Analysis and Design of A Learning Feedforward Controller Using Bartlet Window [1]

YangQuan Chen†, Ping Jiang‡ and Huadong Chen‡

† Center for Self-Organizing and Intelligent Systems (CSOIS),
Dept. of Electrical and Computer Engineering, Utah State University,
4160 Old Main Hill, Logan, UT 84322-4160, USA.
E-mail: yqchen@ieee.org; URL: http://www.crosswinds.net/~yqchen.

‡ Dept. of Information and Control, Tongji University,
Siping Rd. 1239, Shanghai, 200092, P. R. China.
Email: pjiang-k@online.sh.cn; URL: http://robot.tongji.edu.cn/.

Abstract

A learning feedforward controller (LFFC) using the Bartlet window function is proposed for a better tracking control of linear system over a finite time interval. LFFC is applied as a feedforward controller to the existing feedback controller. This paper demonstrates that using a simple window function – Bartlet (Fejer or triangular) window in signal processing, the design of a learning feedforward controller reduces to determining only two design parameters: the learning gain and the number of point in the window. Convergence analysis is presented together with a design procedure.

Key Words: Learning feedforward control (LFFC); iterative learning control (ILC); window function; Bartlet window; convergence analysis; controller design and tuning.

1 Introduction

Learning Feedforward Control (LFFC) [1, 2, 3, 4, 5] can be regarded as a variant of 'Iterative Learning Control' (ILC) [6, 7]. 'Iterative Learning Control' (ILC) [6] can be considered as a *value-added block* to enhance the feedback control performance by capitalizing the repetitiveness of system's operation. Clearly, the purpose of introducing the ILC is to utilize the system repetitions as *experience* to improve the system control performance even under incomplete *knowledge* of the system to be controlled. While the formal mathematically rigorous analysis is initially due to [6], the basic idea can be traced back to [8] and even to [9] which is commented in [10]. A recent survey and historical note on LFFC can be found in [11] while the design procedures are summarized in [12]. Detailed literature reviews on ILC research can be found in [13, 14].

Most of the existing work focused on the *analysis* issue of ILC schemes while the obtained convergence condition is clearly not enough for actual ILC applications. Therefore, in recent years increasing efforts have been made on the *design* issue of ILC. These can be observed from the latest books [15, 14], the dedicated ILC web server [16] and a recent survey on ILC *design* issue [17]. It is now quite clear that the efforts in LFFC or ILC research should be directed to developing a learning scheme with as few as possible design parameters or tuning knobs so that it could be as attractive as PID (proportional-integral-derivative) controllers widely used in industries with great success.

Using the second order B-splines network, LFFC [1, 3, 4, 5] scheme leads to a simple design procedure with only two tuning knobs: the B-splines support (d) and the learning gain (γ). This is an attractive feature to industrial applications and deserves focused investigation. In general, a neural network can be used in LFFC to generate the feedforward control signal $u_F(t)$. The inputs to the neural network can include signal $u_C(t)$ from the (existing) feedback controller, time t or state $x(t)$, and reference or desired trajectory $y_d(t)$. When repetitive operations are performed, all state dependent disturbances (e.g. cogging effect in linear motor servo [3]) can also be considered as time-periodic. Therefore, it is a common practice to choose the time t as the input of neural network to get the feed-forward control signal. To simplify the implementation, a second order B-splines network (BSN) is considered. With experimentally verified empirical design formula for d and γ given in [18], satisfactory experimental results

[1]This paper is partly supported by the EYTP of China. Corresponding author: Dr Chen, YangQuan. Tel.: 1(435)-797-0148, Fax:1(435)-797-2003; E-mail: yqchen@ieee.org.

were obtained [3, 18]. A stability analysis can be found in [4, 5] for LFFC scheme using time driven BSN and with assumptions about the plant to be controlled that a) the initial feedforward is with a triangular waveform; b) the learning updating law is in continuous time form. With relaxed assumptions, using the idea of noncausal filtering, a frequency domain learning convergence analysis and design formulae for d and γ are given in [19] where the second-order BSN with dilation 2 and 3 are also considered. Performance optimization is also considered for LFFC using the clustering and regularization techniques [20, 21]. According to the learning stability analysis [4, 19], it is required that d should be greater than a minimum value $d_{\min}$ which actually determines a cutoff frequency of the low pass learning filter.

This paper uses the similar idea of LFFC but without using the complex BSN filter. Instead, we propose to use a simple signal window function - Bartlet (Fejer or triangular) window for LFFC. Barnet window acts as a FIR filter to filter the feedback control signal of the previous iteration, which is in turn used to construct the LFFC signal of current iteration from the LFFC signal of the previous iteration. Therefore, the LFFC updating law takes a very simple form with only two design parameters: the learning gain and the total number of points of the Bartlet window. Convergence analysis is presented together with a design procedure. Some practical considerations in the parameter tuning are also outlined. The major contribution of this paper is the introduction of the windowing function commonly used in signal processing to learning feedforward control analysis and design.

The remainder of the paper is organized as follows. Sec. 2 briefly introduces LFFC scheme using Bartlet window function. In Sec. 3, a convergence analysis is given. Sec. 4 details the design issues of the proposed LFFC scheme. Finally, Sec. 5 concludes this paper.

2 Learning Feedforward Control (LFFC) Using Bartlet Window Function

2.1 LFFC Configuration

ILC, originally proposed as an open-loop control [6], has been considered as a feedforward control in addition to an existing feedback controller. The feedforward-feedback configuration of ILC algorithms has already been a standard consideration in either '*analysis*' or '*design*' [17] work on ILC. In fact, LFFC is in such a feedforward-feedback configuration. A block-diagram is shown in Fig. 1 where **FBC** stands for "feedback controller" and y_d is the given desired output trajectory to be tracked. After the i-th iteration (repetitive operation), the feedforward control signal u_{ff}^i and the feedback control signal u_{fb}^i are to be stored in the memory bank for constructing the feedforward control signal at the next iteration, i.e., u_{ff}^{i+1}. The stored feedback control signal u_{fb}^i are to go through a symmetrical Bartlet window (SBW) and then multiplied by a learning gain γ.

Remark 2.1 System Class: *As suggested in [17], when ILC starts to have a substantial impact on how control is actually done in industry, it will be the linear based ILC that leads the way. In engineering practice, to design a control system, it is very fundamental to have a linear proximal model $G(z)$ for frequencies below a frequency of interest, say, ω_c. For a feedback controlled system, it is almost sure that its frequency response can be well approximated by a linear system's, i.e., $G_c(z)$, the closed-loop transfer function. Therefore, at this point, it is understood that, $G(z)$ in Fig. 1 is a linearly major part of the* plant *which may be nonlinear.*

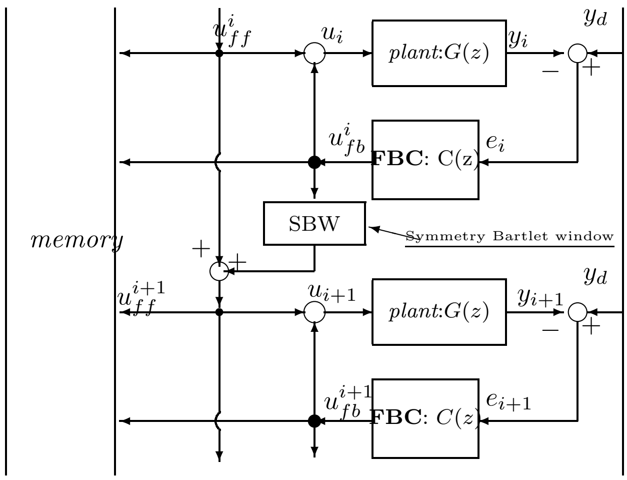

Figure 1: Block diagram of LFFC using symmetrical Bartlet window

2.2 Bartlet Window

Bartlet window (or Fejer window, triangular window) [22] w_B is a very simple signal window. It is defined by

$$w_B(n) = 1 - \frac{|n|}{M/2}, \quad n = -\frac{M}{2}, \cdots, -1, 0, 1, \cdots, \frac{M}{2}$$

where M is an even integer and $M+1$ the total number of points of the discrete Bartlet window. For discrete Fourier transform, the window is represented by

$$w_B(n) = \begin{cases} \frac{n}{M/2} & , \quad n = 0, 1, \cdots, \frac{M}{2} \\ \frac{M-n}{M/2} & , \quad n = \frac{M}{2}+1, \cdots, M-1 \end{cases}$$

with its DFT by

$$\bar{W}_B(\omega') = e^{-j(\frac{M}{2}-1)\omega'}\left(\frac{\sin(\frac{M}{4}\omega')}{\sin\frac{\omega'}{2}}\right)^2$$

since the symmetrical function of finite support $w(n)$ is shifted by $M/2-1$ positions to produce the DFT sequence. Therefore, for the symmetrical discrete Bartlet window, its DFT is simply

$$W_B(\omega') = (\frac{\sin(\frac{M}{4}\omega')}{\sin\frac{\omega'}{2}})^2.$$

Note that here the ω' is normalized to $[0, \pi]$.

2.3 LFFC Updating Law

Consider that the sampling period is T_s. Let $T_L = MT_s$. Clearly, T_L is far less than T, the total number of sampling points in each iteration. The learning updating law of the simple Bartlet window function w_B based LFFC as shown in Fig. 1 can be written as

$$u_{ff}^{i+1}(k) = u_{ff}^i(k) + \gamma_0 \sum_{j=-M/2}^{M/2} w_B(j) u_{fb}^i(k+j) \quad (1)$$

where γ_0 is a learning gain to be designed. The overall control signal is simply that

$$u_{i+1}(k) = u_{ff}^{i+1}(k) + u_{fb}^{i+1}(k). \quad (2)$$

As shown in Fig. 1, we have two parameters - γ the learning gain and T_L, are to be designed and specified.

Note that in [23], a local symmetrical integral (LSI) type ILC is proposed with continuous time domain updating law given by

$$u_{ff}^{i+1}(t) = u_{ff}^i(t) + \frac{\gamma}{2T_L}\int_{t-T_L}^{t+T_L} u_{fb}^i(\tau)\mathrm{d}\tau \quad (3)$$

and a discrete-time formula by

$$u_{ff}^{i+1}(k) = u_{ff}^i(k) + \frac{\gamma}{2M+1}\sum_{j=-M}^{M} u_{fb}^i(k+j). \quad (4)$$

This is a special case of the Bartlet window function based LFFC scheme (1) when $w_B = 1$. A more general noncausal filtering idea for ILC design was discussed in [24].

Remark 2.2 When $T_L \to 0$: *learning updating law (3) reduced to*

$$u_{ff}^{i+1}(t) = u_{ff}^i(t) + \gamma u_{fb}^i(t). \quad (5)$$

This was discussed in [25]. As will be shown later, this bare scheme will not *work properly in practice due to the lack of proper filtering of high frequency contents in* $u_{fb}(t)$.

Here, our control task is to track the given desired output trajectory $y_d(t)$ over a fixed time interval $[0, T]$ as closely as possible. With an existing feedback controller $C(s)$, the main objective of this paper is to use a learning feed forward controller given by updating law (1) to achieve a better tracking performance. In what follows, we will perform an analysis on the proposed LFFC convergence and then present a practical procedure for ILC design.

3 Convergence Analysis

Before performing the convergence analysis of the proposed learning scheme, we should clarify the existence problem. That is, for a given $y_d(t)$, we assume that there exists a unique feedforward $u_{ff}^{\infty}(t)$ such that $y_\infty(t) \to y_d(t)$ for all $t \in [0, T]$.

The convergence of the proposed learning controller is in the sense that u_{ff}^i approaches to a fixed point signal as i increases and meanwhile, $y_i(t) \to y_d(t)$. This is summarized in the following theorem.

Theorem 3.1 *A linear system shown in Fig. 1 is controlled by a suitable feedback controller which performs a given task repeatedly. A Bartlet window based LFFC scheme (1) is applied as a learning feedforward controller. There exists a real constant γ and a positive $T_L(0 < T_L < T)$ such that the learning process is convergent and furthermore,*

$$\lim_{i\to\infty} U_{ff}^i(z) \to Y_d(z)/G(z). \quad (6)$$

where $U_{ff}^i(z) = \mathcal{Z}[u_{ff}^i(t)]$ and $Y_d(z) = \mathcal{Z}[y_d(t)]$. The convergence rate is given by

$$\rho(\omega, \gamma, T_L) \stackrel{\triangle}{=} |\, 1 - \gamma H(\omega, T_L) G_c(e^{j\omega T_s}) \,| < 1, \quad (7)$$

where $G_c(z)$ is the closed loop transfer function and $G_c(z) = C(z)G(z)/(1 + C(z)G(z))$.

Theorem 3.1 implies that the iterative learning controller is essentially applied to inverse the plant to be controlled in an iterative manner. Since linear system is considered in this paper, in the sequel, frequency domain notion is used. Using $\mathcal{Z}$-transformation, the updating law (1) becomes

$$U_{ff}^{i+1}(z) = U_{ff}^i(z) + \gamma H(\omega, T_L) U_{fb}^i(z) \quad (8)$$

where $U_{fb}^i(z) = \mathcal{Z}[u_{fb}^i(t)]$, $z = e^{j\omega T_s}$ and

$$H(\omega, T_L) = (\frac{\sin(\frac{M}{4}\omega T_s)}{\frac{M}{2}\sin\frac{\omega T_s}{2}})^2,$$

$$\gamma = \gamma_0(\frac{M}{2})^2.$$

Now we proceed to present a proof of Theorem 3.1.

Proof: From Fig. 1, the feedback signal can be written as

$$U_{fb}(z) = -G_c(z)U_{ff}(z) + G_c(z)Y_d(z)/G(z). \quad (9)$$

Learning updating law (8) becomes,

$$\begin{aligned} U_{ff}^{i+1}(z) &= [1 - \gamma H(\omega, T_L)G_c(z)]U_{ff}^i(z) \\ &+\gamma H(\omega, T_L)G_c(z)Y_d(z)/G(z). \end{aligned} \quad (10)$$

Iterating (10), one obtains

$$\begin{aligned} U_{ff}^{i+1}(z) &= [1 - \gamma H(\omega, T_L)G_c(z)]^i U_{ff}^0(z) \\ &+\{1 - [1 - \gamma H(\omega, T_L)G_c(z)]^{i+1}\}Y_d(z)/G(z). \end{aligned} \quad (11)$$

Since $H(\omega, T_L)$ and $G_c(z)$ are essentially with a low pass filter characteristics, it is clearly possible to choose a suitable γ such that (7) is true. In addition, T_L can be used to shape $\rho(\omega, \gamma, T_L)$ which is the convergence rate in (11). With (7) and from (11)$\lim_{i\to\infty} U_{ff}^i(z) \to Y_d(z)/G(z)$ and moreover, $y_i(t) \to y_d(t)$ for all $t \in [0, T]$ as $i \to \infty$. ■

Remark 3.1 *It is implied in (8) that the initial condition of each iteration should be the same.*

4 Design Issues

4.1 Prior Knowledge

In this paper, it has been shown in the above that the design parameters of the proposed learning control scheme are only γ and T_L. One may argue that in the original D-type ILC scheme [6], there is only one design parameter Γ.

To make a fair comparison between two control schemes, one must take into account many factors. Among these factors, the amount of prior knowledge assumed for the controller design is vital for the control scheme to survive. In [6], the range of the first Markov parameter should be known *a priori*. This is an unusual and nonconventional requirement. Moreover, the derivatives of output tracking error are prone to noise amplification. As argued in [17], when faced to an actual system, one cannot assume zero knowledge available. In most engineering practice, it is quite common that the Nyquist curve information about the system is available. In this paper, the knowledge we used includes

- The frequency of the desired trajectory, which is less than a known frequency denoted by ω_d and $\omega_d < \omega_c$. ω_c is the cut-off frequency of the closed-loop system;
- An estimate of $G_c(e^{j\omega_c T_s})$ or $G_c(e^{j\omega_d T_s})$.

Clearly, the above knowledge is minimal for controller design.

4.2 Design Method for T_L

It is an intuition that a small T_L will bring in more high frequency signal components stored in the memory bank. These high frequency signal components may be accumulated due to different phase relationship from iteration to iteration. This is the major reason of the divergence for some ILC schemes which may be convergent at the initial iterations but as ILC runs, divergence can be observed in practical applications [17]. Meanwhile, a too large T_L may deteriorate the low frequency components of the signal when smoothing out the high frequency components in it. Therefore, a suitably chosen T_L is very important.

A simple consideration is that, the signal's energy can not be attenuated by half via $H(\omega, T_L)$. From the expression of $H(\omega, T_L)$, an estimate of T_L can be made.

An alternative practical design procedure can be like this. As at the first iteration only feedback controller is commissioned, at the end of the first iteration, performing discrete Fourier transform (DFT) of the feedback signal $u_{fb}^0(t)$ gives the spectrum information of the feedback signal. A frequency ω_c' can be chosen a little bit higher than the frequency corresponding to the magnitude peak in the amplitude-frequency plot of $u_{fb}^0(t)$. Then, one can use this ω_c' to obtain a design of T_L.

From the above discussions, T_L can now be designed.

Remark 4.1 *The learning scheme analyzed in [25] is actually the case when $T_L \to 0$ which caters for tracking desired trajectory with ultra high signal frequency, according to the discussion of this subsection. This is too stringent to be practically useful. Therefore, in practical use of learning control scheme like the one proposed in [25], a window type low pass filtering is required with a suitable T_L.*

4.3 Design Method for γ

To get a reasonable estimate of the upper limit of γ with **less** modeling effort is hard if not impossible. However, during practice, one can always start with a smaller, conservative γ via which the learning process converges. Then fine tune of γ is still possible as discussed in Sec. 4.5. The above discussions indicate that it is an easy task to make the proposed LFFC scheme work.

Full knowledge of $G_c(e^{j\omega T_s}), \quad \omega \leq \omega_c$ may be sometimes impractical. Therefore, it is assumed that at least the value of $G_c(e^{j\omega_c T_s})$ is available. In what follows, it will be shown that $G_c(e^{j\omega_c T_s})$ can be used to design a reasonable γ. Let $G_c(e^{j\omega T_s}) = A(\omega)e^{\theta(\omega)}$. Denote $\bar{\rho} = \rho^2(\omega, \gamma, T_L)$. Then, from (7),

$$\bar{\rho} = 1 - 2\gamma H(\omega, T_L)A(\omega)\cos\theta(\omega) + \gamma^2 H^2(\omega, T_L)A^2(\omega). \tag{12}$$

Clearly, γ should be chosen to minimize $\bar{\rho}$. From (12), the best γ should be

$$\gamma = \frac{\cos\theta(\omega)}{H(\omega, T_L)A(\omega)} \tag{13}$$

by setting $\frac{d\bar{\rho}}{d\gamma} = 0$. At frequency ω_c,

$$\gamma = \frac{\cos\theta(\omega_c)}{H(\omega_c, T_L)A(\omega_c)}. \tag{14}$$

It should be noted that γ may be negative at certain frequency range. For most applications, ω_d is quite small and in this case γ can be given approximately by

$$\gamma \leq \sqrt{2}. \tag{15}$$

When only $G_c(e^{j\omega_d T_s})$ is known, γ can be design similarly according to (13). If the knowledge of $G_c(e^{j\omega T_s})$ within a frequency range $[\omega_L, \omega_H]$ is known. A plot of $\gamma(\omega)$ is available from (14). This plot is useful in selecting a suitable γ when different frequencies of interest are to be considered in $[\omega_L, \omega_H]$.

In any case, it is possible to tune γ to make the LFFC convergence as fast as possible. However, as shown in Sec. 4.4, the convergence rate has its limit governed by the closed loop system dynamics alone.

4.4 Discussion on A Limit of LFFC Convergence Rate

Substituting (13) into (12), the corresponding minimal $\bar{\rho}$ is given by

$$(\bar{\rho})_{\gamma,\min} = \sin^2\theta(\omega), \tag{16}$$

i.e.,

$$\rho \geq |\sin\theta(\omega)| = \rho^*.$$

The above inequality implies that, for a given ω of interest, the LFFC convergence rate cannot be faster than the limit characterized by ρ^*. This limit is independent of learning schemes applied. The only way to achieve a faster LFFC convergence process is to well design the feedback controller $C(e^{j\omega T_s})$ such that the phase response of the closed-loop system will well behave as required in (16).

4.5 Some Heuristic Design Consideration

The following heuristic ideas may be helpful in tuning LFFC parameters. When applicable, some rule-based methods could be used.

- Re-determine T_L, or M, at the end of every iteration. This will not cost a lot but can keep a tight monitoring of possible variations of the system dynamics and the uncertainty/disturbance.
- When the LFFC starts with a smaller γ, increase γ while the tracking error keep decreasing and decrease γ while the tracking error keep increasing.
- Use a cautious (larger) T_L at the beginning of the LFFC process. Decrease T_L when the LFFC converges to a stage with little improvement. In this case, smaller T_L leaves more high frequency components of the feedback control signal in the memory bank. This in turn may further improve the convergence performance.

5 Conclusions

A new learning feedforward control (LFFC) updating law using Bartlet window function is proposed for tracking control of linear systems over a finite time interval. The LFFC is applied as a feedforward controller to the existing feedback controller. It is shown that the LFFC updating law takes a simple form with only two design parameters: the learning gain and the total number of points in Bartlet window. Convergence analysis is presented together with a design procedure. Some practical considerations in the parameter tuning are also outlined. Also, a limit on the ILC convergence rate has been discussed.

The major contribution of this paper is the introduction of the windowing function commonly used in signal processing to learning feedforward control analysis and design. Following the work presented in this paper, more window functions can be investigated and compared in our future research. Moreover, based on the work presented here, we observe that, in addition to the FIR and symmetrical properties of windowing functions, the analytical DFT formulae of the windows are particularly useful in designing M. Explicitly taking into account the uncertainty bound of $G(z)$ with a known nominal model $G_n(z)$ in designing γ is currently under investigation from H^∞ robust control theory point of view.

References

[1] J. G. Starrenburg, W. T. C. van Luenen, W. Oelen, and J. van Amerongen, "Learning feedforward controller for a mobile robot vehicle," *Control Engineering Practice*, vol. 14, no. 9, pp. 1221–1230, 1996.

[2] T.J.A. de Vries W.J.R. Velthuis and J. van Amerongen, "Learning feed forward control of a flexible beam," in *Proc. IEEE Int. Symp. on Intelligent Control ISIC '96*, Dearborn, MI, USA, 1996, pp. 103–108.

[3] G. Otten, T. J. A. de Vries, J. van Amerongen, A. M. Rankers, and E. W. Gaal, "Linear motor motion control using a learning feedforward controller," *IEEE/ASME Trans. Mechatronics*, vol. 2, no. 3, pp. 179–187, 1997.

[4] W. J. R. Velthuis, T. J. A. de Vries, P. Schaak, and E. W. Gaal, "Stability analysis of learning feedforward control," *Automatica*, vol. 36, no. 12, pp. 1889–1895, 12 2000.

[5] W. J. R. Velthuis, *Learning feed-forward control - theory, design and applications*, Ph.D. thesis, University of Twente, Enschede, The Netherlands, 2000.

[6] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.

[7] K. L. Moore, *Iterative learning control for deterministic systems*, Advances in Industrial Control. Springer-Verlag, 1993.

[8] M. Uchiyama, "Formulation of high-speed motion pattern of a mechanical arm by trial," *Trans. SICE (Soc. Instrum. Contr. Eng.)*, vol. 14, no. 6, pp. 706–712(in Japanese), 1978.

[9] Murray Garden, "Learning control of actuators in control systems," *United States Patent #3555252*, 1967.

[10] YangQuan Chen and Kevin L. Moore, "Comments on US Patent #3555252: Learning control of actuators in control systems," in *Proceedings of ICARCV'2000 (The Sixth International Conference on Control, Automation, Robotics and Vision*, Singapore, 2000, pp. CD–ROM.

[11] T. J. A. de Vries, W. J. R. Velthuis, and J. van Amerongen, "Learning feed-forward control: A survey and historical note," in *Proc. of the First IFAC Conference on Mechatronic Systems*, Darmstadt, Germany, 2000, IEEE, pp. 949 –954.

[12] W. J. R. Velthuis, T. J. A. de Vries, and J. van Amerongen, "Design procedure for a learning feedforward controller," in *Proc. of the First IFAC Conference on Mechatronic Systems*, Darmstadt, Germany, 2000, IEEE, pp. 955 –960.

[13] K. L. Moore, "Iterative learning control - an expository overview," *Applied & Computational Controls, Signal Processing, and Circuits*, vol. 1, no. 1, pp. 151–241, 1999.

[14] YangQuan Chen and Changyun Wen, *Iterative Learning Control: Convergence, Robustness and Applications*, vol. LNCIS-248 of *Lecture Notes series on Control and Information Science*, Springer-Verlag, London, 1999.

[15] Zeungnam Bien and Jian-Xin Xu, *Iterative Learning Control - Analysis, Design, Integration and Applications*, Kluwer Academic Publishers, 1998.

[16] YangQuan Chen, "Dedicated web server for Iterative Learning Control research," `http://cicserver.ee.nus.edu.sg/~ilc`, Oct. 1998.

[17] Richard W. Longman, "Designing iterative learning and repetitive controllers," in *Iterative Learning Control - Analysis, Design, Integration and Application*, Zeungnam Bien and Xu Jian-Xin, Eds., chapter 6, pp. 107–145. Kluwer Academic Publishers, 1998.

[18] W. J. R. Velthuis, T. J. A. de Vries, and E. Gaal, "Experimental verification of the stability analysis of learning feed-forward control," in *Proceedings 37th IEEE Conference on Decision and Control*, Tampa, Florida, USA, 1998, pp. 1225–1229.

[19] YangQuan Chen, Kevin L. Moore, and Vikas Bahl, "Improved path following of USU ODIS by learning feedforward controller using dilated B-spline network," in *The 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (IEEE CIRA 2001)*, Banff, Alberta, Canada, 2001, IEEE, pp. 59–64.

[20] W. J. R Velthuis, T. J. A. de Vries, and J. van Amerongen, "Performance optimisation of learning feed forward control," in *Proc. IFAC Symp. AI in Real Time Control (AIRTC'97)*, Kuala Lumpur, Malays, 1997, pp. 391–396.

[21] W. J. R. Velthuis, T. J. A. de Vries, and Menno Haring, "Regularisation in learning feed-forward control," in *Proceedings 6th Int. Conf. on Control, Automation, Robotics and Vision*, Singapore, 2000, pp. 406.1–406.6.

[22] A. D. Poularikas, *The handbook of formulas and table for signal processing*, The Electrical Engineering Handbook Series. CRC Press LLC and IEEE Press, New York, 1999.

[23] Y. Q. Chen, H. F. Dou, and K. K. Tan, "Local-symmetrical-integral-type iterative learning control," *Control Theory and Applications (ISSN 1000-8152)*, vol. 17, no. 3, pp. 347–352, 2000.

[24] Yangquan Chen, Tong Heng Lee, Jian-Xin Xu, and S. Yamamoto, "Noncausal filtering based design of iterative learning control," in *Proc. of the First Int. Workshop on Iterative Learning Control*, K. L. Moore, Ed., Hyatt Regency, Tampa, FL, USA, 1998, pp. 63–70.

[25] T.-Y. Kuc, J. S. Lee, and K. Nam, "An iterative learning control theory for a class of nonlinear dynamic systems," *Automatica*, vol. 28, no. 6, pp. 1215–1221, 1992.