} in (5) is lower than that of input data z_k. Namely, although the MBT-map values change significantly at a data point around the grid point and are affected by the closest data value in most cases, the lower (higher) limit is the minimum (maximum) value of the measured data.

Table 2. Number of iteration until effect of the first data becomes 10% or less

λ	Number of Iteration
0.9	22
0.99	230
0.999	2302

4. IMPROVEMENT OF GAUSSIAN FILTERING

In this section, we introduce additional factors of the Gaussian filter to improve the filtering performance.

4.1 Forgetting factor

Application of forgetting factor In the present scheme, in spite of the fact that engines deteriorate over time, identification of the parameters uses all data. Namely, it is undesired that the past and current data is used to estimate parameters in the same degree of importance. We thus multiply past estimation values by a forgetting factor $(0 <)\lambda < 1$ to reduce the influence of past data. By introducing the forgetting factor λ, equations (7) and (8) are rewritten as follows.

$$N_{ij}(n+1) = z_{n+1} w(x_{n+1}, y_{n+1}, X_i, Y_j) + \lambda N_{ij}(n), \quad (9)$$

$$D_{ij}(n+1) = w(x_{n+1}, y_{n+1}, X_i, Y_j) + \lambda D_{ij}(n). \quad (10)$$

Tab. 2 shows examples of λ and the corresponding number of the iteration until the effect of the first data becomes 10% or less.

Range in application of forgetting factor Forgetting factors are intended to weaken the old data effect when new data enters. However, the forgetting factor affects the whole region by (9) and (10). Namely, the weight of the past learning becomes too small in the region where the number of data is not sufficiently large. The next measured data thus comes into the neighborhood of the grid point, and the impact of the data becomes too large. Consequently, robustness against noise might be lost. Therefore, we apply a small forgetting factor only to grid points around new data so that we restrict the range in application of the forgetting factor as follows.

$$\left(\frac{x_i - X_k}{x_{i+1} - x_i}\right)^2 + \left(\frac{y_j - Y_k}{y_{i+1} - y_i}\right)^2 \leq 2^2. \qquad (11)$$

Equation (11) means that the forgetting factor makes sense only when the data enters into the interior of an ellipse whose axis length is twice of the grid point spacing.

4.2 Clustering

So far, we consider Gaussian filtering as the weight for input data. We now note that it is also possible in (6) to obtain equivalent results by replacing X_i and Y_j by x_k and y_k, respectively. Let us consider that Gaussian functions have a fixed distribution to each grid point, and the weighted average of the map is updated when input data rides on the distribution of each point. According to this idea, we can change characteristics of the filter by adjusting σ_x and σ_y in (6) for point. Thus, if we desire to emphasize the data around the fluctuating area, then we should select small σ_x and σ_y.

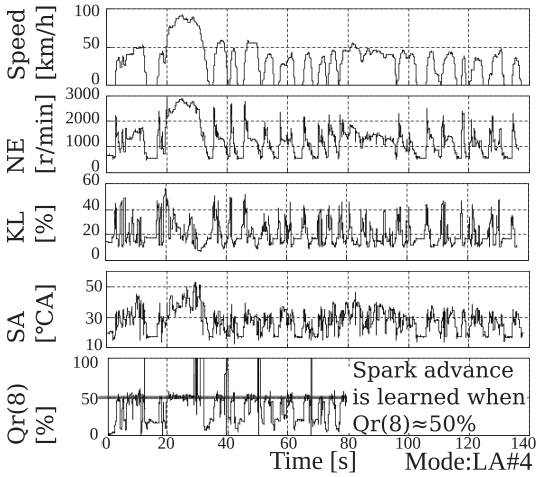


Fig. 8. ECU data (LA#4 mode)

5. EXPERIMENTS

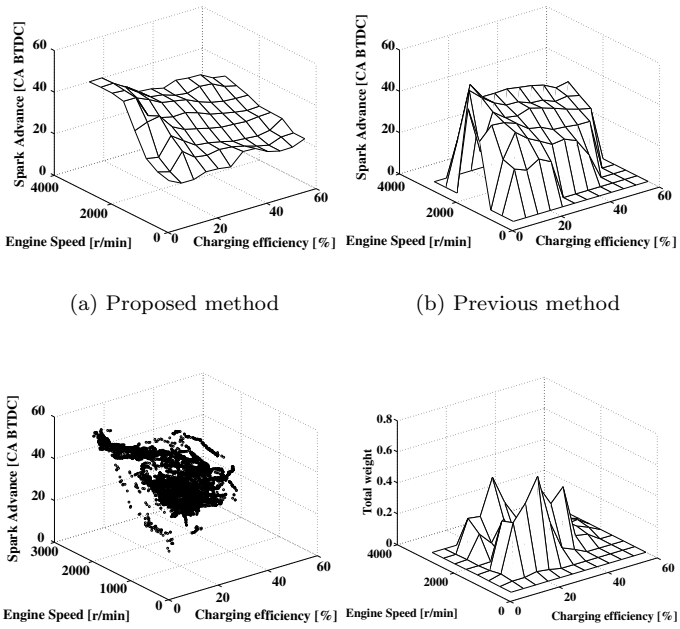
In this section, we demonstrate the effectiveness of the proposed MBT learning algorithm through experiments.

5.1 Test setting

Input data is the LA#4 driving mode shown in Fig. 8. We set the initial MBT-map as 0 at all the points. The MBT-map is composed of ne-axis and kl-axis, and each grid is divided into equal intervals, i.e. ne axis is $[400, 800, \ldots, 3200]$ and kl axis is $[5, 10, \ldots, 60]$. Numbers of ne and kl grids are 8 and 12, respectively. We now set $\sigma_{ne} = 100$ and $\sigma_{kl} = 1.25$, i.e. 1/4 of the grid intervals. In the previous method, we set $M = 2.3$ by seeking the best parameter. The number of data points, the running time and the number of the data satisfying $CA50 \approx 8$ are set as $40,000$, 800s and 11879, respectively. In order to see the effectiveness of our algorithms, we compare learning errors of the present method with the previous one. Here, a learning error is defined as the difference between the actual driving data and the learned MBT-map.

5.2 Results

The experimental results are shown in Fig. 9. Fig. 9(c) shows the measured data obtained during the LA#4 driving mode. Fig. 9(a) depicts the learning results using the method proposed in Sec. 3. We can see from this figure that the proposed method achieves almost the same fineness and shape of MBT-map as in the existing adaptation. Fig. 9(b) shows the learning results of the previous method described in Sec. 3. The obtained MBT-map can represent the overall trend through the rough grid points. However, since it cannot represent the details, the end points of the MBT-map missing values have not small errors, i.e. the learning is poor. Tab. 3 shows the comparison of the MBT-map accuracy between the proposed method and the existing adaptation. Our method is about 60 % better than the previous one with respect to the mean error and root-mean-square error (RMSE). Especially, in the region where the number of data points is large enough (see D_{ij} in Fig. 9(d)), the estimation is so good that the absolute values of errors are less than 1°CA. This effect is also confirmed in



(a) Proposed method (b) Previous method

(c) $CA50 \approx 8$ data (d) Total weight D map

Fig. 9. Experimental results

Table 3. Comparison of learning error

	Mean Error	RMSE	Maximum Error
Proposed Method	1.00	1.40	12.80, -8.66
Previous Method	1.67	2.34	8.4, -20.67

Table 4. Comparison of learning error with initial value

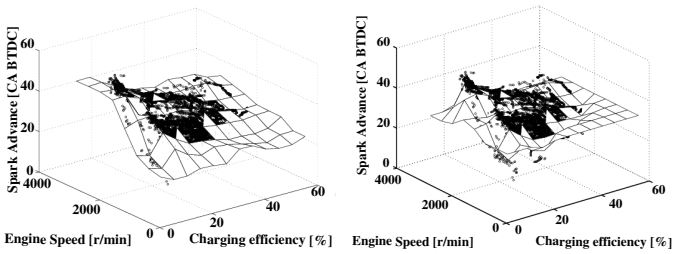
	Mean error	RMSE	Maximum error
Basic our method	1.12	1.77	13.89, -19.52
Forgetting factor	1.10	1.59	13.21, -17.14

Tab. 3 reducing the maximum errors. Thus, it can be said that sufficient accuracy is ensured for the present MBT-map control.

5.3 Improvement of the algorithm

Forgetting factor To check the effectiveness of forgetting factors introduced in Subsec. 4.1, we run another simulation. First, the initial data in MBT-map is set as all 30°CA BTDC. We now presume erroneous learning or change of the engine model. Here, erroneous learning means that we use the dummy data instead of the real data for the first $1,000$ iteration, and then we apply the actual driving data. Therefore, we can see the effectiveness of forgetting factors from this setting. We compare the basic learning method (Subsec. 3.3) with the algorithm having forgetting factors in terms of learning accuracy. Fig. 10 shows the results of the learned MBT-map. We see from the figure that the previous method yields not small errors in Fig. 10(a) around $kl = 20$ [%] and $ne = 1500$ [r/min], and the error is reduced in Fig. 10(b). Tab. 4 also represents the effectiveness numerically.

Clustering We finally investigate the effect of clustering σ_x, σ_y (Subsec. 4.2). We divide the MBT-map into four



(a) Basic method (b) Forgetting factor

Fig. 10. Forgetting factor

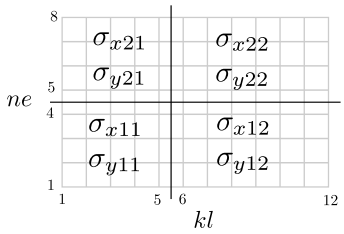


Fig. 11. Clustering of σ

Table 5. Comparison of learning error with initial value

	Mean error	RMSE	Maximum error
Basic our method	1.00	1.40	12.80, -8.66
Clustering	0.98	1.40	14.34, -8.07

clusters as shown in Fig. 11. Here we set $\sigma_{x11} = 0.72$, $\sigma_{y11} = 102$, $\sigma_{x12} = 0.53$, $\sigma_{y12} = 25$, $\sigma_{x21} = 0.64$, $\sigma_{y21} = 132$, $\sigma_{x22} = 1.67$, and $\sigma_{y22} = 97$. The other parameters are the same as in the basic proposed method. Tab. 5 shows the comparison of the MBT-map accuracy between the basic proposed method and the clustering one. Although this approach has an effect of error reduction, the difference is small. This might be because the small change of the data gradient yields the small effect of the clustering σ. One of future works is to find a real example suitable for this algorithm.

6. CONCLUSION

In this paper, we have developed on-line learning methods that can express complex characteristics of control maps with addressing noise and deviations of measurements. We have first introduced our experimental system for MBT control. Next, we have proposed learning algorithms to build MBT-maps, and confirmed the methods work successfully. In the experimental verification, almost the same shape and accuracy as in the existing adaptation have been obtained. As the main contribution, in contrast to the fact that the existing adaptation method needs a lot of man-hours, the proposed method enables to obtain a control map automatically.

The optimal control value is evaluated by a simple proportional method. However, in the case that control objects have local maxima, additional optimization methods have to be applied to seek the best parameter. Therefore, one of our future directions is to choose the best on-line optimization method to extend the present scheme.

ACKNOWLEDGEMENTS

We would like to express sincere gratitude for our colleagues in Toyota Motor Corporation. Their invaluable comments and help have greatly improved this work.

REFERENCES

C. Manzie, H. Watson and S. Halgamuge. Fuel economy improvements for urban driving: Hybrid vs. intelligent vehicles. *Transportation Research Part C*, 15(1):1–16, 2007.

J. Wang. Hybrid robust air-path control for diesel engines operating conventional and low temperature combustion modes. *IEEE Transactions on Control Systems Technology*, 16(6):1138–1151, 2008.

B. Saerens, J. Vandersteen, T. Persoons, J. Swevers, M. Diehl and E. Van den Bulck. Minimization of the fuel consumption of a gasoline engine using dynamic optimization. *Applied Energy*, 86(9):1582–1588, 2009.

M.F. Hsieh and J. Wang. A two-cell backstepping-based control strategy for diesel engine selective catalytic reduction systems. *IEEE Transactions on Control Systems Technology*, 19(6):1504–1515, 2011.

Y. Moriya, A. Watanabe, H. Uda, H. Kawamura, M. Yoshioka and M. Adachi. A newly developed intelligent variable valve timing system: Continuously controlled cam phasing as applied to a new 3-liter in-line-6 engine. *SAE Technical Paper*, No. 960579, 1996.

J. Ma, G. Zhu, H. Schock and J. Winkelman. Adaptive control of a pneumatic valve actuator for an internal combustion engine. *Proceedings of the 2007 American Control Conference*, pages 3678–3685, 2007.

G.G. Zhu, I. Haskara and J. Winkelman. Closed-loop ignition timing control for SI engines using ionization current feedback. *IEEE Transactions on Control Systems Technology*, 15(3):416–427, 2007.

S. Leonhardt, N. Muller and R. Isermann. Methods for engine supervision and control based on cylinder pressure information. *IEEE/ASME Transactions on Mechatronics*, 4(3):235–245, 1999.

S. Park, P. Yoon and M. Sunwoo. Feedback error learning neural networks for spark advance control using cylinder pressure. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 215(5):625–636, 2001.

J. Olsson, P. Tunestal and B. Johansson. Closed-loop control of an HCCI engine. *SAE Technical Paper*, No. 2001-01-1031, 2001.

L. Eriksson, L. Nielsen and M. Glavenius. Closed loop ignition control by ionization current interpretation. *SAE Technical Paper*, No. 970854, 1997.

D. Upadhyay and G. Rizzoni. AFR control on a single cylinder engine using the ionization current. *SAE Technical Paper*, No. 980203, 1998.

C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *Proceedings of the Sixth IEEE International Conference on Computer Vision*, pages 839–846, 1998.

S. Miyashita, H. Tanaka and H. Hokuto. Ignition timing control apparatus for internal combustion engine. *U. S. Patent 7,055,500, 2006-06-06*, 2006.