

Iterative Learning and Internal Model Control Applied to Diesel Engine Throttle

Withit Chatlatanagulchai^{1,*} and Anupab Kerdlam¹

¹ Control of Robot and Vibration Laboratory, Department of Mechanical Engineering, Faculty of Engineering, Kasetsart University, 50 Phahonyothin Rd., Chatuchak, Bangkok, 10900

*Corresponding Author: fengwtc@ku.ac.th, Tel. +66(0) 2797-0999 ext. 1803, 1804, Fax. +66(0) 2579-4576

Abstract

Iterative learning control is an adaptive, feed-forward, performance enhancing control technique. Its advantages include its simple algorithm, its stability without the need of persistently exciting input, and its being plant-model-independent. Internal model control is a feedback control technique. Its main advantage includes its ability to deal with plant having time delay or non-minimum phase. However, the internal model control requires accurate plant model. Model uncertainty can cause performance deterioration. In this paper, iterative learning control is used to improve the performance of the internal model control. Two practical schemes in combining the iterative learning control with internal model control are explored. The first scheme consists of an internal model control as feedback and a frequency-domain iterative learning control as a performance-enhancing feed-forward to reduce tracking error. The second scheme is model matching where the iterative learning control reduces the model output error. Based on actual experiments on a Diesel engine throttle in our laboratory, discussions are made regarding their performance comparison.

Keywords: Iterative learning control; Model reference; Throttle; Internal model control.

1. Introduction

Iterative learning control (ILC) was proposed by [1] in the name “betterment process”, based on an idea proposed in [2].

ILC can be viewed as an “add-on” feed-forward enhancement for plants that have repeated reference, repeated disturbance, and repeated initial conditions. The control effort from ILC learns from previous repetition, called “iteration”, to decay error exponentially to zero.

Because ILC is a feed-forward adaptive algorithm, it does not de-stabilize the closed-loop system as does other adaptive algorithms that

use feedback. It does not require input to be persistently exciting for convergence. It also does not require plant mathematical model. Due to these reasons, ILC has seen numerous applications since its birth. Several survey papers on ILC include [3] – [7].

ILC has been developed in both frequency domain and time domain. In frequency domain, a discrete-time band-pass filter is used to specify frequency range for learning, while another filter is used to ensure exponentially decaying error. In time domain, state-space model and Lyapunov stability theorem are normally used.

DRC-6

Internal model control (IMC) is a feedback control design technique that can deal with time delay or non-minimum-phase plant effectively. The controller consists of the all-pass part of the plant that contains all the time delay and non-minimum-phase terms, the inverse of the minimum-phase part of the plant, and the inverse of a low-pass filter.

Although IMC is a simple and powerful technique, it depends on accuracy of the plant model. Performance of the closed-loop system may aggravate if the plant model differs too much from the actual plant.

In this paper, ILC is used as feed-forward to improve the performance of the IMC. A frequency-domain ILC from Chapter 36 of [8] is used. Two schemes are presented. The first one uses ILC to reduce the tracking error, while the second one uses ILC to reduce the model output error in the model matching setting.

Experiments are performed on a Diesel engine's throttle in our laboratory using Arduino Mega 2560 board and Matlab and Simulink toolboxes.

Both schemes can be implemented with the throttle successfully. However, the second scheme, "the model matching", uses less control effort due to smoother reference input.

This paper is organized in this way. Section 2 introduces the Diesel engine throttle and its system identifications. Section 3 contains block diagrams of the two schemes used and also introduces the IMC and the ILC. Section 4 has the experimental results of both schemes, followed by conclusions in Section 5.

2. Diesel Engine Throttle

In Diesel engine, throttle is placed at the fresh-air entrance to the engine. Its opening affects primarily the manifold absolute pressure and the mass air flow. Accurate control of the throttle opening is vital to performance and emission of the engine.

Fig. 1 depicts the Diesel engine throttle used in our experiment. A DC motor needs 12V DC power supply and a PWM signal with varying duty cycle to move the throttle plate. A potentiometer measures the opening of the throttle plate as an analog signal. Arduino Mega 2560 microcontroller board, together with Matlab and Simulink toolboxes, are used as control hardware and software.

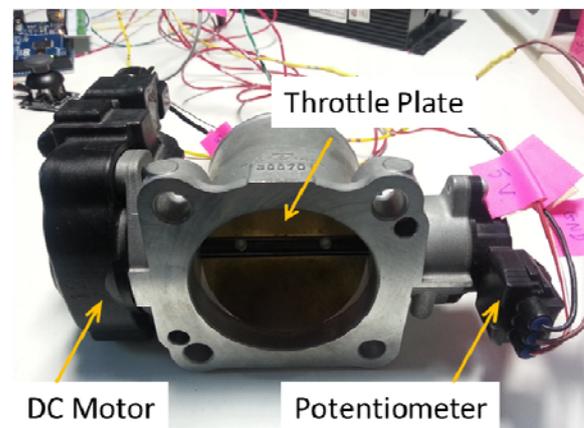


Fig. 1 The Diesel engine throttle used in the experiment.

A closed-loop system identification is performed by letting the reference for the opening percentage be a frequency-varying sine wave with frequency range 0.1 – 30 rad/s in 50 seconds. A roughly tuned PID controller is used to control the throttle plate. The experiment uses 0.05 seconds sampling time.

DRC-6

The duty cycle of the PWM input signal is an 8-bit integer representing values from 0 – 255. These values are recorded as control input u .

The corresponding analog signal from the potentiometer is a 10-bit integer representing values from 0 – 1023. After calibration, these numbers are converted to 0 – 100 opening percentage. The opening percentage is recorded as plant output y .

The identified continuous-time transfer function from u to y is given by

$$\frac{Y(s)}{U(s)} = \frac{13.19s + 0.2925}{s^2 + 11.58s + 0.7845}, \quad (1)$$

and the identified continuous-time state-space model is given by $\dot{x} = Ax + Bu$, $y = Cx$, where

$$A = \begin{bmatrix} 0 & 1 \\ -661.4 & -19.46 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 253.2 \end{bmatrix}, \quad (2)$$

$$C = [1 \quad 0].$$

State x_1 is the opening percentage y ; state x_2 is the opening velocity in % per second.

A reference model is chosen to be $\dot{x}_m = A_m x_m + B_m r$, $y_m = C x_m$, where

$$A_m = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}, \quad B_m = \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix}. \quad (3)$$

ω_n and ζ are chosen to be 10 rad/s and 1 to have appropriate transient response. x_m and y_m are the state and output of the reference model.

3. Frequency-Domain ILC-IMC System

In this section, two schemes in combining the ILC with the IMC are presented.

The first scheme has its block diagram shown in Fig. 2, where $C(s)$ is the IMC feedback controller, $P(s)$ is the plant whose plant model

is given by (1), the “ILC” and “Memory” blocks represent the iterative learning controller. r, u, u_1, u_2, y , and e are the repeated reference, total control effort, feedback control effort, iterative control effort, actual output, and tracking error, respectively. j denotes the j^{th} iteration of the iterative control. The ILC’s objective is to reduce the tracking error $e = r - y$.

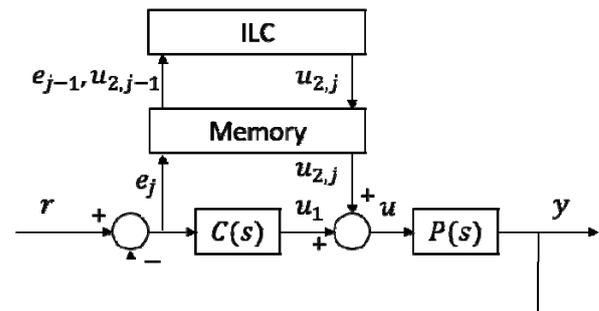


Fig. 2 ILC-IMC control system: tracking-error scheme.

The second scheme has its block diagram shown in Fig. 3. In addition to those in Fig. 2, P is the plant whose plant model is given by (2), G_m is the reference model (3), K is a weighting vector that places an emphasis on following the model output y_m or its time derivative \dot{y}_m , and \dot{y} is the time derivative of y . K is chosen as $[1, 0.001]$.

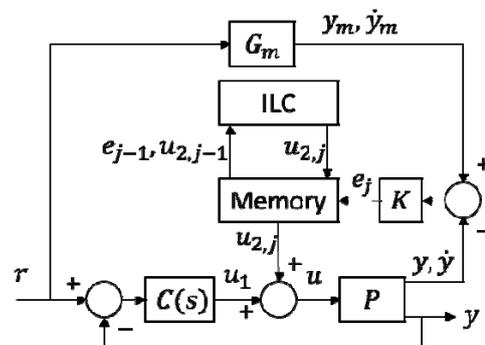


Fig. 3 ILC-IMC control system: model-matching scheme.

DRC-6

3.1 Internal model controller

This version of IMC belongs to [9]. Design of the IMC involves factorizing the plant model $P(s)$ into an invertible minimum-phase part $P_m(s)$ and a non-invertible all-pass part $P_a(s)$ that contains all right-half-plane zeros and time delays, that is,

$$P(s) = P_m(s)P_a(s),$$

$$P_a(s) = e^{-\theta s} \prod_i \frac{-s + z_i}{s + z_i}, \text{Re}(z_i) > 0, \theta > 0.$$

A desired closed-loop transfer function $T(s)$ from r to y is then specified as a multiplication of a low-pass filter and the all-pass part of the plant, that is,

$$y = T(s)r,$$

$$T(s) = f(s)P_a(s)$$

$$= \frac{P(s)C(s)}{1 + P(s)C(s)},$$

$$f(s) = \frac{1}{(\tau_c s + 1)^n},$$

where τ_c and n are the time constant and order of the filter.

By solving the equation above for $C(s)$, the resulting controller will contain only the inverse of the invertible part of the plant, not the non-invertible one, that is,

$$C(s) = P^{-1}(s) \frac{T(s)}{1 - T(s)}$$

$$= P_m^{-1}(s) \frac{1}{f^{-1}(s) - P_a(s)}. \quad (4)$$

The IMC idea is simple and has been proven to be successful in chemical-process applications that involve time delays.

3.2 Iterative learning controller

A general ILC algorithm is of the form

$$u_{2,j+1}(k) = Q(z) [u_{2,j}(k) + L(z)e_j(k)], \quad (5)$$

where $Q(z)$ and $L(z)$ are two filters to be designed.

Normally, $Q(z)$ is a band-pass filter that selects a range of frequencies where learning occurs. $L(z)$ can be any type of filter but most common types are P-type ($L(z) = \gamma z$) and PD-type ($L(z) = \gamma z + \lambda(z-1)$), where γ and λ are two positive design constants.

For example, the P-type ILC algorithm (5) with $Q(z) = 1$ is given by

$$u_{2,j+1}(k) = u_{2,j}(k) + \gamma e_j(k+1), \quad (6)$$

which can be thought of as an integrator in the iteration domain. Also, the error is measured one time step before the control effort is computed.

For the tracking-error scheme in Fig. 2, a condition that guarantees the convergence of the ILC control effort u_2 can be found as follows:

$$y_j = \frac{CP}{1+CP} r + \frac{P}{1+CP} u_{2,j},$$

$$e_j = r - y_j$$

$$= \frac{1}{1+CP} r - \frac{P}{1+CP} u_{2,j}$$

$$= e_0 - H u_{2,j},$$

$$u_{2,j+1}(k) = Q(z) [u_{2,j}(k) + L(z)e_j(k)]$$

$$= Q(z) (1 - L(z)H(z)) u_{2,j}$$

$$+ Q(z)L(z)e_0.$$

Therefore, a contractive mapping from $u_{2,j}$ to $u_{2,j+1}$ is achieved if

$$\|Q(z)(1 - L(z)H(z))\|_{\infty} < 1. \quad (7)$$

DRC-6

It can be shown that the contractive mapping condition for the model-matching scheme in Fig. 3 is the same as (7).

4. Experimental Results

This section contains the results of applying the ILC-IMC tracking-error scheme in Fig. 2 and the ILC-IMC model-matching scheme in Fig. 3 to the Diesel engine throttle in Section 2.

4.1 ILC-IMC tracking-error scheme

The control system in Fig. 2 is implemented. The sampling period is 0.05 seconds. The reference input is a square wave of 0.1 Hz frequency and values from 25 to 75 opening percentages.

The IMC controller (4), with $P_m(s)$ as in (1), $P_a = 1$, and $f(s) = 1/(0.05s + 1)$, is given by

$$C(s) = \frac{1.516s^2 + 17.55s + 1.189}{s^2 + 0.022s}$$

The P-type ILC (6) is used with $\gamma = 1$. To check the stability condition (7), with $Q = 1$, the magnitude of the transfer function

$$1 - \frac{\gamma z P}{1 + CP}$$

is plotted in Fig. 4. It can be seen that the condition (7) is met and the ILC control effort will convert with the designed parameters.

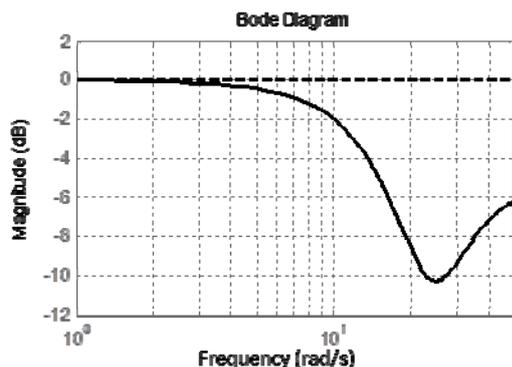


Fig. 4 Magnitude plot of $1 - \gamma z P / (1 + CP)$.

Fig. 5 shows the tracking results during the 2nd to 6th iterations and the 102nd to 106th iterations. Improvement in tracking from using the ILC is evident.

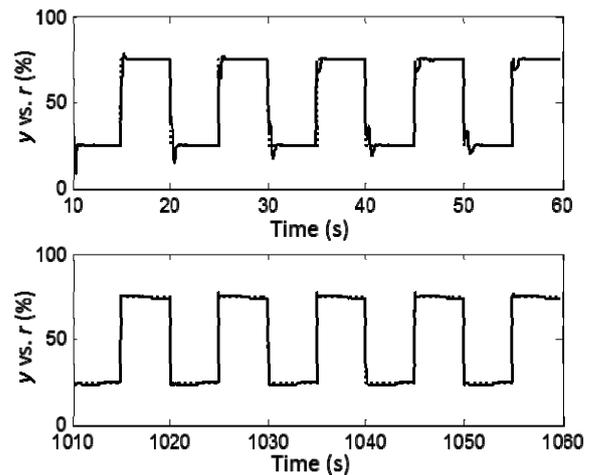


Fig. 5 Tracking-error scheme. Tracking results: (dash) reference r , (solid) output y .

Fig. 6 shows the root-mean-square values, taken over each iteration, of the tracking error $e = y_m - y$, the feedback IMC effort u_1 , and the feed-forward ILC effort u_2 . The tracking error decreases with some spikes probably due to non-repeated disturbances. The feedback control effort is reduced and replaced by the ILC effort. The ILC effort increases and will convert to a steady-state value with more iterations.

DRC-6

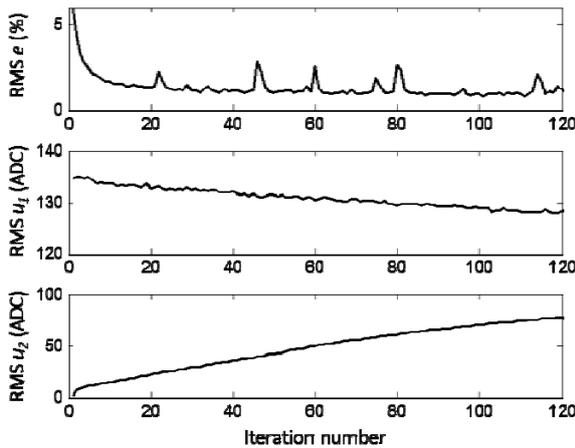


Fig. 6 Tracking-error scheme. (Top) Root-mean-square (RMS) value of the error $e = r - y$. (Middle) RMS value of the IMC effort u_1 . (Bottom) RMS value of the ILC effort u_2 .

Fig. 7 shows the tracking error e , the feedback IMC effort u_1 , the feed-forward ILC effort u_2 , and the total control effort u , during the 1st, 10th, and 125th iterations. As the iteration number increases, the tracking error reduces to near zero, and the control effort changes from using feedback to using the feed-forward ILC control.

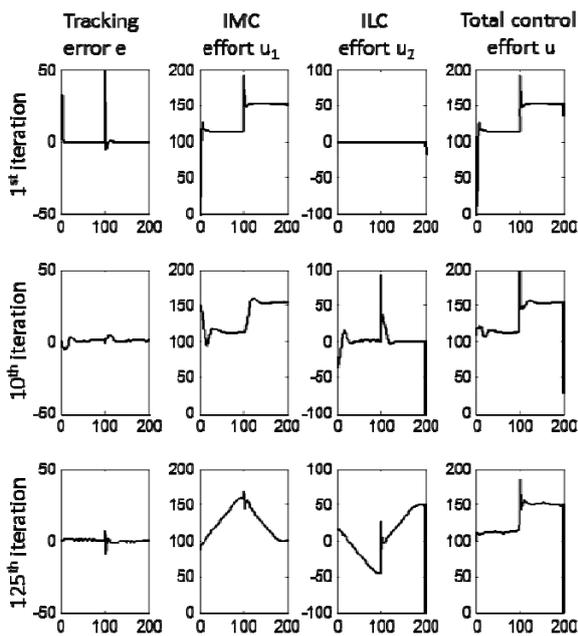


Fig. 7 Tracking-error scheme. The tracking error e , the IMC effort u_1 , the ILC effort u_2 , and the total control effort u , during the 1st, 10th, 125th iterations.

4.2 ILC-IMC model-matching scheme

The control system in Fig. 3 is implemented with the same IMC and ILC as in Section 4.1. Fig. 8 to Fig. 10 are analogous to Fig. 5 to Fig. 7.

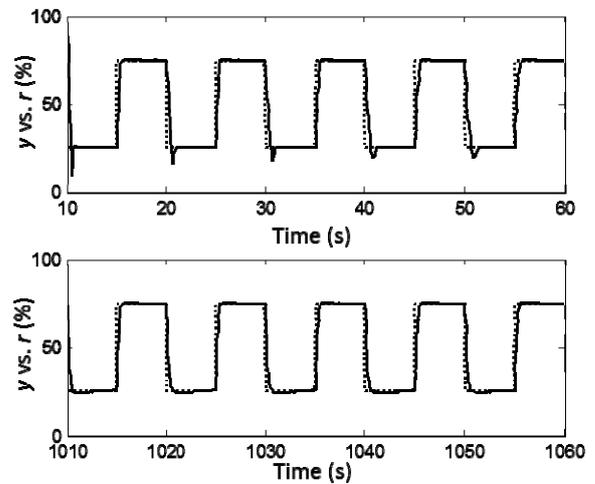


Fig. 8 Model-matching scheme. Tracking results: (dash) reference r , (solid) output y .

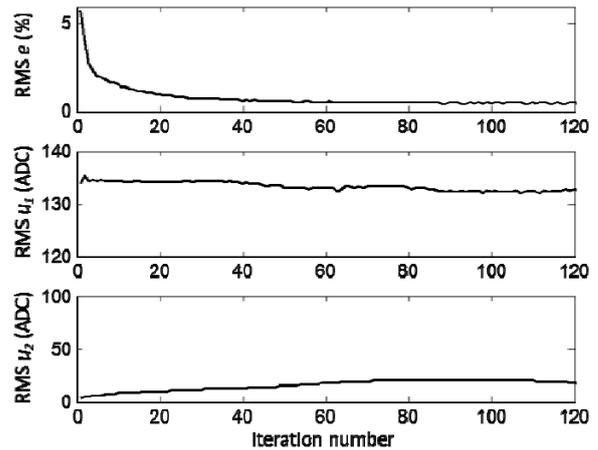


Fig. 9 Model-matching scheme. (Top) Root-mean-square (RMS) value of the error $e = r - y$.

DRC-6

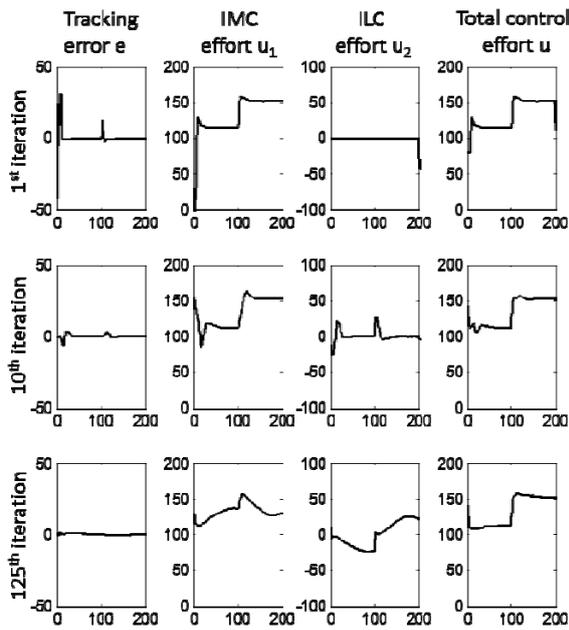


Fig. 10 Model-matching scheme. The tracking error e , the IMC effort u_1 , the ILC effort u_2 , and the total control effort u , during the 1st, 10th, 125th iterations.

4.3 Results comparison

By comparing Fig. 5 to Fig. 8, less overshoot can be seen in the model-matching scheme. This is due to the design of the reference model (3) such that its output y_m to be followed is a smooth signal not a square wave with abrupt change as in r .

By comparing Fig. 6 to Fig. 9, less amount of the overall control effort is used in the model-matching scheme; moreover, there is less spike in the RMS error. This is also due to the fact that the reference y_m is smoother than r .

By comparing Fig. 7 to Fig. 10, lower peak of the total control effort can be seen.

5. Conclusions

Two schemes in combining the ILC and IMC are presented. The first one uses ILC to reduce the tracking error $e = r - y$. The second one

uses ILC to reduce the reference model output error $e = y_m - y$. Both schemes successfully improve the performance of the IMC by reducing the tracking error to near zero after some iterations.

By comparing the two schemes, the second scheme, which uses model matching, offers some advantages in using less control effort and having better transient response.

6. References

- [1] Arimoto, S., Kawamura, S. and Miyazaki, F. (1984). Bettering operation of robots by learning, *J. of Robot. Syst.*, vol. 1(2), pp. 123 – 140.
- [2] Uchiyama, M. (1978). Formation of high-speed motion pattern of a mechanical arm by trial, *Trans. Soc. Instrum. Control Engineering (Japan)*, vol. 14(6), pp. 706 – 712.
- [3] Longman, R.W. (2000). Iterative learning control and repetitive control for engineering practice, *Int. J. of Control*, vol. 73(10), pp. 930 – 954.
- [4] Bristow, D.A., Tharayil, M. and Alleyne, A.G. (2006). A survey of iterative learning control, *Control Systems Magazine*, vol. 26(3), pp. 96 – 114.
- [5] Ahn, H.-S., Chen, Y.Q. and Moore, K.L. (2007). Iterative learning control: brief survey and categorization, *IEEE Trans. on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol. 37(6), pp. 1099 – 1121.
- [6] Wang, Y., Gao, F. and Doyle III, F.J. (2009). Survey on iterative learning control, repetitive control, and run-to-run control, *J. of Process Control*, vol. 19, pp. 1589 – 1600.

DRC-6

[7] Xu, J.-X. (2011). A survey on iterative learning control for nonlinear systems, *Int. J. of Control*, vol. 84(7), pp. 1275 – 1294.

[8] Levine, W.S. (2011). *The Control Handbook: Control System Advanced Methods*, 2nd edition, ISBN: 978-1-4200-7364-5, Taylor & Francis, Boca Raton.

[9] Skogestad, S. and Postlethwaite, I. (2005). *Multivariable Feedback Control: Analysis and Design*, 2nd edition, ISBN: 978-0-4700-1168-3, Wiley-Interscience.